2016 IEEE 57th Annual Symposium on Foundations of Computer Science

Compressing Interactive Communication under Product Distributions

Alexander A. Sherstov
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095 USA
Email: sherstov@cs.ucla.edu

***Abstract*—We study the problem of compressing interactive communication to its information content I, defined as the amount of information that the participants learn about each other's inputs. We focus on the case when the participants' inputs are distributed independently and show how to compress the communication to $O(I \log^2 I)$ bits, with no dependence on the original communication cost. This result improves quadratically on previous work by Kol (STOC 2016) and essentially matches the well-known lower bound $\Omega(I)$.**

***Keywords*-information complexity; communication complexity; protocol compression; interactive compression; product distributions**

I. Introduction

Classic work by Shannon [1], [2] shows how to optimally compress one-way communication to its information content, achieving in the limit a transmission cost equal to the entropy of the message. The corresponding problem for *interactive* communication has attracted increasing attention over the past two decades. Consider two computationally unbounded parties, Alice and Bob, with inputs $X \in \mathscr{X}$ and $Y \in \mathscr{Y}$, respectively, where $\mathscr{X}$ and $\mathscr{Y}$ are finite sets and the pair (X, Y) is distributed according to some known probability distribution on $\mathscr{X} \times \mathscr{Y}$. Alice and Bob exchange messages back and forth according to an agreed-upon randomized *protocol* in order to implement some functionality that depends on both inputs. One distinguishes between *public-coin* and *private-coin* protocols, corresponding to communication with or without a shared source of random bits. Information complexity theory [3], [4], [5], [6] studies a protocol's *information cost*, defined as the amount of information that Alice and Bob learn on average about each other's inputs from the history of messages exchanged between them (the *protocol transcript*). This complexity measure is quite different from *communication cost*, studied in Yao's communication complexity theory [7] and defined as the number of bits exchanged between Alice and Bob in the worst case on any input.

Basic properties of the entropy function ensure that a protocol's communication cost is always at least as large as its information cost, and the gap between the two quantities can be arbitrary. In this light, it is natural to ask whether the communication in every protocol π can be compressed to its information content while approximately preserving the protocol's functionality. In more detail, the approximate simulation of a given protocol π on given inputs X and Y by another protocol π' involves running π' on (X, Y) and interpreting the resulting transcript as a transcript of π. Alice and Bob may base their interpretations on their respective inputs X and Y, potentially arriving at distinct conclusions. In an accurate simulation, we require that their interpretations almost always agree and approximately follow the distribution of π's transcript on the input in question. Formally, π' *simulates* π *with error* ϵ if there exist a pair of "transcript interpretation" functions $a\colon \{0,1\}^* \to \{0,1\}^*$ and $b\colon \{0,1\}^* \to \{0,1\}^*$ for Alice and Bob such that the random variables (X, Y, Π, Π) and $(X, Y, a(X, \Pi'), b(Y, \Pi'))$ are at statistical distance at most ϵ, where Π and Π' denote the transcripts of π and π', respectively, on input (X, Y). The compression problem for interactive communication is the problem of simulating, with small error ϵ, a given protocol π by a protocol with communication cost as close as possible to the information cost of π. Apart from its basic importance, protocol compression is intimately related to *direct sum theorems* in communication complexity theory [3], [8], [9].

Protocol compression has been actively studied [10], [6], [9], [11], [12], [13], [14], [15] over the past two decades. In a groundbreaking paper, Barak et al. [6] showed how to compress any protocol with information cost I and communication cost C to a protocol with communication cost $\sqrt{IC}\,\mathrm{polylog}(C)$. Since the original communication cost C can be essentially infinite, it is natural to ask if compression independent of C is a possibility. The influential results of Braverman [11] and Braverman and Weinstein [12] answer this question in the affirmative, showing how to compress the communication in any protocol to $2^{O(I)}$ bits. Despite much subsequent research, these two incomparable bounds remain the strongest results for general protocol compression. On the lower bounds side, Ganor, Kol, and Raz [16], [17], [18] prove that Braverman's $2^{O(I)}$ compression is in general the best possible bound that does not depend on the original communication cost C. It is consistent with our current knowledge, however, that any protocol can be compressed to $I\,\mathrm{polylog}(C)$ bits, with only a nominal dependence on the original communication cost.

In this paper, we focus on the well-studied special case [6], [19], [20] of the protocol compression problem

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.64

when Alice and Bob's inputs X and Y are distributed independently. The resulting joint probability distribution μ of the inputs is called a *product distribution*, in reference to its representation as $\mu = \mu_{\mathscr{X}} \times \mu_{\mathscr{Y}}$ for some distributions $\mu_{\mathscr{X}}$ and $\mu_{\mathscr{Y}}$ on Alice and Bob's input sets, respectively. Braverman's $2^{O(I)}$ compression [11] of course applies to this special case as well, whereas Barak et al. [6] are able to strengthen their compression bound to $I\,\mathrm{polylog}(C)$ bits. These two bounds have complementary strengths, namely, independence of C and moderate growth with I. In a remarkable recent paper, Kol [20] shows how to achieve these desiderata simultaneously, for a compressed communication cost of $I^2\,\mathrm{polylog}(I)$ bits. We obtain a quadratic improvement on Kol's work, achieving a compressed communication cost of $O(I \log^2 I)$ bits and essentially matching the well-known lower bound of $\Omega(I)$.

Theorem 1 (Main result). *Let $0 < \epsilon < 1/2$ be given. Fix any public- or private-coin protocol π with input space $\mathscr{X} \times \mathscr{Y}$. Let μ be a product distribution on $\mathscr{X} \times \mathscr{Y}$, and let I be the information cost of π under μ. Then there is a public-coin protocol π' that simulates π with error ϵ under μ and has worst-case communication cost*

$$O\left(\frac{I}{\epsilon} \log^2 \frac{I}{\epsilon}\right).$$

Theorem 1 improves on previous compression schemes for product distributions with respect to all parameters. Our proof is inspired by the work of Barak et al. [6] and Kol [20], which we will describe shortly and contrast with our approach.

A. Background for Protocol Compression

We start with a brief review of relevant terminology and background; a thorough treatment of these technical preliminaries is available in Section II. Throughout this paper, we consider binary strings to be ordered by the prefix ordering $\preceq$. The terms *minimal* and *maximal*, when applied to strings, refer to this ordering $\preceq$. All trees in our work are binary and finite. We identify the vertices of a tree with binary strings in the usual manner, namely, the root corresponds to the empty string ε, and inductively the left child and right child of a vertex v correspond to the strings $v0$ and $v1$, respectively. A *cut* in a binary tree is any subset of the tree's vertices that intersects every root-to-leaf path in exactly one vertex. For example, the leaves of the tree form a cut. More generally, by truncating a given tree arbitrarily and considering the resulting set of leaves, one obtains a cut in the original tree. Given our identification of tree vertices with binary strings, we view cuts as subsets of $\{0,1\}^*$. The *floor* of cuts $\mathscr{C}_1$ and $\mathscr{C}_2$, denoted $\lfloor \mathscr{C}_1, \mathscr{C}_2 \rfloor$, is the set of minimal elements of $\mathscr{C}_1 \cup \mathscr{C}_2$. Analogously, the *ceiling* of cuts $\mathscr{C}_1$ and $\mathscr{C}_2$, denoted $\lceil \mathscr{C}_1, \mathscr{C}_2 \rceil$, is the set of maximal elements of $\mathscr{C}_1 \cup \mathscr{C}_2$. These definitions generalize in the obvious way to three or more cuts. For any collection of cuts, their floor and ceiling are also cuts (see Propositions 3.4 and 3.5 in the full version of this paper [21]).

Consider a randomized protocol with input space $\mathscr{X} \times \mathscr{Y}$. Assume for simplicity that it is a private-coin protocol, meaning that Alice and Bob do not have access to a shared source of random bits. They communicate by sending one bit at a time. A multibit message corresponds to several consecutive single-bit transmissions by the same sender. For any given history of previously transmitted bits, the protocol specifies which of the participants must send the next bit, which in turn is a function of the sender's private random string, the sender's input, and the history of previously transmitted bits. Formally, a private-coin protocol is given by a finite binary tree and a function $\pi\colon (\mathscr{A} \times \mathscr{X}) \cup (\mathscr{B} \times \mathscr{Y}) \to [0,1]$, where the sets $\mathscr{A}$ and $\mathscr{B}$ form a partition of the tree's internal vertices. We identify the protocol with its corresponding function π and use the same symbol for both. The vertices in $\mathscr{A}$ and $\mathscr{B}$ are said to be *owned* by Alice and Bob, respectively. The execution of π on a fixed pair of inputs (x, y) corresponds to a random walk on the protocol tree that starts at the root and proceeds one edge at a time, as follows. On reaching a vertex v owned by Alice, the walk proceeds to the left child with probability $\pi(v, x)$ and right child with the complementary probability $1 - \pi(v, x)$. Analogously, on reaching a vertex v owned by Bob, the walk proceeds to the left subtree with probability $\pi(v, y)$ and right subtree with probability $1-\pi(v, y)$. The walk terminates upon reaching a leaf vertex, which represents a *transcript* of the computation on input (x, y). Given our identification of tree vertices with binary strings, the transcript on a given input (x, y) is a random variable with range $\{0,1\}^*$.

