Towards a Reverse Newman's Theorem in Interactive Information Complexity

Joshua Brody*[1], Harry Buhrman[2,3], Michal Koucký†[4], Bruno Loff‡[2], Florian Speelman§[2], and Nikolay Vereshchagin¶[5]

[1]Aarhus University
[2]CWI, Amsterdam
[3]University of Amsterdam
[4]Mathematical Institute of Czech Academy of Sciences, Prague
[5]Moscow State University

[joshua.e.brody,bruno.loff]@gmail.com, [buhrman,f.speelman]@cwi.nl
koucky@math.cas.cz, ver@mccme.ru

March 13, 2013

Abstract

Newman's theorem states that we can take any public-coin communication protocol and convert it into one that uses only private randomness with only a little increase in communication complexity. We consider a reversed scenario in the context of information complexity: can we take a protocol that uses private randomness and convert it into one that only uses public randomness while preserving the information revealed to each player?

We prove that the answer is yes, at least for protocols that use a bounded number of rounds. As an application, we prove new direct sum theorems through the compression of interactive communication in the bounded-round setting. Furthermore, we show that if a Reverse Newman's Theorem can be proven in full generality, then full compression of interactive communication and fully-general direct-sum theorems will result.

*Supported in part by Danish Council for Independent Research grant LOBO 438146. The author also acknowledges support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part of this work was performed.

†Supported in part by grant IAA100190902 of GA AV ČR, the Center of Excellence CE-ITI (P202/12/G061 of GA ČR) and RVO: 67985840.

‡Supported by grant SFRH/BD/43169/2008, given by FCT, Portugal.

§Supported by the NWO DIAMANT project.

¶The work was in part supported by th e RFBR grant 12-01-00864 and the ANR grant ProjetANR-08-EMER-008.

1 Introduction

Information cost was introduced by a series of papers [CSWY01, BYJKS04, JRS03, BBCR10, BR11] as a complexity measure for two-player protocols. Internal information cost measures the amount of information that each player learns about the input of the other player while executing a given protocol. In the usual setting of communication complexity we have two players, Alice and Bob, each having an input x and y, respectively. Their goal is to determine the value $f(x, y)$ for some predetermined function f. They achieve the goal by communicating to each other some amount of information about their inputs according to some *protocol*.

The usual measure considered in this setting is the number of bits exchanged by Alice and Bob, whereas the internal information cost measures the amount of information transferred between the players during the communication. Clearly, the amount of information is upper bounded by the number of bits exchanged but not vice versa. There might be a lengthy protocol (say even of exponential size) that reveals very little information about the players' inputs.

In recent years, a substantial research effort was devoted to proving the converse relationship between the information cost and the length of protocols, i.e., to proving that each protocol can be simulated by another protocol that communicates only the number of bits corresponding to the information cost of the original protocol. Such results are known as *compression theorems*. [BBCR10] prove that a protocol that communicates C bits and has internal information cost I can be replaced by another protocol that communicates $O(\sqrt{I \cdot C})$ bits. For the case when the inputs of Alice and Bob are sampled from independent distributions they also obtain a protocol that communicates $O(I \cdot \log C)$ bits. These conversions do not preserve the number of rounds. In a follow up paper [BR11] consider a bounded round setting and give a technique that converts the original q-round protocol into a protocol with $O(q \cdot \log I)$ rounds that communicates $O(I + q \log \frac{q}{\varepsilon})$ bits with additional error ε.

All known compression theorems are in the randomized setting. We distinguish two types of randomness — *public* and *private*. Public random bits are seen by both communicating players, and both players can take actions based on these bits. Private random bits are seen only by one of the parties, either Alice or Bob. We use *public-coin* (*private-coin*) to denote protocols that use only public (private) randomness. If a protocol uses both public and private randomness, we call it a *mixed-coin* protocol.

