Internal Compression of Protocols to Entropy*

Balthazar Bauer[1], Shay Moran[2], and Amir Yehudayoff[3]

1 **Département d'Infomatique, ENS Lyon**
Lyon, France
balthazarbauer@aol.com
2 **Departments of Computer Science, Technion-IIT**
Haifa, Israel, and
Max Planck Institute for Informatics
Saarbrücken, Germany
shaymrn@cs.technion.ac.il
3 **Departments of Mathematics, Technion-IIT**
Haifa, Israel
amir.yehudayoff@gmail.com

Abstract

We study internal compression of communication protocols to their internal entropy, which is the entropy of the transcript from the players' perspective. We provide two internal compression schemes with error. One of a protocol of Feige et al. for finding the first difference between two strings. The second and main one is an internal compression with error $\varepsilon > 0$ of a protocol with internal entropy H^{int} and communication complexity C to a protocol with communication at most order $(H^{int}/\varepsilon)^2 \log(\log(C))$.

This immediately implies a similar compression to the internal information of public-coin protocols, which provides an exponential improvement over previously known public-coin compressions in the dependence on C. It further shows that in a recent protocol of Ganor, Kol and Raz, it is impossible to move the private randomness to be public without an exponential cost. To the best of our knowledge, No such example was previously known.

1998 ACM Subject Classification H.1.1 Coding and Information Theory: Data Compaction and Compression

Keywords and phrases Communication complexity, Information complexity, Compression, Simulation, Entropy

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2015.481

1 Introduction

The problem of compressing information and communication is fundamental and useful. This paper studies one shot compression of interactive communication (as opposed to amortized compression).

The basic scenario, the transmission problem, was studied by Fano and Shannon [12] and by Huffman [17]. In it, Alice wishes to transmit to Bob a message $u \in U$ with u that is distributed according to a known distribution μ over U. They proved that the above transmission can be optimally compressed in the sense that Alice may send u to Bob using roughly $\log(1/\mu(u))$ many bits on average, and conversely if Alice sends fewer

* This work was supported by the Taub foundation, by ISF, and by BSF.

18th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'15) / 19th Int'l Workshop on Randomization and Computation (RANDOM'15).
Editors: Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim; pp. 481–496

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

than $\log(1/\mu(u))$ bits on average then information is lost. In the transmission problem, the information flow is one-way, only Alice talks.

How about more complex communication protocols in which both sides are allowed to talk? The standard model for interactive communication was introduced by Yao [28]. Interactive communication, not surprisingly, allows for more efficient conversations than one-way ones. For example, the following lemma (which we also use later on) demonstrates the power of interaction (and of public randomness) in handling a variant of the transmission problem in which only Bob knows the distribution μ over U.

▶ **Lemma 1.1.** *Let U be a finite set, and $0 < \varepsilon < 1/2$. Assume Alice knows some $u_a \in U$ and that Bob knows a distribution μ on U which Alice does not know. Using public randomness, Alice and Bob can communicate at most $2\log(1/\mu(u_a)) + \log(1/\varepsilon) + 5$ bits, after which Bob outputs u_b so that $u_a = u_b$ with probability at least $1 - \varepsilon$.*

This lemma describes a one shot protocol (i.e. for a single instance) that enables transmission when Bob has some prior knowledge on Alice's input. A stronger version of this lemma was proved in [5] and also in [6], but since this lemma is sufficient for us and its proof is simpler than that of [5, 6] we provide its proof in Section A.3. A related result for the case when there is also an underlying distribution on Alice's input is the Slepian-Wolf theorem [26] which solves an amortized version of this problem. It is also related to the transmission problem considered by Harsha et al. [16] who studied the case that Alice knows μ and Bob wishes to sample from it.

Continuing recent works which we survey below, the main question we study is compression of interactive communication protocols. Compression of protocols, on a high level, means to simulate a given protocol π by a more efficient protocol σ in the sense that the communication complexity of σ is roughly the "information content" of π. It was shown to be strongly related to direct sum and product questions in randomized communication complexity [5, 2, 7].

We describe new compression schemes, and also provide a preliminary discussion of concepts and basic facts related to compression.

1.1 A preliminary discussion

In this section we provide intuitive definitions of important concepts. See Section 2 for formal definitions.

1.1.1 Computation and simulation

There is a distinction between external computation and internal computation [2, 7]. A protocol externally computes a function f if an external observer can deduce the value of f from the transcript, and a protocol internally computes f if the value of f may be privately obtained by Alice and Bob but not necessarily by an external observer (who only sees the transcript of the protocol but not the inputs). Note that for $f : X \times Y \to Z$ the difference between internal and external computation of f can be at most $\log|Z|$. Indeed, every protocol that internally computes f can be transformed to a protocol that externally computes f by adding one more message in which one of the parties sends the value of the function. Therefore, this distinction is only meaningful for large Z.

It is interesting that for deterministic protocols these two seemingly different notions coincide, so the strength of internal computation is evident only in randomized or distributional settings (the proof is given in Section A.1).

▶ **Proposition 1.2.** *Let $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. If π is a deterministic protocol that internally computes f then it also externally computes f.*

External and internal computations induce the corresponding types of simulations. Here we provide an intuitive meaning of the notion of simulation. In Section 2 we provide formal definitions, and discuss them in more detail. A protocol σ externally simulates the protocol π if an external observer who has access only to the public data (i.e. transcript and public randomness) of σ can deduce from it the public data of π. The protocol σ internally simulates π if each of Alice and Bob can obtain their private data of π from their private data of σ (i.e. transcript and private randomness).

As an example which illustrates the difference between internal and external simulation, consider the simple case when (x, y) are jointly distributed so that $x = y$, Alice knows x, Bob knows y and π is the protocol in which Alice sends x to Bob. In this case, it is clear that the empty protocol internally simulates π but every external simulation of π must in general use many bits. This example also demonstrates that Proposition 1.2 does not hold for promise problems, when the inputs are guaranteed to come from a non rectangular set.