In the rest of the introduction, let π be an arbitrary but fixed private-coin protocol, and let μ be a product distribution on the protocol's input space $\mathscr{X} \times \mathscr{Y}$. Let I denote the information cost of π with respect to μ. Let X and Y be a pair of inputs with joint distribution μ, and let Π be the transcript of π on input (X, Y). For fixed values $x \in \mathscr{X}$ and $y \in \mathscr{Y}$, define P, P_x, P_y, and $P_{x,y}$ to be the probability distributions that govern the random variables Π, $\Pi \mid X = x$, $\Pi \mid Y = y$, and $\Pi \mid X = x, Y = y$, respectively. Thus, $P, P_x, P_y, P_{x,y}$ are probability distributions on the leaves of the protocol tree. For a leaf or internal vertex v, we define $P(v), P_x(v), P_y(v), P_{x,y}(v)$ to be the corresponding probabilities of reaching a leaf in the subtree rooted at v. With this convention, $P, P_x, P_y, P_{x,y}$ are nonnegative functions defined at every vertex of the protocol tree. The restriction of any one of these functions to a cut of the protocol tree is a probability distribution. We further use the shorthands $P(v \mid u), P_x(v \mid u), P_y(v \mid u), P_{x,y}(v \mid u)$ to refer to the probabilities of reaching a leaf in the subtree rooted at v conditioned on reaching a leaf in the subtree rooted at u. Using the fact that μ is a product distribution, one easily verifies the identity $P(v)P_{x,y}(v) = P_x(v)P_y(v)$ for all x, y, v.

With this setup in place, we now describe the previous work by Barak et al. [6] and Kol [20].

B. Sampling Algorithm of Barak et al.

For $x \in \mathcal{X}$ and an internal vertex v, define $\mathbb{D}_x(v)$ to be the Kullback–Leibler divergence between the Bernoulli distributions $(P_x(v0 \mid v), P_x(v1 \mid v))$ and $(P(v0 \mid v), P(v1 \mid v))$. Similarly, define $\mathbb{D}_y(v)$ to be the Kullback–Leibler divergence between the Bernoulli distributions $(P_y(v0 \mid v), P_y(v1 \mid v))$ and $(P(v0 \mid v), P(v1 \mid v))$. Let $0 < \delta < 1$ be a small parameter, with order of magnitude $\delta = O(1/\log I)$. Without loss of generality [6], we may assume that $\mathbb{D}_x(v) \leqslant \delta$ and $\mathbb{D}_y(v) \leqslant \delta$ for all v, x, y. A key notion introduced by Barak et al. is that of a δ-*frontier*, defined separately for Alice and Bob. Alice's δ-frontier $\mathcal{F}_{x,\delta}$ is the set of minimal vertices v such that either v is a leaf or the sum of the $\mathbb{D}_x$ values of v's proper ancestors is at least δ. Analogously, Bob's δ-frontier $\mathcal{F}_{y,\delta}$ is the set of minimal vertices v such that either v is a leaf or the sum of the $\mathbb{D}_y$ values of v's proper ancestors is at least δ. A moment's reflection shows that $\mathcal{F}_{x,\delta}$ and $\mathcal{F}_{y,\delta}$ are cuts in the protocol tree.

Execution of π on input X, Y corresponds to sampling a random leaf of the protocol tree according to the probability distribution $P_{X,Y}$. Unfortunately, neither Alice nor Bob knows $P_{X,Y}$. Indeed, Alice only knows P and P_X, and Bob only knows P and P_Y. As the technical centerpiece of their analysis, Barak et al. prove that the restrictions of P_X and P_Y to the cut $\lfloor \mathcal{F}_{X,\delta}, \mathcal{F}_{Y,\delta} \rfloor$ are within a multiplicative constant c_0 of P almost at every vertex. We assume in this overview that the multiplicative bound holds everywhere. Under this simplifying assumption, the sampling procedure is as follows. Alice and Bob start by computing their respective frontiers $\mathcal{F}_{X,\delta}$ and $\mathcal{F}_{Y,\delta}$. They then use the shared randomness to sample a vertex V of the cut $\lfloor \mathcal{F}_{X,\delta}, \mathcal{F}_{Y,\delta} \rfloor$ according to the probability distribution P, by sampling a leaf according to P and sending each other its ancestors in $\mathcal{F}_{X,\delta}$ and $\mathcal{F}_{Y,\delta}$, respectively. To adjust for any multiplicative disparity between P and $P_{X,Y}$, they use *rejection sampling* [10], [22], [23], [6], whereby Alice accepts V with probability $P_X(V)/c_0P(V)$ and Bob independently accepts V with probability $P_Y(V)/c_0P(V)$. Conditioned on both parties accepting, which happens with probability $1/c_0^2$, the vertex V is a random element of the cut $\lfloor \mathcal{F}_{X,\delta}, \mathcal{F}_{Y,\delta} \rfloor$ governed by the correct probability distribution:

$$P(V) \cdot \frac{P_X(V)}{P(V)} \cdot \frac{P_Y(V)}{P(V)} = \frac{P_X(V)P_Y(V)}{P(V)} = P_{X,Y}(V).$$

By generating V in this manner, Barak et al. execute the initial part of π that corresponds to the shaded region of the protocol tree in Figure 1*a*. They then run their algorithm recursively on the protocol subtree rooted at V, eventually outputting a leaf distributed according to $P_{X,Y}$. For the cost analysis, consider the intermediate vertices generated by the algorithm as it works its way from the root to a leaf. The path segment between any two of them contributes at least δ toward the path's cumulative $\mathbb{D}_X$ or $\mathbb{D}_Y$ value. By the chain rule for the Kullback–Leibler divergence, it follows that the process terminates on average after $O(I/\delta) = O(I \log I)$ recursive calls. The communication cost of a single recursive call is $O(\log C)$, where C is the height of the protocol tree for π. As a result, the overall simulation has communication cost $I \operatorname{polylog}(C)$.

C. Kol's Sampling Algorithm

The most expensive step in the algorithm of Barak et al. is the transmission of the intersection points of $\mathcal{F}_{X,\delta}$ and $\mathcal{F}_{Y,\delta}$ with the root-to-leaf path sampled according to P. Their implementation involves the exchange of the actual intersection points, for a communication cost of $\Theta(\log C)$ bits, which can be essentially infinite even when the information cost I is small. Kol [20] proposed an alternate sampling procedure, based on discretization, that ingeniously eliminates the dependence of the cost on C. Specifically, Kol rounds the frontiers $\mathcal{F}_{X,\delta}$ and $\mathcal{F}_{Y,\delta}$ up with respect to a small and fixed collection of cuts known to both Alice and Bob, resulting in a pair of approximate frontiers $\overline{\mathcal{F}_{X,\delta}}$ and $\overline{\mathcal{F}_{Y,\delta}}$. Figure 1*b* illustrates Kol's construction, with the approximate frontiers shown as dashed lines. Instead of sampling from the cut $\lfloor \mathcal{F}_{X,\delta}, \mathcal{F}_{Y,\delta} \rfloor$ as Barak et al. do, Kol samples from the cut $\lceil \lfloor \overline{\mathcal{F}_{X,\delta}}, \mathcal{F}_{Y,\delta} \rfloor, \lfloor \overline{\mathcal{F}_{Y,\delta}}, \mathcal{F}_{X,\delta} \rfloor \rceil$. Using the fact that μ is a product distribution, Kol shows that this new sampling cut coincides almost always with $\lfloor \overline{\mathcal{F}_{X,\delta}}, \overline{\mathcal{F}_{Y,\delta}} \rfloor$ and therefore enables the efficient transmission of the intersection points with any root-to-leaf path.