Simulating a private-coin protocol using public randomness is straightforward: Alice views a part of the public random bits as her private random bits, Bob does the same using some other portion of the public bits, and they communicate according to the original private-coin protocol. This new protocol communicates the same number of bits as the original protocol and computes the same function. In the other direction, an efficient simulation of a public-coin protocol using private randomness is provided by Newman's Theorem [New91]. Sending over Alice's private random bits to make them public could in general be costly as they may need say polynomially many public random bits, but Newman showed that it suffices for Alice to transfer only $O(\log n + \log \frac{1}{\delta})$ random bits to be able to simulate the original public-coin protocol, up to an additional error of δ.

In the setting of information cost the situation is quite the opposite. Simulating public

randomness by private randomness is straightforward: one of the players sends a part of his private random bits to the other player and then they run the original protocol using these bits as the public randomness. Since the random bits contain no information about either input, this simulation reveals no additional information about the inputs; thus the information cost of the protocol stays the same. This is despite the fact that the new protocol may communicate many more bits than the original one.

However, the conversion of a private-randomness protocol into a public-randomness protocol seems significantly harder. For instance, consider a protocol in which in the first round Alice sends to Bob her input x bit-wise XOR-ed with her private randomness. Such a message does not reveal any information to Bob about Alice's input — as from Bob's perspective he observes a random string — but were Alice to reveal her private randomness to Bob, he would learn her complete input x. This illustrates the difficulty in converting private randomness into public.

We will generally call "Reverse Newman's Theorem" (R.N.T.) a result that makes randomness public in an interactive protocol without revealing more information. This paper is devoted to attacking the following:

> **R.N.T. Question.** *Can we take a private-coin protocol with information cost I and convert it into a public-coin protocol with the same behavior and information cost $\tilde{O}(I)$?*

Interestingly, the known compression theorems [BBCR10, BR11, JPY12] give compressed protocols that use only public randomness, and hence as a by-product they give a conversion of private-randomness protocols into public-randomness equivalents. However, the parameters of this conversion are far from the desired ones.[1] In Section 6 we show that the R.N.T. question represents the core difficulty in proving full compression theorems; namely, we will prove that any public-coin protocol that reveals I bits of information can already be compressed to a protocol that uses $\tilde{O}(I)$ bits of communication, and hence a fully general R.N.T. would result in fully general compression results, together with the direct-sum results that would follow as a consequence. This was discovered independently by Denis Pankratov, whom in his MsC thesis [Pan12] extended the analysis of the [BBCR10] compression schemes to show that they achieve full compression in the case when only public randomness is used. Our compression scheme is similar but slightly different: we discovered it originally while studying the compression problem in a Kolmogorov complexity setting (as in [BKV08]), and our proof for the Shannon setting arises from the proper "translation" of this proof; we include it for completeness and because we think it makes for a more elementary proof.

Main contributions. Our main contribution is a Reverse Newman's Theorem in the bounded-round scenario. We will show that any q-round private-coin protocol can be converted to an $O(q)$-round public-coin protocol that reveals only additional $\tilde{O}(q)$ bits of information (Theorem 3.1). Our techniques are new and interesting. Our main technical tool is a conversion of one round private-randomness protocols into one round public-randomness protocols. This conversion proceeds in two main steps. After *discretizing* the protocol so that the private randomness is sampled uniformly from some finite domain, we

[1] We discuss the differences in more detail in Section 7.

convert the protocol into what we call a 1-1 protocol, which is a protocol having the property that for each input and each message there is at most one choice of private random bits that will lead the players to send that message. We show that such a conversion can be done without revealing too much extra information. In the second step we take any 1-1 protocol and convert it into a public-coin protocol while leaking only a small additional amount of information about the input. This part relies on constructing special bipartite graphs that contain a large matching between the right partition and any large subset of left vertices.

Furthermore, we will prove two compression results for public-randomness protocols: a round-preserving compression scheme to be used in the bounded-round case, and a general (not round-preserving) compression scheme which can be used with a fully general R.N.T. Either of these protocols achieves much better parameters than those currently available for general protocols (that make use of private randomness as well as public). The round-preserving compression scheme is essentially a constant-round average-case one-shot version of the Sleepian–Wolf coding theorem, and is interesting in its own right.