1.1.2 Compression

To define compression, we should ask ourselves what is the "benchmark quantity" that we should strive to compress to. Shannon's source coding theorem [25] states that in the transmission problem (i.e., one-way communication), the entropy of the message is equal to the amortized communication of sending many independent messages. Braverman and Rao [5] analogously showed that the internal information (defined below) is equal to the amortized cost of making several independent conversations. Entropy and internal information are therefore two reasonable choices for "benchmark quantities". Below we survey several additional options.

1.1.3 Information complexities

The most studied measures in the context of protocol compression are information complexities. For every communication protocol π and every distribution μ on inputs, two versions of information have been defined: The internal information [1, 2] denoted $I^{int}_{\mu}(\pi)$ and the external one [9] denoted $I^{ext}_{\mu}(\pi)$ (see Section 2 for formal definitions). The intuitive semantic of internal information is the amount of information the communication transcript reveals to Alice and Bob about the inputs, and the intuitive semantic of external information is the amount of information the communication transcript reveals to an external observer about the inputs. It always holds that the internal information is at most the external one, which is at most the average communication complexity $\mathsf{CC}^{avg}_{\mu}(\pi)$ (see e.g. [2, 18]).

The following claim shows that information provides a lower bound for errorless simulations. This generalizes the basic fact that entropy provides a lower bound for errorless transmission. This claim seems to be known but we could not find an explicit reference to it so we provide a proof in Section A.2 (the special case of deterministic external simulation was proved in [23]).

▶ **Claim 1.3.** *Let π be a general protocol with input distribution μ.*

- *If σ simulates π externally without error then $\mathsf{CC}^{avg}_{\mu}(\sigma) \geq I^{ext}_{\mu}(\pi)$.*
- *If σ simulates π internally without error then $\mathsf{CC}^{avg}_{\mu}(\sigma) \geq I^{int}_{\mu}(\pi)$.*

Although $I^{int}_{\mu}(\pi) \leq I^{ext}_{\mu}(\pi)$, the second bullet in the claim above does not follow from the first, since not every internal simulation is an external simulation.

In the other direction, [2] provided two different compression schemes for general protocols. An external compression with error that uses roughly $I^{ext}_{\mu}(\pi)\log(CC(\pi))$ bits, and an internal compression with error that uses roughly $\sqrt{I^{int}_{\mu}(\pi)\cdot CC(\pi)}$ bits. A second internal compression with error that uses at most roughly $2^{I^{int}_{\mu}(\pi)}$ bits, regardless of $CC(\pi)$, appears in [3]. Later on, [8, 24] showed that the internal compression from [2] applied to public-coin protocols yields a much better compression with only order $I^{int}_{\mu}(\pi)\log(CC(\pi))$ bits. We discuss connections of these works to ours below.

1.1.4 Entropy complexities

We consider two additional complexity measures for compression:

The first one, which was studied in [11], is the external entropy $H^{ext}_{\mu}(\pi)$. Its semantic is how many bits are required for describing the transcript of π to an external observer. The second measure we consider is the internal entropy $H^{int}_{\mu}(\pi)$. Its semantic is the number of bits required in order to describe the transcript to Alice plus the number of bits required to describe the transcript to Bob (see Section 2 for formal definitions).

Some connections between the information measures and the entropy measures are provided in the following claim.

▶ **Claim 1.4.** *Let π be a protocol with input distribution μ. Then,*

$$H^{ext}_{\mu}(\pi) \geq I^{ext}_{\mu}(\pi) \quad \textit{and} \quad H^{int}_{\mu}(\pi) \geq I^{int}_{\mu}(\pi).$$

Moreover, if π does not have private randomness then

$$H^{ext}_{\mu}(\pi) = I^{ext}_{\mu}(\pi) \quad \textit{and} \quad H^{int}_{\mu}(\pi) = I^{int}_{\mu}(\pi).$$

As mentioned, in the case of one-way deterministic protocols, the external entropy fully captures the compression problem. The above claim combined with Claim 1.3 implies that, more generally, for public-coin protocols entropy provides a lower bound on errorless simulation. Interestingly, the authors of [11] proved that this lower bound is essentially tight. They gave an optimal external compression of general protocols[1]

▶ **Theorem 1.5** ([11])**.** *Every protocol π can be externally simulated without error by a protocol σ so that $\mathsf{CC}^{avg}_{\mu}(\sigma) \leq O(H^{ext}_{\mu}(\pi))$.*

1.1.5 With or without error

Another important distinction is between exact simulation and simulation with error.

A meaningful example already appears in the transmission problem, when there is a distribution μ on inputs x and Alice sends a (prefix free) encoding of x to Bob. Any exact solution to this problem requires expected communication of at least $H(\mu)$. However, if μ is highly concentrated on a point but with probability ε it is uniform on the remaining elements, an empty protocol simulates μ with ε error while the entropy is potentially huge. So entropy and information are not in general lower bounds for simulation with error, and the lower bounds from Claim 1.3 do not hold for simulation with error.

In the other direction, we have seen that entropy (or information) provides a lower bound on errorless simualtion. We shall see below that this lower bound is not tight, that is, there are protocols with small entropy that can not be efficiently simulated without error.

[1] They only considered deterministic protocols but their arguments can be generalized to general protocols.

1.2 Internal compression

1.2.1 Impossibility of errorless compression

Theorem 1.5 above provides errorless compression to external entropy. The main compression question is, however, whether a protocol can be internally simulated with communication that is close to its internal information. Motivation for studying this questions comes from direct sum and product questions in randomized communication complexity [5, 2, 7].

How about an errorless internal compression to internal entropy? It is long known that internal compression to internal entropy is not always possible [14, 22, 20, 3]. In [22], for example, Orlitsky studied zero-error compression of the transmission protocol in which Alice sends her input x to Bob who knows y, and constructed distributions on (x, y) such that every errorless internal simulation of the transmission protocol must communicate at least $H(x)$ bits, which is strictly larger than the internal entropy $H(x|y)$. Later on, Naor, Orlitskty and Shor [20] strengthened it to the amortized setting. A concrete statement (that can be proved e.g. using ideas from [3, 18]) is that for every n, there is a one round deterministic protocol π and input distribution μ so that $H^{int}_{\mu}(\pi) \leq 1$ and $CC(\pi) \leq n$ but if σ is an errorless internal simulation of π then $\mathsf{CC}^{avg}_{\mu}(\sigma) \geq n - 2$.