Assuming for simplicity that Alice and Bob's frontiers $\mathcal{F}_{X,\delta}$ and $\mathcal{F}_{Y,\delta}$ are disjoint, Kol's complete sampling algorithm is as follows. First, one of the parties is randomly designated as the *leader*. Under Alice's leadership, the algorithm starts by sampling a root-to-leaf path according to P_X. This step uses the correlated sampling algorithm of Braverman and Rao [9] for the probability distributions P_X and P, with communication cost $O(\mathbf{E}\,\mathrm{KL}(P_X \parallel P)) \leqslant O(I)$ in expectation. If Bob's frontier $\mathcal{F}_{Y,\delta}$ precedes Alice's frontier $\mathcal{F}_{X,\delta}$ along the sampled path, they reject the path and go back to randomly choosing a leader. Otherwise, they compute the path's intersection V with the cut $\lfloor \overline{\mathcal{F}_{X,\delta}}, \mathcal{F}_{Y,\delta} \rfloor$, and Bob performs rejection sampling on V as in the work of Barak et al. If Bob rejects V, they go back to randomly choosing a leader; otherwise they accept V and run the algorithm recursively on the subtree rooted at V. This completes the description of the algorithm when Alice is the leader. Under Bob's leadership, the roles of Alice and Bob, and the roles of X and Y, are reversed. The cost analysis is similar to that of Barak et al., with the difference that the expected cost of a recursive call is now $O(I)$ rather than $O(\log I)$. Since the expected number of recursive calls does not exceed $I \operatorname{polylog}(I)$, the overall algorithm has communication cost $I^2 \operatorname{polylog}(I)$.

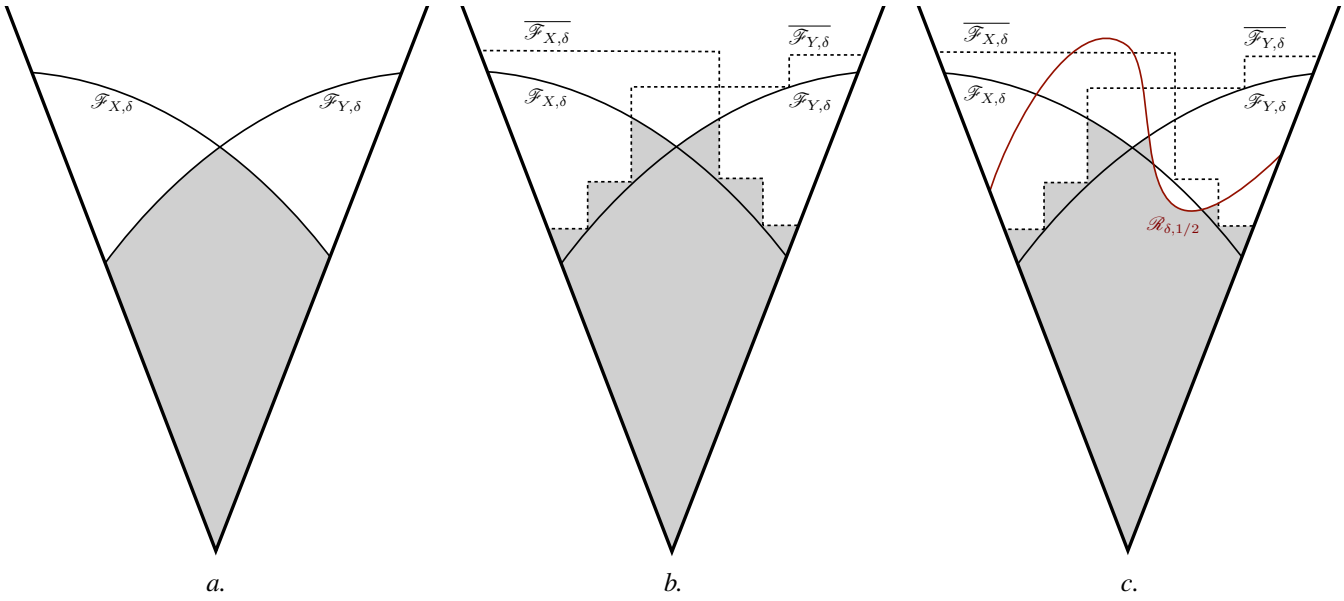

Figure 1. The sampling step in the algorithms of (*a*) Barak et al., (*b*) Kol, and (*c*) this work. The shaded area corresponds to the sampling subtree.

D. Our Sampling Algorithm

Kol's algorithm incurs essentially its entire communication cost at the beginning of a recursive call, when sampling a root-to-leaf path. The expected communication cost $\Theta(I)$ of this operation far exceeds its expected contribution $\Theta(1/\log I)$ to the cumulative $\mathbb{D}_X$ or $\mathbb{D}_Y$ value of the path that the algorithm eventually outputs. There are two reasons for this inefficiency. First, the portion of the sampled path beyond the sampling cut is always discarded, forfeiting the corresponding sampling effort. Second, the entire sampled path is discarded if the follower's frontier precedes the leader's along that path. We eliminate both sources of inefficiency and obtain an algorithm in which every step has communication cost proportional to that step's expected contribution to the progress measure.

We address the first problem by sampling the root-to-leaf path according to a "hybrid" distribution. The portion of the path up to the *leader's* sampling cut is distributed according to either P_X or P_Y as in Kol's algorithm, whereas the rest of the path is distributed according to the publicly known distribution P. The effect of this modification is that the segment of the path beyond the leader's sampling cut does not contribute to the sampling cost. To address the second source of inefficiency, we use a sampling cut different from Kol's. Let $\mathscr{R}_{\mathscr{X},\delta,1/2}$ denote the set of minimal vertices v such that the frontier $\mathscr{F}_{x,\delta}$ is encountered on the path from the root to v for at least half of the inputs $x \in \mathscr{X}$ weighted according to μ. Define $\mathscr{R}_{\mathscr{Y},\delta,1/2}$ analogously, and abbreviate $\mathscr{R}_{\delta,1/2} = \lfloor \mathscr{R}_{\mathscr{X},\delta,1/2}, \mathscr{R}_{\mathscr{Y},\delta,1/2} \rfloor$. These definitions ensure that for random X and Y, neither of the frontiers $\mathscr{F}_{X,\delta}$ or $\mathscr{F}_{Y,\delta}$ is very likely to precede $\mathscr{R}_{\delta,1/2}$ along a fixed root-to-leaf path. This motivates the use of $\lceil \lfloor \overline{\mathscr{F}_{X,\delta}}, \mathscr{F}_{Y,\delta}, \mathscr{R}_{\delta,1/2} \rfloor, \lfloor \overline{\mathscr{F}_{Y,\delta}}, \mathscr{F}_{X,\delta}, \mathscr{R}_{\delta,1/2} \rfloor \rceil$ as the sampling cut, instead of Kol's $\lceil \lfloor \overline{\mathscr{F}_{X,\delta}}, \mathscr{F}_{Y,\delta} \rfloor, \lfloor \overline{\mathscr{F}_{Y,\delta}}, \mathscr{F}_{X,\delta} \rfloor \rceil$. Figure 1*c* illustrates the resulting sampling subtree. To be precise, the sampling cut that we actually use is $\lceil \lfloor \overline{\mathscr{F}_{X,\delta}}, \mathscr{F}_{X,\Delta}, \mathscr{F}_{Y,\delta}, \mathscr{R}_{\delta,1/2} \rfloor, \lfloor \overline{\mathscr{F}_{Y,\delta}}, \mathscr{F}_{Y,\Delta}, \mathscr{F}_{X,\delta}, \mathscr{R}_{\delta,1/2} \rfloor \rceil$ for a large parameter $\Delta \gg 1$, but the distinction can be ignored on a first reading.

Summarizing, our modifications ensure that the sampling cost of every step in the algorithm is a constant plus a quantity proportional to the step's expected contribution to the progress measure. To prove that the overall sampling cost is at most $I \operatorname{polylog}(I)$, we must further argue that every step of the algorithm contributes on average $1/\operatorname{polylog}(I)$ to the progress measure. The corresponding claims in the work of Barak et al. and Kol were trivial to prove. In particular, the leader in Kol's algorithm is always guaranteed to contribute at least δ to the progress measure. Our situation is different because our choice of sampling cut effectively truncates the tree at $\mathscr{R}_{\delta,1/2}$, making a zero contribution a possibility for both the leader and the follower. Information-theoretically, the difficulty is as follows. For any *fixed* vertex $v \in \mathscr{R}_{\delta,1/2}$ and random X and Y, the probability that at least one of the frontiers $\mathscr{F}_{X,\delta}$ and $\mathscr{F}_{Y,\delta}$ is encountered on the path from the root to v is at least $1/2$. However, the sampled vertex V in the sampling cut is neither fixed nor independent of X or Y. We solve the problem by showing that any correlation between V and the protocol inputs causes information to be revealed about X and Y in a way that on average contributes to the progress measure instead of defeating it. We complete the proof of our main result with an amortized analysis of the cost versus progress, which too is more demanding than in previous work.

II. Preliminaries

We let $\log x$ denote the logarithm of x to base 2. We adopt the convention that $0/0 = 0$, justified throughout this paper by continuity arguments. For a binary string v, the shorthand $|v|$ stands for the length of v. We use calligraphic letters for finite sets ($\mathscr{A}, \mathscr{B}, \mathscr{C}, \mathscr{X}, \mathscr{Y}$), lowercase letters for set elements (x, y, u, v, w), and uppercase letters for random variables (X, Y, U, V, W). For a random variable X and an event E in the probability space, we let $X \mid E$ denote the random variable obtained from X by conditioning on E. The notation $X \sim \mu$ means that the random variable X is governed by the probability distribution μ. For random variables X and Y with a certain joint probability distribution, recall that $\mathbf{E}[Y \mid X]$ is not a specific number but a random variable defined as a function of X. Specifically, $\mathbf{E}[Y \mid X] = f(X)$ where f is given by $f(x) = \mathbf{E}[Y \mid X = x]$. Analogously, $\mathbf{P}[E \mid X]$ for an event E is not a specific number but a random variable defined as a function of X.

A. Strings

Recall that $\{0,1\}^*$ and $\{0,1\}^+$ refer to the set of binary strings and the set of nonempty binary strings, respectively. The empty string is denoted ε. The concatenation of the strings u and v is denoted uv. Consider the standard partial order $\prec$ on $\{0,1\}^*$, whereby $u \prec v$ if and only if $uw = v$ for some $w \neq \varepsilon$. The derived relations $\succ, \preceq, \succeq$ are given by

$$\begin{aligned} u \succ v \quad &\Leftrightarrow \quad v \prec u, \\ u \succeq v \quad &\Leftrightarrow \quad v \prec u \text{ or } v = u, \\ u \preceq v \quad &\Leftrightarrow \quad u \prec v \text{ or } v = u. \end{aligned}$$

Strings u and v are called *comparable* if $u \preceq v$ or $u \succeq v$, and *incomparable* otherwise. In addition to their role as relational operators, we use $\prec, \succ, \preceq, \succeq$ as the unary operators

$$\begin{aligned} \prec v &= \{u : u \prec v\}, & \succ v &= \{u : u \succ v\}, \\ \preceq v &= \{u : u \preceq v\}, & \succeq v &= \{u : u \succeq v\}. \end{aligned}$$

We refer to the elements of $\preceq v$ and $\succeq v$ as the *ancestors of* v and the *descendants of* v, respectively. Analogously, we call the elements of $\prec v$ and $\succ v$ the *proper ancestors of* v and the *proper descendants of* v, respectively. These unary operators naturally extend from strings to *sets* of strings, according to

$$\prec \mathscr{V} = \bigcup_{v \in \mathscr{V}} \prec v, \quad \succ \mathscr{V} = \bigcup_{v \in \mathscr{V}} \succ v, \quad \preceq \mathscr{V} = \bigcup_{v \in \mathscr{V}} \preceq v, \quad \succeq \mathscr{V} = \bigcup_{v \in \mathscr{V}} \succeq v.$$

In their unary capacity, the operators $\prec, \succ, \preceq, \succeq$ have the highest precedence.

B. Information Theory

All probability distributions in this work are defined on finite sets. For a probability distribution p on a set $\mathscr{X}$, its *support* is given by $\operatorname{supp} p = \{x \in \mathscr{X} : p(x) \neq 0\}$. For a subset $\mathscr{X}' \subseteq \mathscr{X}$, we let $p|_{\mathscr{X}'}$ denote the probability distribution induced by p on $\mathscr{X}'$. For probability distributions p and q on $\mathscr{X}$, their *Kullback–Leibler divergence* is given by $\mathrm{KL}(p \parallel q) = \sum_{x \in \mathscr{X}} p(x) \log(p(x)/q(x))$. In the context of the Kullback–Leibler divergence, we frequently identify a real number $0 \leqslant p \leqslant 1$ with the corresponding Bernoulli distribution $(p, 1-p)$ and use the shorthand $\mathrm{KL}(p \parallel q) = \mathrm{KL}((p, 1-p) \parallel (q, 1-q))$. Another distance measure for probability distributions is *statistical distance*, also known as *total variation distance* and defined by $\mathrm{TV}(p, q) = \max_{\mathscr{E} \subseteq \mathscr{X}} |p(\mathscr{E}) - q(\mathscr{E})|$. The following fact is proved in the full version of this paper [21, Fact 2.6].

Fact 2. *Let p and q be probability distributions on $\mathscr{X}$ such that $p(x) \leqslant c \cdot q(x)$ for all $x \in \mathscr{X}$. Then* $\mathrm{TV}(p, q) \leqslant 1 - \frac{1}{c}$.

In the context of the Kullback–Leibler divergence and statistical distance, we identify random variables with their corresponding probability distributions. For example, the notation $\mathrm{TV}(X, Y)$ refers to the statistical distance between the probability distributions of X and Y.

We use the notation $I(X; Y)$ and $I(X; Y \mid Z)$ for mutual information and conditional mutual information, respectively.

C. Communication Protocols

We consider communication between two computationally unbounded parties, called Alice and Bob, each with an input from some fixed finite set and with a private source of random bits. They send messages back and forth according to an agreed-upon protocol, where each message is a function of the sender's input, the sender's private random bits, and previously exchanged messages. Formally, a *private-coin communication protocol* is a tuple $(\mathscr{X}, \mathscr{Y}, T, \mathscr{A}, \mathscr{B}, \pi)$, where $\mathscr{X}$ and $\mathscr{Y}$ are the sets of possible inputs for Alice and Bob, respectively; T is a finite nonempty binary tree; $\mathscr{A}$ and $\mathscr{B}$ are disjoint sets that form a partition of the internal vertices of T; and $\pi \colon (\mathscr{A} \times \mathscr{X}) \cup (\mathscr{B} \times \mathscr{Y}) \to [0,1]$ is any function. The tree T is called the *protocol tree*. The vertices of $\mathscr{A}$ are said to be *owned by Alice*, and those of $\mathscr{B}$ are said to be *owned by Bob*. For brevity, we will identify a communication protocol with its corresponding function π.

The operational interpretation of a protocol π on a given pair of inputs $x \in \mathscr{X}$ and $y \in \mathscr{Y}$ is in terms of a random walk from the root of the protocol tree to a leaf. Specifically, at an internal vertex $v \in \mathscr{A}$, Alice sends 0 with probability $\pi(v, x)$ and sends 1 with the complementary probability $1 - \pi(v, x)$, directing the random walk to the left or right subtree, respectively. At an internal vertex $v \in \mathscr{B}$, Bob analogously sends 0 with probability $\pi(v, y)$, directing the random walk to the left subtree, and sends 1 with complementary probability. A *transcript* is the complete sequence of bits sent by Alice and Bob on a given pair of inputs over the course of the random walk from the root of the protocol tree to a leaf. Given our identification of tree vertices with binary strings, we identify the transcript with the leaf reached by

the random walk. The *communication cost* of protocol π, denoted $|\pi|$, is the height of the protocol tree, or equivalently the maximum number of bits exchanged by Alice and Bob in the worst case on any input. We let $\mathscr{V}(\pi)$ denote the set of vertices of the protocol tree for π, which includes both the internal vertices and the leaves. The set of leaves of the protocol tree is denoted $\mathscr{L}(\pi)$. We regard $\mathscr{V}(\pi)$ and $\mathscr{L}(\pi)$ as subsets of $\{0,1\}^*$.

A *public-coin communication protocol* is a probability distribution over a finite number of private-coin communication protocols, each with its own protocol tree. In a public-coin protocol, Alice and Bob use a shared source of random bits (a "public coin") to sample a random string R and then proceed to execute the private-coin protocol that corresponds to R. The *communication cost* of a public-coin protocol π, denoted $|\pi|$, is the maximum communication cost of the associated private-coin protocols. In particular, the length of the shared random string R does not count toward the communication cost of a public-coin protocol. The shared string is, however, always considered to be a part of the protocol transcript.