Our internal compression scheme and the ones from [2, 3] must therefore introduce errors.

1.2.2 Finding the first difference

Before stating our general compression scheme, we demonstrate its ideas by an internal compression of the *finding the first difference* problem, which lies at the heart of the internal compression schemes of [2, 8, 24]. Feige et al. [13] gave an optimal randomized protocol for this problem in terms of communication complexity (Viola [27] proved a matching lower bound).

▶ **Theorem 1.6** ([13]). *There is a public-coin protocol that on inputs $x, y \in \{0,1\}^n$ externally outputs the smallest index i in which x, y differ (or outputs "equal" if $x = y$) with probability at least $1 - \varepsilon$. The communication complexity of this protocol is at most $O(\log(n/\varepsilon))$.*

The protocol of Feige et al. externally solves the problem. The following Theorem provides an internal solution for this problem, which is more efficient when the internal information is small (the protocol is presented in Section 3).

▶ **Theorem 1.7.** *Let μ be a distribution on $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, and let $\varepsilon > 0$. Denote by $i = i(x, y)$ the smallest index in which x, y differ (or $i =$ "equal" if $x = y$). Denote $h^{int} = H(i|x) + H(i|y)$. There is a public-coin protocol and an event $\mathcal{E} \subset \{0,1\}^n \times \{0,1\}^n$ with probability $\mu(\mathcal{E}) < \varepsilon$ so that for all $(x, y) \notin \mathcal{E}$, the communication complexity of the protocol on input (x, y) is at most*[2]

$$O\left(\log\left(\frac{1}{\mu(i|x)\cdot\mu(i|y)}\right)\log\left(\log(n)h^{int}/\varepsilon\right)\right),$$

and it internally computes i with probability at least $1 - \varepsilon$. The overall communication complexity with error ε is at most

$$O\left(\frac{h^{int}}{\varepsilon}\log\left(\log(n)h^{int}/\varepsilon\right)\right).$$

[2] Here and below, for simplicity of notation, we write $\mu(i|y)$ to denote $\mu(\{(x, y) : i(x, y) = i\}|\{y\})$.

We state the theorem in this form since it hints at the core of its proof. To understand it better, it may be helpful to observe

$$H(i|x) + H(i|y) = I(i;y|x) + I(i;x|y) = \mathbb{E}_\mu \log\left(\frac{1}{\mu(i|x)\cdot\mu(i|y)}\right).$$

This protocol gives an improvement over that of [13] when the internal entropy is small. It highlights the importance of internal computation and may help to understand the more general compression below. It may also be useful in future internal compression schemes.

1.2.3 Main compression

We finally state our internal compression scheme (see Section 4 for its description). As mentioned above, such a compression must have positive error, even for one round protocols.

▶ **Theorem 1.8.** *Let μ be a distribution on $\mathcal{X}\times\mathcal{Y}$ and let $\varepsilon > 0$. Let π be a protocol with inputs from μ. Then, there is a public-coin protocol σ with communication complexity*

$$CC(\sigma) \le O\left(\frac{\left(H^{int}_\mu(\pi)\right)^2}{\varepsilon^2}\cdot\log(\log(CC(\pi)))\right)$$

that internally simulates π with error ε.

As noted earlier, if π is a protocol that uses no private randomness then the internal entropy of π is equal to the internal information of π. So, for public-coin protocols, Theorem 1.8 gives an internal compression in terms of internal information, which exponentially improves [8, 24] in terms of the dependence on $CC(\pi)$. It, therefore, also concerns the power of private randomness in saving information, which we now discuss.

1.2.4 Transferring private to public randomness

Every private-coin protocol can be simulated by a public-coin protocol with the same *communication* complexity. Conversely, Newman [21] proved that for communication complexity public randomness may be efficiently replaced by private one, when dealing with computation of relations (it however does not yield a simulation of public-coin protocols by private-coin protocols). In the information complexity context the situation is opposite, every public-coin protocol can be simulated by a private-coin protocol with the same *information* complexity. The authors of [8, 4] showed that for information complexity private randomness may be efficiently simulated by public one when the number of rounds is bounded. If any private-coin protocol could be simulated by a private-coin one without increasing the information and communication complexities, then to compress general protocol it would suffice to compress public-coin protocols.

Our compression shows limitations on moving private randomness to being public. A recent work of Ganor, Kol and Raz [15] shows that for every large enough $k\in\mathbb{N}$ there is a distribution μ and a private-coin protocol π_0 with internal information $O(k)$ so that every protocol that internally simulates π_0 with small error must communicate at least 2^k bits. This marks the first known separation between information and communication complexities. The protocol π_0 has communication complexity $O(k\cdot 2^{4^k})$ so that $\log(\log(CC(\pi_0)) = O(k)$. Together with our compression scheme, this means that there is no way to simulate π_0 using only public randomness without a cost; for example, every public-coin internal simulation of π_0 with near-optimal information complexity of $O(k)$ must communicate at least $2^{2^{2^{\Omega(k)}}}$ bits.

1.2.5 Discussion of the proof of Theorem 1.8

Compression to internal entropy, as mentioned above, must be done in an internal fashion. Namely, an observer of the conversation (who does not know the inputs nor the private randomness) should not be able to make much sense of it.

The only two compression schemes with this property that were previously known are from [2, 3]. The scheme from [3] is not efficient in terms of information complexity so we do not discuss it in detail here. In the scheme from [2] the players privately sample a candidate transcript, and they communicate to fix errors. After an error is located, the candidate transcript is modified until converging to the correct transcript. The errors are fixed using the protocol of Feige et al. for finding the first difference, and each error fixing costs about $\log(CC(\pi))$ bits.

The main problem in analyzing their protocol is bounding the number of errors in terms of the internal information. They are able to do so but the cost is quite high and the overall upper bound on the number of errors they show is order[3] $\sqrt{CC(\pi)I^{int}_{\mu}(\pi)}$. The authors of [8, 24] showed that for a deterministic protocol π, the expected number of errors in this scheme decreases[4] to roughly $I^{int}(\pi)$ which sums up to total communication of order $I^{int}(\pi)\log(CC(\pi))$ bits.

It is natural to consider a slight variation of this scheme in which the errors are fixed using our protocol from Theorem 1.7, instead of the protocol of Feige et al. However, it is not clear that this modified scheme yields the desired result. On a high level, this is because it may be the case that the additional information that is revealed from correcting the mistakes is large, and we do not know how to bound it by the internal information of the simulated protocol.

Our approach is different and starts with the compression of deterministic protocols to external entropy of [11]. The main idea there is that a deterministic protocol induces a distribution on the leaves of the protocol tree, and that there is always a vertex u in the tree with probability mass roughly $1/2$ (Lemma 2.1 below). Both players know u and they can check if the rectangle[5] it defines contains x and y with 2 bits of communication. It can easily be shown that by doing so they (roughly) learn one bit of information. This yields an optimal but external compression (an observer knows u as well).

In the internal case, there is no single node that is good for both players. Alice knows a node v_a and Bob a node v_b, which are in general arbitrary nodes in the protocol tree. The crux of our protocol is an efficient way for Alice and Bob to learn enough about v_a, v_b so that at least one of them obtains one bit of information. We show that using Theorem 1.6 one of them, say Alice, can identify a good vertex u to focus on (roughly, u is somewhere in between v_a, v_b). Using Lemma 1.1 Alice then tries to internally transmit u to Bob. If this transmission succeeds, then, with high probability, Bob learns one bit of information, and if this transmission fails then, with high probability, Alice learns one bit of information. The transmission is indeed internal in that an external observer does not in general learn u even when Bob does. The full protocol appears in Section 4.

[3] On a high level this cost occurs for the following reason: if we denote by $h(p)$ the entropy of a random bit with bias $p \in [0,1]$, then $h(\frac{1}{2}+\delta) - h(\frac{1}{2})$ is of order δ^2. The second power of δ yields the square root $CC(\pi)$ in the analysis.

[4] The improvement comes from that $h(\delta) - h(0)$ is of order δ.

[5] The set of inputs that reach u is a rectangle, that is, it is of the form $\mathcal{X}' \times \mathcal{Y}' \subset \mathcal{X} \times \mathcal{Y}$.

2 Definitions and preliminaries

Logarithms in this text are to the base two. We provide the basic definitions needed for this text. For background and more details on information theory see the book [10] and on communication complexity see the book [19].

2.1 Information theory

The entropy of a random variable X taking values in U is defined as

$$H(X) = \sum_{u \in U} \Pr[X = u] \log(1/\Pr[X = u]).$$

The entropy of X conditioned on Y is defined as $H(X|Y) = H(X,Y) - H(Y)$. The mutual information between X, Y conditioned on Z is defined as $I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$.

2.2 Protocols

A deterministic communication protocol π with inputs from $\mathcal{X} \times \mathcal{Y}$ is a rooted directed binary tree with the following structure. Edges are directed from root to leaves. Each internal node in the protocol is owned by either Alice or Bob. For every $x \in \mathcal{X}$, each internal node v owned by Alice is associated with an edge $e_{v,x}$ from v to one of the children of v. Similarly, for every $y \in \mathcal{Y}$, each internal node v owned by Bob is associated with an edge $e_{v,y}$. On input x, y, a protocol π is executed by starting at the root and following the unique path defined by $e_{v,x}, e_{v,y}$ until reaching a leaf. We denote by $T_\pi = T_\pi(x,y)$ the leaf reached, which we also call the transcript of π with input (x,y). The length of a transcript, denoted $|T_\pi|$, is the depth of the corresponding leaf.

In a public-coin protocol, Alice and Bob also have access to public randomness r that they both know. In a private-coin protocol, Alice has access to a random string r_a, and Bob has access to a random string r_b. A general protocol is a protocol which uses both public and private coins. The four random variables $(x,y), r, r_a, r_b$ are always assumed independent.

The communication complexity of a deterministic π, denoted by $\mathsf{CC}(\pi)$, is the maximum length of a transcript. For general protocols, $\mathsf{CC}(\pi)$ is defined as the maximum communication complexity over all randomness as well (i.e. over x, y, r, r_a, r_b), and $\mathsf{CC}^{avg}_\mu(\pi)$ is the expected length of a transcript over all randomness.

2.3 Computation

A deterministic protocol π externally computes a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ if there is a map M so that $f(x,y) = M(T_\pi(x,y))$ for all x, y. A deterministic protocol π internally computes a function f if there are two maps M_a, M_b so that $M_a(x, T_\pi(x,y)) = M_b(y, T_\pi(x,y)) = f(x,y)$ for all x, y. In the randomized setting, M may depends on r; M_a may depend on r, r_a; M_b may depend on r, r_b; and the equalities should hold with probability at least $1 - \varepsilon$ over the distribution of r, r_a, r_b for all x, y. In the distributional setting, the probability is taken over x, y as well.

2.4 Information and entropy of protocols

For a distribution μ on the inputs, define

$$I^{int}_\mu(\pi) = I(T_\pi; X|Y, R, R_b) + I(T_\pi; Y|X, R, R_a)$$

and

$$I^{ext}_{\mu}(\pi) = I(T_\pi; X, Y|R).$$

Similarly, define

$$H^{int}_{\mu}(\pi) = H(T_\pi|Y, R, R_b) + H(T_\pi|X, R, R_a)$$

and

$$H^{ext}_{\mu}(\pi) = H(T_\pi|R).$$

Note that each of these measures also induce a corresponding complexity measure for functions/relations in the standard way.