D. Information Cost

Fix a private-coin communication protocol π and a probability distribution μ on the input space of π. Let X and Y be random variables with joint distribution μ, corresponding to Alice and Bob's inputs, and let Π be the transcript of π on inputs X and Y. The *internal information cost of π with respect to μ* is defined as $\mathrm{IC}_\mu(\pi) = I(\Pi; X \mid Y) + I(\Pi; Y \mid X)$. Introduced by Barak et al. [6], this quantity measures the amount of information that Alice and Bob learn on average about each other's inputs by executing the protocol. A closely related notion is the *external information cost*, defined for π with respect to μ as $\mathrm{IC}^*_\mu(\pi) = I(\Pi; XY)$. This alternate quantity was introduced several years earlier by Chakrabarti et al. [3], with implicit uses in several other works. External information cost measures the amount of information that the protocol transcript reveals to an outside observer about the inputs X and Y.

Theorem 3 (Barak et al. [6]). *For any private-coin protocol π and any probability distribution μ, one has* $\mathrm{IC}_\mu(\pi) \leqslant \mathrm{IC}^*_\mu(\pi)$, *with equality for product distributions.*

The internal and external information cost of a *public-coin* protocol π are defined by conditioning on the shared random string R. Formally, $\mathrm{IC}_\mu(\pi) = I(\Pi; X \mid RY) + I(\Pi; Y \mid RX)$ and $\mathrm{IC}^*_\mu(\pi) = I(\Pi; XY \mid R)$.

E. Local View of Information Cost

External information cost admits a useful alternate characterization, based on the chain rule for mutual information. As before, fix a private-coin communication protocol π with input space $\mathscr{X} \times \mathscr{Y}$ and consider a probability distribution μ on $\mathscr{X} \times \mathscr{Y}$. Let X and Y be random variables with joint distribution μ, and let Π be the transcript of π on input X, Y. For $x \in \mathscr{X}$ and $y \in \mathscr{Y}$, define P, P_x, P_y, and $P_{x,y}$ to be the probability distributions that govern the random variables Π, $\Pi \mid X = x$, $\Pi \mid Y = y$, and $\Pi \mid X = x, Y = y$, respectively. Thus, $P, P_x, P_y, P_{x,y}$ are probability distributions on the leaves of the protocol tree. For a leaf or internal vertex v, recall from the Introduction that the shorthands $P(v), P_x(v), P_y(v), P_{x,y}(v)$ refer to the probability of reaching a leaf in the subtree of v. Similarly, $P(v \mid u), P_x(v \mid u), P_y(v \mid u), P_{x,y}(v \mid u)$ refer to the probability of reaching a leaf in the subtree of v conditioned on reaching a leaf in the subtree of u. For any vertex v of the protocol tree and inputs $x \in \mathscr{X}$ and $y \in \mathscr{Y}$, define

$$\mathbb{D}^{\pi,\mu}_x(v) = \begin{cases} \mathrm{KL}(P_x(v0 \mid v) \parallel P(v0 \mid v)) & \text{if } v \in \mathscr{A}, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathbb{D}^{\pi,\mu}_y(v) = \begin{cases} \mathrm{KL}(P_y(v0 \mid v) \parallel P(v0 \mid v)) & \text{if } v \in \mathscr{B}, \\ 0 & \text{otherwise,} \end{cases}$$

where as usual $\mathscr{A}$ and $\mathscr{B}$ stand for the sets of vertices owned by Alice and Bob, respectively. These quantities, introduced by Barak et al. [6], measure the information revealed about the protocol inputs locally due to the bit transmission at vertex v. Observe that for an internal vertex v, at most one of the quantities $\mathbb{D}^{\pi,\mu}_x(v), \mathbb{D}^{\pi,\mu}_y(v)$ is nonzero, whereas for every leaf vertex v, both quantities are zero. We abbreviate $\mathbb{D}^{\pi,\mu}_{x,y}(v) = \mathbb{D}^{\pi,\mu}_x(v) + \mathbb{D}^{\pi,\mu}_y(v) = \mathrm{KL}(P_{x,y}(v0 \mid v) \parallel P(v0 \mid v))$. For $\mathscr{S} \subseteq \mathscr{V}(\pi)$, we define $\mathbb{D}^{\pi,\mu}_x(\mathscr{S}) = \sum_{v \in \mathscr{S}} \mathbb{D}^{\pi,\mu}_x(v)$ and analogously for $\mathbb{D}^{\pi,\mu}_y(\mathscr{S})$ and $\mathbb{D}^{\pi,\mu}_{x,y}(\mathscr{S})$.

Theorem 4. *For any private-coin protocol π and distribution μ, one has* $\mathrm{IC}^*_\mu(\pi) = \mathbf{E}\, \mathbb{D}^{\pi,\mu}_{X,Y}(\prec\Pi)$, *where X and Y are random variables with joint distribution μ, and Π is the protocol transcript of π on input X, Y.*

The lower bound in this theorem was proved in [6]. A proof of the complete theorem is available in the full version of this paper [21], along with the following related result.

Theorem 5. *For every private-coin protocol π and distributions μ and $\tilde{\mu}$, one has* $\mathbf{E}\, \mathbb{D}^{\pi,\mu}_{X,Y}(\prec\Pi) \leqslant \mathbf{E}\, \mathbb{D}^{\pi,\tilde{\mu}}_{X,Y}(\prec\Pi)$, *where X and Y are random variables with joint distribution μ, and Π is the protocol transcript of π on input X, Y.*

F. Protocol Simulation

Let π be a private- or public-coin communication protocol with input space $\mathscr{X} \times \mathscr{Y}$, and let μ be a probability distribution on $\mathscr{X} \times \mathscr{Y}$. We say that π' *simulates π with error ϵ with respect to μ*, denoted

$$\pi' \hookrightarrow_{\mu,\epsilon} \pi,$$

if there are functions $a\colon \{0,1\}^* \to \{0,1\}^*$ and $b\colon \{0,1\}^* \to \{0,1\}^*$ such that $\mathrm{TV}((X, Y, \Pi, \Pi), (X, Y, a(X, \Pi'), b(Y, \Pi')) \leqslant \epsilon$,

where X and Y are random variables with joint distribution μ, and Π and Π' are the transcripts of π and π', respectively, on input X, Y. We remind the reader that for public-coin protocols, the protocol transcript always includes the shared random string. The triangle inequality for statistical distance gives:

THEOREM 6. *Let π, π', π'' be private- or public-coin protocols with input space $\mathscr{X} \times \mathscr{Y}$. Let μ be a probability distribution on $\mathscr{X} \times \mathscr{Y}$. Assume that $\pi'' \hookrightarrow_{\mu,\epsilon} \pi'$ and $\pi' \hookrightarrow_{\mu,\delta} \pi$. Then $\pi'' \hookrightarrow_{\mu,\epsilon+\delta} \pi$.*

The following well-known result shows that any public-coin protocol can be faithfully simulated by a private-coin protocol with no increase in information cost. A proof is available in the full version of this paper [21, Theorem 2.14].

THEOREM 7 (Folklore). *Let π be a public-coin protocol with input space $\mathscr{X} \times \mathscr{Y}$. Let μ be a probability distribution on $\mathscr{X} \times \mathscr{Y}$. Then there is a private-coin protocol π' such that*

$$\begin{aligned} &\pi' \hookrightarrow_{\mu,0} \pi, \\ &\mathrm{IC}_\mu(\pi') = \mathrm{IC}_\mu(\pi), \\ &\mathrm{IC}^*_\mu(\pi') = \mathrm{IC}^*_\mu(\pi). \end{aligned}$$

A private-coin protocol $\pi\colon (\mathscr{A} \times \mathscr{X}) \cup (\mathscr{B} \times \mathscr{Y}) \to [0,1]$ is *β-balanced* if the range of π is contained in the interval $[\frac{1}{2} - \beta, \frac{1}{2} + \beta]$. The following result, obtained by Barak et al. [6] and revisited recently by Kol [20], shows that any protocol can be simulated by a β-balanced protocol at the expense of an infinitesimal increase in information cost.

THEOREM 8 (Barak et al., Kol). *Let π be a private-coin protocol with input space $\mathscr{X} \times \mathscr{Y}$. Let μ be a probability distribution on $\mathscr{X} \times \mathscr{Y}$. Then for every $\beta > 0$ and $\epsilon > 0$, there exists a private-coin β-balanced protocol π' such that*

$$\begin{aligned} &\pi' \hookrightarrow_{\mu,\epsilon} \pi, \\ &\mathrm{IC}_\mu(\pi') \leqslant \mathrm{IC}_\mu(\pi) + \epsilon, \\ &\mathrm{IC}^*_\mu(\pi') \leqslant \mathrm{IC}^*_\mu(\pi) + \epsilon. \end{aligned}$$

III. Partial Simulation

Let π be a given protocol with information cost I under a product distribution μ. Recall that the goal of this paper is to construct a public-coin randomized protocol that accurately simulates π with respect to μ and has communication cost $O(I \log^2 I)$. We start by developing a public-coin randomized procedure that simulates a nontrivial initial portion of the protocol π. The complete simulation, analyzed in a later section, will involve repeated execution of this partial procedure until the communication allotment is reached.