2.5 Simulation

Let π, σ be protocols, let μ be a distribution on the input space $\mathcal{X} \times \mathcal{Y}$ and let $\varepsilon \geq 0$.

Our goal is defining when σ simulates π with error ε in the distributional setting[6]. Namely, probabilities are taken over all randomness of inputs as well as private and public coins.

The randomness in σ is s, s_a, s_b, and the randomness in π is r, r_a, r_b. We say that σ externally simulates π with error ε if there exists a function $M = M(T_\sigma, s)$ so that the distribution of $(x, y, (T_\pi, r))$ is ε-close in L_1 distance to the distribution of $(x, y, M(T_\sigma, s))$.

We say that σ internally simulates π with error ε if there exist functions $M_a = M_a(T_\sigma, x, s_a, s)$ and $M_b = M_b(T_\sigma, y, s_b, s)$ so that the distribution of $(x, y, (T_\pi, r, r_a))$ is ε-close in L_1 distance to the distribution of (x, y, M_a), and the distribution of $(x, y, (T_\pi, r, r_b))$ is ε-close in L_1 distance to the distribution of (x, y, M_b).

The simulation we present in the proof of Theorem 1.8 is in fact of a stronger form. In the beginning of σ, Alice and Bob interpret the public randomness as a pair $s = (r, s')$ and their private randomness as $s_a = (r_a, s'_a)$ and $s_b = (r_b, s'_b)$. They think of r, r_a, r_b as the fixed randomness of π, and communicate in order to privately compute the fixed transcript $T_\pi = T_\pi(x, y, r, r_a, r_b)$, with error probability (over the remaining randomness s', s'_a, s'_b) of at most ε.

This stronger type of simulation is sometimes too strong to be useful, as the following example demonstrates. Consider a protocol in which x, r_a are uniform in $\{0,1\}^n$, and Alice just sends $x + r_a \in \{0,1\}^n$ to Bob. The transcript of this protocol is just a random noise, and its external information is indeed zero. It can, indeed, be externally simulated without error by a protocol with zero communication; interpret s as a uniform element in $\{0,1\}^n$ and set $M(\emptyset, s) = (s, \emptyset)$. However, every strong simulation of this protocol (as the one in Theorem 1.8 mentioned above) must communicate many bits. Indeed, the transcript of a strong simulation must reveal the value of $x + r_a$ to Bob, with high probability. This stronger type of simulation corresponds to internal entropy rather than internal information. In the example above, the internal information is 0 but the internal entropy is n.

2.6 Balanced nodes in trees

We use the following well known lemma (see e.g. [19]).

▶ **Lemma 2.1.** *Let μ be a probability measure on the leaves of a rooted binary tree. The distribution μ may be extended to a function on all nodes in the tree by setting $\mu(v)$ to be the μ-probability that a leaf that is a successor of v is chosen. Then, there exists a node u such that either u is a leaf and $\mu(u) \geq 2/3$, or $1/3 \leq \mu(u) \leq 2/3$.*

[6] There is also a natural variant of this definition in the randomized setting but it is not relevant for this text.

3 Finding the first difference

Proof of Theorem 1.7. Denote by $\mathcal{E}$ the (event) set of inputs (x, y) so that

$$\mu(i|x) \cdot \mu(i|y) < 2^{-2h^{int}/\varepsilon}.$$

By Markov's inequality,

$$\mu(\mathcal{E}) < \varepsilon/2.$$

For inputs in $\mathcal{E}$, the protocol may fail. For the rest of the proof, fix $(x, y) \notin \mathcal{E}$ and set $i = i(x, y)$.

The protocol proceeds in iterations indexed by $t \in \mathbb{N}$. For every t, Alice knows a distribution α_t on $[n] \cup \{\text{“}equal''\}$ and Bob a distribution β_t on $[n] \cup \{\text{“}equal''\}$ where we use the order $1 < 2 < \ldots < n < \text{“}equal''$. It may help to think of the distributions α_t and β_t as representing Alice and Bob's opinions for what is the first difference, given what they have learned upto iteration t. They start with

$$\alpha_0(j) = \Pr_{\mu}[i = j|x] \quad \text{and} \quad \beta_0(j) = \Pr_{\mu}[i = j|y], \quad \text{for all } j.$$

Iteration t starts with Alice knowing α_t and Bob knowing β_t, and ends with an update of these distributions to $\alpha_{t+1}, \beta_{t+1}$. There are $O(h^{int}/\varepsilon)$ iterations, and the probability of failure in each iteration is at most δ for $\delta = c\varepsilon^2/h^{int}$ for a small constant $c > 0$. The union bound implies that the overall error is at most ε.

The goal of every iteration is, given α_t, β_t, to construct with probability at least $1 - O(\delta)$ distributions $\alpha_{t+1}, \beta_{t+1}$ so that (if they did not stop)

$$\alpha_{t+1}(i) \geq \alpha_t(i) \ , \quad \beta_{t+1}(i) \geq \beta_t(i)$$

and

$$\alpha_{t+1}(i) \cdot \beta_{t+1}(i) \geq \frac{3}{2} \cdot \alpha_t(i) \cdot \beta_t(i).$$

This immediately implies that the number of iterations is at most $O\left(\log\left(\frac{1}{\mu(i|x)\cdot\mu(i|y)}\right)\right) = O(h^{int}/\varepsilon)$ since we conditioned on not $\mathcal{E}$ and since α_t, β_t are probability distributions so their maximum value is at most 1.

The protocol uses the following subroutine we call *check*(j) with error δ. It gets as input $j \in [n] \cup \{\text{“}equal''\}$ and with communication $O(\log(1/\delta))$ it externally outputs “yes” if $j = i$ and “no” if $j \neq i$. This subroutine just uses public randomness[7] to check if $x_{<j} = y_{<j}$ and $x_j \neq y_j$ for $j \in [n]$ or if $x = y$ for $j = \text{“}equal''$.

Iteration t is performed as follows:

1. Let d_a be the maximum integer so that $\alpha_t(\{1, 2, \ldots, d_a - 1\}) < 2/3$ and let d_b be the maximum integer so that $\beta_t(\{1, 2, \ldots, d_b - 1\}) < 2/3$. Alice knows d_a and Bob d_b. Using the protocol from Theorem 1.6, with communication $O(\log(\log(n)/\delta))$ the players find[8] d that is between d_a, d_b with error at most δ.
2. If $\alpha_t(d) > 1/3$ then the players check(d) with error δ. If the answer is “yes” then they stop and output d.
   If the answer is “no” then they update α_t, β_t to $\alpha_{t+1}, \beta_{t+1}$ by conditioning on the event $([n] \cup \text{“}equal'') \setminus \{d\}$ and continue to the next iteration.

[7] For example, using the standard randomized protocol for equality [19].
[8] If we represent d_a, d_b as binary strings of length order $\log(n)$ then to find d it suffices to find the first index in which d_a, d_b differ.

3. If $\beta_t(d) > 1/3$ then the players check(d) with error δ. If the answer is "yes" then they stop and output d. If the answer is "no" then they update α_t, β_t to $\alpha_{t+1}, \beta_{t+1}$ by conditioning on the event $([n] \cup \text{"}equal''\text{)} \setminus \{d\}$ and continue to the next iteration.
4. The players check using public randomness with error δ if $x_{<d} = y_{<d}$.
   If the answer is "yes" then they update α_t, β_t to $\alpha_{t+1}, \beta_{t+1}$ by conditioning on the event $\{d, d+1, \ldots, n\} \cup \{\text{"}equal''\}$ and continue to the next iteration.
   If the answer is "no" then they update α_t, β_t to $\alpha_{t+1}, \beta_{t+1}$ by conditioning on the event $\{1, 2, \ldots, d-1\}$ and continue to the next iteration.

We analyse the correctness step by step assuming that no error occurred (we have already bounded the probability of error):

1. The players found d that is between d_a, d_b.
2. If $\alpha_t(d) > 1/3$ and the players output d then indeed the output is correct. If $\alpha_t(d) > 1/3$ and the players do not output d then $d \neq i$ which means that
$$\alpha_{t+1}(i) = \frac{\alpha_t(i)}{1 - \alpha_t(d)} > \frac{\alpha_t(i)}{2/3}$$
and $\beta_{t+1}(i) \geq \beta_t(i)$.
3. As in previous case.
4. If the players reached here then $\alpha_t(d), \beta_t(d) \leq 1/3$. Assume without loss of generality that $d_a \leq d_b$. The proof in the other case is similar.
   If $x_{<d} = y_{<d}$ then $i \geq d \geq d_a$. This implies that $\beta_{t+1}(i) \geq \beta_t(i)$. By choice,
$$\alpha_t(\{d, d+1, \ldots, n\}) = \alpha_t(d) + 1 - \alpha_t(\{1, \ldots, d\}) \leq \frac{1}{3} + \frac{1}{3} \leq \frac{2}{3},$$
   which implies $\alpha_{t+1}(i) \geq 3\alpha_t(i)/2$.
   If $x_{<d} \neq y_{<d}$ then $i < d \leq d_b$. This implies that $\alpha_{t+1}(i) \geq \alpha_t(i)$. By choice,
$$\beta_t(\{1, 2, \ldots, d-1\}) \leq \frac{2}{3},$$
   which implies $\beta_{t+1}(i) \geq 3\beta_t(i)/2$. ◀

4 Internal compression

Proof of Theorem 1.8. Let x, y be the inputs to π, let r be the public randomness, and let r_a, r_b be the private randomness. The first observation is that
$$H^{int} = H^{int}_{\mu}(\pi) = \mathbb{E}_{x,y,r,r_a,r_b} \log \left(\frac{1}{\mu(T_\pi | x, r, r_a) \cdot \mu(T_\pi | y, r, r_b)} \right),$$
where here $T_\pi = T_\pi(x, y, r, r_a, r_b)$. Denote by $\mathcal{E}$ the event (i.e. set of (x, y, r, r_a, r_b)) that
$$\mu(T_\pi | x, r, r_a) \cdot \mu(T_\pi | y, r, r_b) < 2^{-2H^{int}/\varepsilon}.$$
By Markov's inequality,
$$\Pr(\mathcal{E}) < \varepsilon/2.$$
When $\mathcal{E}$ occurs, the protocol σ may fail. For the rest of the proof, fix $(x, y, r, r_a, r_b) \notin \mathcal{E}$ and set $T_\pi = T_\pi(x, y, r, r_a, r_b)$.

The protocol σ proceeds in iterations indexed by $t \in \mathbb{N}$. The starting point of every iteration is a distribution α_t on leaves of π that Alice knows and a distribution β_t on the

leaves of π that Bob knows. These distributions reflect the current perspective of the players after the communication so far. The first distributions are

$$\alpha_0(v) = \Pr[v|x, r, r_a] \text{ and } \beta_0(v) = \Pr[v|y, r, r_b]$$

for all leaves v of the protocol tree (the probability in α_0 for example is over Bob's randomness). The goal of every iteration is to construct with probability at least $1-\delta$ distributions $\alpha_{t+1}, \beta_{t+1}$ so that

$$\alpha_{t+1}(T_\pi) \geq \alpha_t(T_\pi) \ , \quad \beta_{t+1}(T_\pi) \geq \beta_t(T_\pi)$$

and

$$\alpha_{t+1}(T_\pi) \cdot \beta_{t+1}(T_\pi) \geq \frac{3}{2} \cdot \alpha_t(T_\pi) \cdot \beta_t(T_\pi).$$

The number of iterations is set to be at most

$$O(\log(2^{2H^{int}/\varepsilon})) = O(H^{int}/\varepsilon),$$

and the communication complexity of each iteration is at most

$$O\left(\log\left(\frac{\log(CC(\pi))}{\delta}\right) + \frac{H^{int}}{\varepsilon} + \log(1/\delta)\right).$$

Thus, setting $\delta = c\varepsilon^2/H^{int}$ for some small constant $c > 0$, the union bound implies the overall bound on the error.