THEOREM 9 (Partial simulation, $\sigma_{\pi,\mu,\epsilon}$). *Let $0 < \epsilon < 1/2$ be given. For $\beta = \beta(\epsilon) > 0$ sufficiently small, fix any β-balanced private-coin protocol π with input space $\mathscr{X} \times \mathscr{Y}$, and any product distribution μ on $\mathscr{X} \times \mathscr{Y}$. Then there is a public-coin randomized protocol $\sigma_{\pi,\mu,\epsilon}$ with input space $\mathscr{X} \times \mathscr{Y}$ whose execution allows Alice and Bob to agree on a vertex of the protocol tree for π, subject to the following properties:*

$$\sum_{w \preceq v} \frac{\mathbf{P}[W = w \mid X, Y]}{\mathbf{P}[\Pi \geq w \mid X, Y]} \leqslant 1 + \epsilon \quad \forall v \in \mathscr{V}(\pi) \qquad \textbf{(accuracy)}$$

$$\mathbf{P}[W \in \mathscr{L}(\pi)] + \log\left(\frac{1}{\epsilon}\right) \mathbf{E}\, \mathbb{D}^{\pi,\mu}_{X,Y}(\prec W) \geqslant \frac{1}{c} \qquad \textbf{(progress)}$$

$$\left.\begin{aligned} &C \leqslant C' + C'' + c \log \frac{1}{\epsilon} \\ &\mathbf{E}\, C' \leqslant c(\mathbf{E}\, \mathbb{D}^{\pi,\mu}_{X,Y}(\prec W) + \epsilon\, \mathbf{E}\, \mathbb{D}^{\pi,\mu}_{X,Y}(\prec \Pi)) \\ &\mathbf{P}[C'' > 0] \leqslant \epsilon \end{aligned}\right\} \qquad \textbf{(cost)}$$

where

(i) *X, Y are random variables with joint distribution μ;*

(ii) *Π is the transcript of π on input X, Y;*

(iii) *$W \in \mathscr{V}(\pi)$ is Alice and Bob's agreed-upon vertex after executing $\sigma_{\pi,\mu,\epsilon}$ on input (X, Y), and $C \in \mathbb{N}$ is the communication cost of that execution;*

(iv) *$C', C'' \in \mathbb{N}$ are auxiliary random variables;*

(v) *W, C, C', C'' are completely determined by the transcript of $\sigma_{\pi,\mu,\epsilon}$;*

(vi) *$c > 1$ is an absolute constant.*

The proof of Theorem 9, sketched in the Introduction, is rather lengthy and technical. It can be found in the full version of this paper [21, Sections 4.1–4.9].

IV. Complete Simulation

Building on the sampling procedure of the previous section, we now prove the main result of this work.

THEOREM 10 (Main theorem). *Let $0 < \epsilon < 1/2$ be given. Fix any public- or private-coin protocol π with input space $\mathscr{X} \times \mathscr{Y}$. Let μ be a product distribution on $\mathscr{X} \times \mathscr{Y}$, and abbreviate $I = \mathrm{IC}_\mu(\pi)$. Then there is a public-coin protocol π' with worst-case communication cost*

$$O\left(\frac{I}{\epsilon} \log^2 \frac{I}{\epsilon}\right)$$

such that $\pi' \hookrightarrow_{\mu,\epsilon} \pi$.

The remainder of this section is devoted to the proof of Theorem 10. Let $\delta = \delta(I, \epsilon) > 0$ be an accuracy parameter to be set later, and let $\beta = \beta(\delta) > 0$ be sufficiently small in the sense of Theorem 9. By Theorems 6–8, we may assume that π is a private-coin β-balanced protocol. Recall that our proof strategy in simulating π will be to repeatedly apply the partial sampling procedure of the previous section until a communication limit is exceeded. We will argue that the resulting simulation reaches a leaf with high probability and that its distribution is statistically close to the distribution of the transcript of π on the corresponding input.

A. A Stochastic Process

Let X, Y be a pair of inputs with joint distribution μ. We define a discrete stochastic process given by the random variables X, Y, and

$$(\pi_t, \mu_t, R_t, M_t, W_t, C'_t, C''_t), \qquad t = 1, 2, 3, \ldots, \tag{1}$$

where $\mu_1, \mu_2, \mu_3, \ldots$ are product distributions on $\mathscr{X} \times \mathscr{Y}$. We let $\pi_1 = \pi$ and $\mu_1 = \mu$. For $t \geqslant 1$, the random variables X, Y, π_t, μ_t give rise to $R_t, M_t, W_t, C'_t, C''_t, \pi_{t+1}, \mu_{t+1}$ in an inductive manner as follows.

(i) Execute the public-coin protocol $\sigma_{\pi_t,\mu_t,\delta}$ from Theorem 9 on input X, Y. Let R_t and M_t denote the shared random string and the rest of the protocol transcript, respectively, from that execution. Let W_t, C'_t, C''_t be the corresponding additional random variables from Theorem 9, each of which is completely determined by the tuple (π_t, μ_t, R_t, M_t).

(ii) Define π_{t+1} to be the private-coin protocol corresponding to the protocol subtree of π_t rooted at W_t. Thus, vertex W_t of the protocol tree for π_t corresponds to vertex ε (the root) of the protocol tree for π_{t+1}.

(iii) Define μ_{t+1} to be the posterior probability distribution on $\mathscr{X} \times \mathscr{Y}$ obtained by conditioning μ_t on the transcript (R_t, M_t) of protocol $\sigma_{\pi_t,\mu_t,\delta}$. Recall that conditioning a product distribution on a protocol transcript results in a product distribution. Thus, μ_{t+1} is a product distribution, maintaining the promised invariant.

We let P denote the resulting infinite sequence (1) of random variables. For $t = 1, 2, 3, \ldots$, we let $P_{\leqslant t}$ denote the restriction of P to the first t stages of the stochastic process. In other words, $P_{\leqslant t}$ stands for

$$(\pi_1, \mu_1, R_1, M_1, W_1, C'_1, C''_1), \ldots, (\pi_t, \mu_t, R_t, M_t, W_t, C'_t, C''_t), \pi_{t+1}, \mu_{t+1},$$

where the inclusion of π_{t+1} and μ_{t+1} is motivated by the fact that they are fully determined by the previous tuple. In this notation, μ_{t+1} is the probability distribution that governs the random variable $XY \mid P_{\leqslant t}$.

We define the random variable Π as the transcript of π on input X, Y. More generally, we define Π_t as the transcript of π_t on input X, Y. We stress that the inputs X, Y and the auxiliary random variables $\Pi, \Pi_1, \Pi_2, \Pi_3, \ldots$ are not part of P and in particular do not appear in any $P_{\leqslant t}$. Observe also that Π_t is independent of P given X, Y, π_t.