Here is how iteration t is performed:

1. Alice finds a vertex v_a promised by Lemma 2.1 with α_t, and Bob finds v_b promised by Lemma 2.1 with β_t. Denote $d_a = depth(v_a)$ and $d_b = depth(v_b)$.
2. Using the protocol from Lemma 1.6, with communication $O(\log(\log(CC(\pi))/\delta))$ the players find[9] d that is between d_a, d_b with error $\delta/2$.
3. If $d_a \geq d_b$, the players do the following: Let u be the ancestor of v_a at depth d and let U be the set of nodes of depth d of π . Using the protocol from Lemma 1.1 Alice sends u to Bob. They use this protocol with error parameter $\delta/2$, where Alice's input is u and Bob's input is the distribution β_t induced on U.
   If this stage takes more than $O((H^{int}/\varepsilon) + \log(1/\delta))$ bits, then the players abort.
   At the end of this stage, either Bob thinks[10] he knows u as well or they have aborted.
   - If Bob thinks he knows u there are two options:
     If u is a leaf then the players stop and internally output u.
     Otherwise, the players set $\alpha_{t+1} = \alpha_t$ and β_{t+1} to be the distribution β_t conditioned on passing through u.
   - Otherwise, the players aborted and they set $\beta_{t+1} = \beta_t$ and α_{t+1} to be the distribution α_t conditioned on not passing through u.
4. When $d_a < d_b$, the players exchange roles.

We now analyse the performance in iteration t. For this, we assume that no error occurred. That is, that the protocols from Theorem 1.6 and Lemma 1.1 gave the desired result (this happens with probability at least $1 - \delta$). The analysis follows the outline of the protocol:

---

[9] Represent d_a, d_b as binary strings of length roughly $\log(CC)$.
[10] There is some small probability that Bob holds some $u' \neq u$ but he still thinks he knows u.

1. Lemma 2.1 says that there are always such nodes v_a, v_b.
2. the players find d that is between d_a, d_b.
3. We distinguish between two cases:

   Bob thinks he knows u: This means that $\beta_t(u) > 0$ and so (y, r_b) is in the rectangle defined by u. Thus, $((x, r_a), (y, r_b))$ is in the rectangle defined[11] by u, which implies that T_π is a successor of u.
   If u is a leaf then indeed $T_\pi = u$.
   Otherwise, there are two cases:
   The first is when v_b is an ancestor of u. In this case, v_b is not a leaf, $\beta_t(v_b) \geq \beta_t(u)$ and

   $$\beta_{t+1}(T_\pi) = \frac{\beta_t(T_\pi)}{\beta_t(u)} \geq \frac{\beta_t(T_\pi)}{\beta_t(v_b)} \geq \frac{\beta_t(T_\pi)}{2/3}.$$

   The second is when v_b is not an ancestor of u. In this case, $\beta_t(u) \leq 1 - \beta_t(v_b) \leq 2/3$ and

   $$\beta_{t+1}(T_\pi) = \frac{\beta_t(T_\pi)}{\beta_t(u)} \geq \frac{\beta_t(T_\pi)}{2/3}.$$

   Bob does not think he knows u: Since we assumed $\mathcal{E}$ does not occur, if u is an ancestor of T_π then

   $$\beta_t(u) \geq \beta_t(T_\pi) \geq \beta_0(T_\pi) \geq 2^{-2H^{int}/\varepsilon}.$$

   Since the players aborted (we ignore possibility of error), this means that u is not an ancestor of T_π. Since u is an ancestor of v_a, $\alpha_t(u) \geq \alpha_t(v_a) \geq 1/3$. Thus, by choice,

   $$\alpha_{t+1}(T_\pi) = \frac{\alpha_t(T_\pi)}{1 - \alpha_t(u)} \geq \frac{\alpha_t(T_\pi)}{1 - \alpha_t(v_a)} \geq \frac{\alpha_t(T_\pi)}{2/3}.$$

4. When $d_a < d_b$, the proof is similar. ◀

Acknowledgments. We thank Anup Rao for helpful conversations. We also thank Makrand Sinha and anonymous referees for comments on an earlier version of this text.

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. *SIAM J. Comput.*, 42(3):1327–1363, 2013.
3 Mark Braverman. Interactive information complexity. In *STOC*, pages 505–524, 2012.
4 Mark Braverman and Ankit Garg. Public vs private coin in bounded-round information. In *ICALP (1)*, pages 502–513, 2014.
5 Mark Braverman and Anup Rao. Information equals amortized communication. In *FOCS*, pages 748–757, 2011.
6 Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct product via round-preserving compression. In *ICALP (1)*, pages 232–243, 2013.

[11] For any fixed public string r, the set of all $(x, r_a), (y, r_b)$ for which $T_\pi(x, r_a, y, r_b)$ is a descendent of u with a positive probability is a rectangle.

7 Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct products in communication complexity. In *FOCS*, pages 746–755, 2013.

8 Joshua Brody, Harry Buhrman, Michal Koucký, Bruno Loff, Florian Speelman, and Nikolay K. Vereshchagin. Towards a reverse newman's theorem in interactive information complexity. In *IEEE Conference on Computational Complexity*, pages 24–33, 2013.

9 Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, , and Andrew Yao. Informational complexity and the direct sum problem for simultaneous message complexity. *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 270–278, 2001.

10 Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Interscience, 2006.

11 Martin Dietzfelbinger and Henning Wunderlich. A characterization of average case communication complexity. *Inf. Process. Lett.*, 101(6):245–249, 2007.

12 R. M. Fano. The transmission of information. Technical Report 65, Research Laboratory for Electronics, MIT, Cambridge, MA, USA, 1949.

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

14 Abbas El Gamal and Alon Orlitsky. Interactive data compression. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 0:100–108, 1984.

15 Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication. *Electronic Colloquium on Computational Complexity*, 2014.