B. Accuracy Analysis

The focal point of the proof is the random walk $\varepsilon \preceq W_1 \preceq W_1W_2 \preceq W_1W_2W_3 \preceq \ldots \preceq W_1W_2\ldots W_t \preceq \ldots$ in the protocol tree for π. We start by studying how accurately this random walk models the actual protocol transcript, Π. For a fixed string w and any $t \geqslant 1$,

$$\begin{aligned}
&\mathbf{P}[W_t\Pi_{t+1} = w \mid X, Y, P_{\leqslant t-1}] \\
&= \sum_{v \preceq w} \mathbf{P}[W_t = v \mid X, Y, P_{\leqslant t-1}]\, \mathbf{P}[v\Pi_{t+1} = w \mid W_t = v, X, Y, P_{\leqslant t-1}] \\
&= \sum_{v \preceq w} \mathbf{P}[W_t = v \mid X, Y, P_{\leqslant t-1}]\, \mathbf{P}[\Pi_t = w \mid \Pi_t \succeq v, X, Y, P_{\leqslant t-1}] \\
&= \mathbf{P}[\Pi_t = w \mid X, Y, P_{\leqslant t-1}] \sum_{v \preceq w} \frac{\mathbf{P}[W_t = v \mid X, Y, P_{\leqslant t-1}]}{\mathbf{P}[\Pi_t \succeq v \mid X, Y, P_{\leqslant t-1}]} \\
&\leqslant (1+\delta)\, \mathbf{P}[\Pi_t = w \mid X, Y, P_{\leqslant t-1}],
\end{aligned} \tag{2}$$

where the second step follows from the definition of Π_{t+1} as the transcript of π_{t+1} on input X, Y, with π_{t+1} in turn obtained from π_t by restricting to the protocol subtree rooted at W_t; and the final step uses Theorem 9. Rewriting (2),

$$\begin{aligned}
&\mathbf{P}[W_1W_2\ldots W_t\Pi_{t+1} = w \mid X, Y, P_{\leqslant t-1}] \\
&\quad \leqslant (1+\delta)\, \mathbf{P}[W_1W_2\ldots W_{t-1}\Pi_t = w \mid X, Y, P_{\leqslant t-1}].
\end{aligned}$$

Passing to expectations with respect to $P_{\leqslant t-1}$,

$$\begin{aligned}
&\mathbf{P}[W_1W_2\ldots W_t\Pi_{t+1} = w \mid X, Y] \\
&\quad \leqslant (1+\delta)\, \mathbf{P}[W_1W_2\ldots W_{t-1}\Pi_t = w \mid X, Y],
\end{aligned}$$

whence by induction

$$\begin{aligned}
&\mathbf{P}[W_1W_2\ldots W_t\Pi_{t+1} = w \mid X, Y] \\
&\qquad \leqslant (1+\delta)^t\, \mathbf{P}[\Pi = w \mid X, Y].
\end{aligned} \tag{3}$$

In view of Fact 2, we arrive at

$$\begin{aligned}
\mathrm{TV}((X, Y, \Pi), (X, Y, W_1W_2\ldots W_t\Pi_{t+1})) &\leqslant 1 - \frac{1}{(1+\delta)^t} \\
&\leqslant t\delta.
\end{aligned} \tag{4}$$

Here, $W_1W_2\ldots W_t\Pi_{t+1}$ refers to the concatenation of $W_1, W_2, \ldots, W_t, \Pi_{t+1}$ rather than to the composite random variable $(W_1, W_2, \ldots, W_t, \Pi_{t+1})$. This distinction is essential from the point of view of information-theoretic distance.

C. Expected Information Gain

We will now obtain an upper bound on the progress measure $\mathbf{E}\, \mathbb{D}^{\pi_t,\mu_t}_{X,Y}(\prec W_t)$, which plays a critical role in relating the communication requirements of the stochastic process to the information cost of the original protocol π. Since π_{t+1} is the protocol corresponding to the subtree of π_t rooted at W_t,

$$\begin{aligned}
&\mathbf{E}\, \mathbb{D}^{\pi_t,\mu_t}_{X,Y}(\prec W_t) \\
&\qquad = \mathbf{E}\, \mathbb{D}^{\pi_t,\mu_t}_{X,Y}(\prec (W_t\Pi_{t+1})) - \mathbf{E}\, \mathbb{D}^{\pi_{t+1},\mu_t|W_t}_{X,Y}(\prec \Pi_{t+1}),
\end{aligned} \tag{5}$$

where the shorthand $\mu_t \mid v$ for a string $v \in \{0,1\}^*$ refers to the posterior probability distribution on $\mathscr{X} \times \mathscr{Y}$ obtained from μ_t by conditioning on $\Pi_t \succeq v$. By (2),

$$\begin{aligned}
&\mathbf{E}[\mathbb{D}^{\pi_t,\mu_t}_{X,Y}(\prec (W_t\Pi_{t+1})) \mid X, Y, P_{\leqslant t-1}] \\
&\qquad \leqslant (1+\delta)\, \mathbf{E}[\mathbb{D}^{\pi_t,\mu_t}_{X,Y}(\prec \Pi_t) \mid X, Y, P_{\leqslant t-1}]
\end{aligned}$$

and hence

$$\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_t,\mu_t}(\prec(W_t\Pi_{t+1})) \leqslant (1+\delta)\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_t,\mu_t}(\prec\Pi_t). \tag{6}$$

We now examine the other expectation on the right-hand side of (5). We claim that

$$\mathbf{E}[\mathbb{D}_{X,Y}^{\pi_{t+1},\mu_t|W_t}(\prec\Pi_{t+1}) \mid P_{\leqslant t}] \geqslant \mathbf{E}[\mathbb{D}_{X,Y}^{\pi_{t+1},\mu_{t+1}}(\prec\Pi_{t+1}) \mid P_{\leqslant t}]. \tag{7}$$

Conditioning on $P_{\leqslant t}$ fixes $\pi_t, \mu_t, W_t, \mu_{t+1}, \pi_{t+1}$, among other things, which means that the expectation on both sides of this inequality is with respect to random input X, Y and the resulting transcript Π_{t+1} in protocol π_{t+1}. But by definition, the posterior probability distribution of X, Y conditioned on $P_{\leqslant t}$ is μ_{t+1}. The claimed inequality (7) now follows from Theorem 5. Passing to expectations with respect to $P_{\leqslant t}$, we conclude that

$$\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_{t+1},\mu_t|W_t}(\prec\Pi_{t+1}) \geqslant \mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_{t+1},\mu_{t+1}}(\prec\Pi_{t+1}), \tag{8}$$

which along with (5) and (6) leads to our sought upper bound on the progress measure in the t-th step of the stochastic process:

$$\begin{aligned}&\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_t,\mu_t}(\prec W_t)\\ &\qquad\leqslant (1+\delta)\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_t,\mu_t}(\prec\Pi_t) - \mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_{t+1},\mu_{t+1}}(\prec\Pi_{t+1}).\end{aligned} \tag{9}$$

As a result,

$$\begin{aligned}\sum_{i=1}^{t}\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec W_i) &\leqslant \sum_{i=1}^{t}(1+\delta)^{t-i}\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec W_i)\\ &\leqslant \sum_{i=1}^{t}(1+\delta)^{t-i+1}\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec\Pi_i)\\ &\qquad - \sum_{i=1}^{t}(1+\delta)^{t-i}\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_{i+1},\mu_{i+1}}(\prec\Pi_{i+1})\\ &= (1+\delta)^t\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_1,\mu_1}(\prec\Pi_1) - \mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_{t+1},\mu_{t+1}}(\prec\Pi_{t+1})\\ &\leqslant (1+\delta)^t\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_1,\mu_1}(\prec\Pi_1)\\ &= (1+\delta)^t\,\mathrm{IC}_{\mu_1}^{*}(\pi_1)\\ &= (1+\delta)^t\,\mathrm{IC}_{\mu_1}(\pi_1)\\ &= (1+\delta)^t I,\end{aligned} \tag{10}$$

where the second, fifth, and sixth steps use (9), Theorem 4, and Theorem 3, respectively. An analogous calculation involving a telescoping sum shows that

$$\sum_{i=1}^{t}(\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec W_i) + \delta\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec\Pi_i)) \leqslant (1+2\delta)^t I. \tag{11}$$

D. Expected Time to Leaf and Communication Cost

Using the new upper bound (10) on the sum of progress terms, we now show that the random walk reaches a leaf reasonably quickly and with high probability has small communication cost. The first t stages of the stochastic process fail to reach a leaf with probability given by

$$\begin{aligned}&\mathbf{P}[W_t \notin \mathscr{L}(\pi_t)]\\ &\quad= \mathbf{P}[W_i \notin \mathscr{L}(\pi_i) \text{ for } i = 1, 2, \ldots, t]\\ &\quad= \prod_{i=1}^{t}\mathbf{P}[W_i \notin \mathscr{L}(\pi_i) \mid W_{i-1} \notin \mathscr{L}(\pi_{i-1})]\\ &\quad\leqslant \left(\frac{1}{t}\sum_{i=1}^{t}\mathbf{P}[W_i \notin \mathscr{L}(\pi_i) \mid W_{i-1} \notin \mathscr{L}(\pi_{i-1})]\right)^t\\ &\quad\leqslant \left(1 - \frac{1}{c} + \frac{\log(1/\delta)}{t}\sum_{i=1}^{t}\mathbf{E}[\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec W_i) \mid W_{i-1} \notin \mathscr{L}(\pi_{i-1})]\right)^t\\ &\quad= \left(1 - \frac{1}{c} + \frac{\log(1/\delta)}{t}\sum_{i=1}^{t}\frac{\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec W_i)}{\mathbf{P}[W_{i-1} \notin \mathscr{L}(\pi_{i-1})]}\right)^t\\ &\quad\leqslant \left(1 - \frac{1}{c} + \frac{\log(1/\delta)}{t\,\mathbf{P}[W_t \notin \mathscr{L}(\pi_t)]}\sum_{i=1}^{t}\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec W_i)\right)^t\\ &\quad\leqslant \left(1 - \frac{1}{c} + \frac{\log(1/\delta)}{t\,\mathbf{P}[W_t \notin \mathscr{L}(\pi_t)]}\cdot(1+\delta)^t I\right)^t,\end{aligned}$$