16 Prahladh Harsha, Rahul Jain, David A. McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. *IEEE Transactions on Information Theory*, 56(1):438–449, 2010.

17 David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, September 1952.

18 Gillat Kol, Shay Moran, Amir Shpilka, and Amir Yehudayoff. Direct sum fails for zero error average communication. In *ITCS*, pages 517–522, 2014.

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

20 Alon Orlitsky Moni Naor and Peter Shor. Three results on interactive communication. *Information Theory, IEEE Transactions on*, 39(5):1608–1615, Sep 1993.

21 Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

22 Alon Orlitsky. Average-case interactive communication. *Information Theory, IEEE Transactions on*, 38(5):1534–1547, Sep 1992.

23 Alon Orlitsky and Abbas El Gamal. Average and randomized communication complexity. *IEEE Transactions on Information Theory*, 36(1):3–16, 1990.

24 Denis Pankratov. *Direct sum questions in classical communication complexity*. PhD thesis, University of Chicago, 2012.

25 C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

26 David Slepian and Rahul Jack K. Wolf. Noiseless coding of correlate information sources. *IEEE Transactions on Information Theory*, 19(4), July 1973.

27 Emanuele Viola. The communication complexity of addition. In *SODA*, pages 632–651, 2013.

28 Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC*, pages 209–213, 1979.

A Appendix

A.1 Internal deterministic computation is also external

Proof of Proposition 1.2. There are maps M_a, M_b so that for all x, y,

$$M_a(x, T_\pi(x,y)) = M_b(y, T_\pi(x,y)) = f(x,y).$$

Fix some rectangle $\rho = \{(x,y) : T_\pi(x,y) = T_\pi(x_0,y_0)\}$. For every $(x,y) \in \rho$, we know $M_a(x,\rho) = f(x,y) = f(x,y_0)$, and similarly $M_b(y_0,\rho) = f(x,y_0) = f(x_0,y_0)$. Therefore, f is constant on ρ and we can define $M(\rho) = f(x_0,y_0)$. ◀

A.2 Information lower bounds errorless simulation

Proof of Claim 1.3. The external case: Let σ be a protocol that externally simulated π without error. By definition of simulation, there exists a function M so that for all (x,y) so that $\mu(x,y) > 0$, it holds that $p_\sigma = p_\pi$, where p_σ is the distribution of $M(T_\sigma, s)$ and p_π is that of (T_π, r). Thus,

$$\begin{aligned}
\mathsf{CC}^{avg}_\mu(\sigma) &\geq I^{ext}_\mu(\sigma) && \text{(see e.g. [18])}\\
&= I(T_\sigma; X, Y | S) \\
&= I(T_\sigma, S; X, Y) && (S \text{ is independent of } (X,Y))\\
&\geq I(M(T_\sigma, S); X, Y) && \text{(data processing inequality)}\\
&= I(T_\pi, R; X, Y) && \text{(errorless simulation)}\\
&= I(T_\pi; X, Y | R) && (R \text{ is independent of } (X,Y))\\
&= I^{ext}_\mu(\pi).
\end{aligned}$$

The internal case: similarly to the external case,

$$\begin{aligned}
\mathsf{CC}^{avg}_\mu(\sigma) &\geq I^{ext}_\mu(\sigma)\\
&\geq I^{int}_\mu(\sigma)\\
&= I(T_\sigma, S, S_b; X | Y) + I(T_\sigma, S, S_a; Y | X)\\
&\geq I(M_b(T_\sigma, Y, S, S_b); X | Y) + I(M_a(T_\sigma, X, S, S_a); Y | X)\\
&= I(T_\pi, R, R_b; X | Y) + I(T_\pi, R, R_a; Y | X)\\
&= I^{int}_\mu(\pi).
\end{aligned}$$

◀

A.3 Transmission

Proof of Lemma 1.1. The players interpret the public randomness as boolean random hash functions on U. The protocol proceeds in iterations indexed by $t \in \mathbb{N}$. In iteration $t = 0$, the following is performed:

1. Alice sends $k = \lceil \log(1/\varepsilon) \rceil + 2$ hash values of u_a to Bob.
2. Bob computes the set
$$S_0 = \{u \in U : \mu(u) \in (1/2, 1]\}.$$
He compares every element of S_0 to the k hash values he received. He deletes every $s \in S_0$ that does not agree with at least one of these k hash values. Denote by S_0' the set S_0 after this deletion.

If S'_0 is empty, he sends a "0" to Alice.
If S'_0 is not empty, he sets u_b as an arbitrary element S'_0, and sends "1" to Alice, and the players stop.

For every $t = 1, 2, \ldots$, the following is performed (until the players stop):

1. Alice sends 2 new hash values of u_a to Bob.
2. Bob computes the set
$$S_t = \{u \in U : \mu(u) \in (2^{-t-1}, 2^{-t}]\}.$$
He compares every element of S_t to the $k + 2t$ hash values he received so far. He deletes every $s \in S_t$ that does not agree with at least one of these hash values. Denote by S'_t the set S_t after this deletion.
If S'_t is empty, he sends a "0" to Alice.
If S'_t is not empty, he sets u_b as an arbitrary element in S'_t, and sends "1" to Alice, and the players stop.

We now analyse the protocol. Let t_0 be so that $u_a \in S_{t_0}$. First, the protocol stops after at most $t_0 \leq \log(1/\mu(u_a)) + 1$ iterations, because u_a agrees with all hash values sent. Second, for every t, by the union bound,

$$\Pr[S'_t \neq \{u_a\} \cap S_t] \leq 2^{-(k+2t)} 2^{t+1} \leq 2^{-\log(1/\varepsilon)-t-1} = \frac{\varepsilon}{2^{t+1}}.$$

Thus, by the union bound, the probability that either there is some $t < t_0$ for which $S'_t \neq \emptyset$ or $S'_{t_0} \neq \{u_a\}$ is at most $\sum_{t=0}^{\infty} \varepsilon/2^{t+1} \leq \varepsilon$. ◀