where the third, fourth, and last steps use convexity, Theorem 9, and (10), respectively, $c > 1$ being the absolute constant from Theorem 9. We have shown that

$$\begin{aligned}&\mathrm{TV}((X, Y, W_1W_2\ldots W_t\Pi_{t+1}), (X, Y, W_1W_2\ldots W_t))\\ &\qquad\leqslant \mathbf{P}[W_t \notin \mathscr{L}(\pi_t)]\\ &\qquad\leqslant \min_{0\leqslant p\leqslant 1}\left\{\left(1 - \frac{1}{c} + \frac{\log(1/\delta)}{tp}\cdot(1+\delta)^t I\right)^t + p\right\}\\ &\qquad\leqslant \left(1 - \frac{1}{c} + \frac{3\log(1/\delta)}{t\epsilon}\cdot(1+\delta)^t I\right)^t + \frac{\epsilon}{3},\end{aligned}$$

which along with (4) gives

$$\begin{aligned}&\mathrm{TV}((X, Y, \Pi), (X, Y, W_1W_2\ldots W_t))\\ &\qquad\leqslant \left(1 - \frac{1}{c} + \frac{3\log(1/\delta)}{t\epsilon}\cdot(1+\delta)^t I\right)^t + \frac{\epsilon}{3} + t\delta.\end{aligned} \tag{12}$$

We now examine the communication requirements. By Theorem 9, stages $1, 2, \ldots, t$ of the stochastic process have communication cost

$$\sum_{i=1}^{t}|M_i| \leqslant \sum_{i=1}^{t}C_i' + \sum_{i=1}^{t}C_i'' + ct\log\frac{1}{\delta},$$

where C_i', C_i'' are nonnegative random variables such that $\mathbf{P}[\sum_{i=1}^{t}C_i'' > 0] \leqslant t\delta$ and

$$\begin{aligned}\mathbf{E}\left[\sum_{i=1}^{t}C_i'\right] &\leqslant c\sum_{i=1}^{t}(\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec W_i) + \delta\,\mathbf{E}\,\mathbb{D}_{X,Y}^{\pi_i,\mu_i}(\prec\Pi_i))\\ &\leqslant c(1+2\delta)^t I.\end{aligned}$$

by (11). Applying Markov's inequality,

$$\mathbf{P}\left[\sum_{i=1}^{t}|M_i| \geqslant \frac{3}{\epsilon}\cdot c(1+2\delta)^t I + ct\log\frac{1}{\delta}\right]$$

$$\leqslant \mathbf{P}\left[\sum_{i=1}^{t} C_i' \geqslant \frac{3}{\epsilon}\cdot c(1+2\delta)^t I\right] + \mathbf{P}\left[\sum_{i=1}^{t} C_i'' > 0\right]$$
$$\leqslant \frac{\epsilon}{3} + t\delta. \tag{13}$$

E. Final Communication Protocol

Sections IV-A through IV-D suggest a natural communication protocol π' for simulating π. Specifically, Alice and Bob simulate the stochastic process on their given inputs, terminating the simulation as soon as they have completed T stages or exchanged

$$\frac{3}{\epsilon}\cdot c(1+2\delta)^T I + cT\log\frac{1}{\delta} \tag{14}$$

bits of communication (whichever occurs first). The communication transcript $(R_1, R_2, R_3, \ldots, M_1, M_2, M_3, \ldots)$ of this simulation fully determines all the other random variables in (1), which are never explicitly communicated. Let E be the event that during the first T stages of the stochastic process, the communication cost exceeds (14). Then π' simulates π with respect to μ with error

$$\mathrm{TV}((X,Y,\Pi),(X,Y,W_1W_2\ldots W_T)) + \mathbf{P}[E]$$
$$\leqslant \left(1 - \frac{1}{c} + \frac{3\log(1/\delta)}{T\epsilon}\cdot(1+\delta)^T I\right)^T + \frac{2\epsilon}{3} + 2T\delta \tag{15}$$

by (12) and (13). The communication cost (14) and the simulation error (15) are bounded by $O(\frac{I}{\epsilon}\log^2\frac{I}{\epsilon})$ and ϵ, respectively, for $\delta = \Theta(\frac{\epsilon}{I})^3$ and $T = \Theta(\frac{I}{\epsilon}\log\frac{I}{\epsilon})$. This completes the proof of Theorem 10.

Acknowledgments

This work was supported in part by the author's NSF CAREER award CCF-1149018 and Alfred P. Sloan Foundation Research Fellowship. The author is thankful to Pei Wu for stimulating discussions, and to Mark Braverman, Gillat Kol, Anup Rao, and Pei Wu for their feedback on an earlier version on this manuscript.

References

[1] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948.

[2] ——, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, October 1948.

[3] A. Chakrabarti, Y. Shi, A. Wirth, and A. C.-C. Yao, "Informational complexity and the direct sum problem for simultaneous message complexity," in *FOCS*, 2001, pp. 270–278.

[4] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar, "An information statistics approach to data stream and communication complexity," *J. Comput. Syst. Sci.*, vol. 68, no. 4, pp. 702–732, 2004.

[5] ——, "Information theory methods in communication complexity," in *CCC*, 2002, pp. 93–102.

[6] B. Barak, M. Braverman, X. Chen, and A. Rao, "How to compress interactive communication," *SIAM J. Comput.*, vol. 42, no. 3, pp. 1327–1363, 2013.

[7] A. C.-C. Yao, "Some complexity questions related to distributive computing," in *STOC*, 1979, pp. 209–213.

[8] R. Jain, J. Radhakrishnan, and P. Sen, "A direct sum theorem in communication complexity via message compression," in *ICALP*, 2003, pp. 300–315.

[9] M. Braverman and A. Rao, "Information equals amortized communication," *IEEE Trans. Information Theory*, vol. 60, no. 10, pp. 6058–6069, 2014.

[10] P. Harsha, R. Jain, D. A. McAllester, and J. Radhakrishnan, "The communication complexity of correlation," *IEEE Trans. Information Theory*, vol. 56, no. 1, pp. 438–449, 2010.

[11] M. Braverman, "Interactive information complexity," *SIAM J. Comput.*, vol. 44, no. 6, pp. 1698–1739, 2015.

[12] M. Braverman and O. Weinstein, "A discrepancy lower bound for information complexity," in *RANDOM*, 2012, pp. 459–470.

[13] J. Brody, H. Buhrman, M. Koucký, B. Loff, F. Speelman, and N. K. Vereshchagin, "Towards a reverse Newman's theorem in interactive information complexity," in *CCC*, 2013, pp. 24–33.

[14] B. Bauer, S. Moran, and A. Yehudayoff, "Internal compression of protocols to entropy," in *RANDOM*, 2015, pp. 481–496.

[15] S. N. Ramamoorthy and A. Rao, "How to compress asymmetric communication," in *CCC*, 2015, pp. 102–123.

[16] A. Ganor, G. Kol, and R. Raz, "Exponential separation of information and communication," in *FOCS*, 2014, pp. 176–185.

[17] ——, "Exponential separation of information and communication for Boolean functions," in *STOC*, 2015, pp. 557–566.

[18] ——, "Exponential separation of communication and external information," in *STOC*, 2016, pp. 977–986.

[19] M. Braverman, A. Rao, O. Weinstein, and A. Yehudayoff, "Direct products in communication complexity," in *FOCS*, 2013, pp. 746–755.

[20] G. Kol, "Interactive compression for product distributions," in *STOC*, 2016, pp. 987–998.

[21] A. A. Sherstov, "Compressing interactive communication under product distributions," in *Electronic Colloquium on Computational Complexity (ECCC)*, May 2016, report TR16-081.

[22] T. Holenstein, "Parallel repetition: Simplification and the no-signaling case," *Theory of Computing*, vol. 5, no. 1, pp. 141–172, 2009.

[23] R. Jain, P. Sen, and J. Radhakrishnan, "Optimal direct sum and privacy trade-off results for quantum and classical communication complexity," Available at http://arxiv.org/abs/0807.1267, 2008.