As a result of our R.N.T. and our round-preserving compression scheme, we will get a new compression result for general (mixed-coin) bounded-round protocols. Whereas previous results for the bounded-round scenario [BR11] gave compression schemes with communication complexity similar to our own result, their protocols were not round-preserving. We prove that a q-round protocol that communicates C bits and reveals I bits of information can be compressed to an $O(q)$-round protocol that communicates $O(I + q\log(\frac{qnC}{\delta}))$ bits, with additional error δ. As a consequence we will also improve the bounded-round direct-sum theorem of [BR11].

Organization of the paper. In Section 3 we discuss our Reverse Newman's Theorem. In Section 4 we show the conversion of a general one-round protocol to a 1-1 protocol, and in Section 5 we provide the conversion of a private-randomness 1-1 protocol into a public-randomness protocol. In Section 6 we will prove our compression results. Finally, Section 7 will be devoted to applications to direct-sum theorems.

2 Preliminaries

We use capital letters to denote random variables, calligraphic letters to denote sets, and lower-case letters to denote elements in the corresponding sets. So typically A is a random variable distributed over the set $\mathcal{A}$, and a is an element of $\mathcal{A}$. We will also use capital and lower-case letters to denote integers numbering or indexing certain sequences. We use $\Delta(A, A')$ to denote the statistical distance between random variables A and A'. We assume the reader is familiar with the basics of communication complexity and information theory. In case this is not so, we include basic facts and definitions in Appendix A.

We will be dealing with protocols that have both public and private randomness; this is not very common, so we will give the full definitions, which are essentially those of [BBCR10, BR11]. We will also be working exclusively in the distributional setting, and our compression and direct theorem results will follow also in the usual (worst-case randomized) setting, with roughly the same parameters, by the use of Yao's Principle in its Information Complexity variants [Bra12] (these details will be left for the full version of the paper). So from here onwards, we will assume that the input is given to two players,

Alice and Bob, by way of two random variables X, Y sampled from a possibly correlated distribution μ over the support $\mathcal{X} \times \mathcal{Y}$.

A *private-coin protocol* π with output set $\mathcal{Z}$ is defined as a rooted tree, called the *protocol tree*, in the following way:

1. Each non-leaf node is owned by either Alice or Bob.
2. If v is a non-leaf node belonging to Alice, then:
   (a) The children of v are owned by Bob and form a set $\mathcal{C}(v)$ and each element of $\mathcal{C}(v)$ is uniquely labeled with a binary string;
   (b) Associated with v is a set $\mathcal{R}_v$, and a function $M_v : \mathcal{X} \times \mathcal{R}_v \to \mathcal{C}(v)$.
3. The situation is analogous for Bob's nodes.
4. With each leaf we associate an *output value* in $\mathcal{Z}$.

On input x, y the protocol is executed as follows:

1. Set v to be the root of the protocol tree.
2. If v is a leaf, the protocol ends and outputs the value associated with v.
3. If v is owned by Alice, she picks a string r uniformly at random from $\mathcal{R}_v$ and sends the label of $M_v(x, r)$ to Bob, they both set $v := M_v(x, r)$, and return to the previous step. Bob proceeds analogously on the nodes he owns.

A general, or *mixed-coin*, protocol is given by a distribution over private-coin protocols. The players run such a protocol by using shared randomness to pick an index r (independently of X and Y) and then executing the corresponding private-coin protocol π_r. A protocol is called *public-coin* if every $\mathcal{R}_v$ has size 1, i.e., no private randomness is used.

We let $\pi(x, y, r, r^{(a)}, r^{(b)})$ denote the messages exchanged during the execution of π, for given inputs x, y, and random choices $r, r^{(a)}$ and $r^{(b)}$, and $\textsc{Out}_\pi(x, y, r, r^{(a)}, r^{(b)})$ be the output of π for said execution. The random variable R is the public randomness, $R^{(a)}$ is Alice's private randomness, and $R^{(b)}$ is Bob's private randomness; we use Π to denote the random variable $\pi(X, Y, R, R^{(a)}, R^{(b)})$.

Definition 1. *The* worst-case communication complexity *of a protocol π, $\mathsf{CC}(\pi)$, is the maximum number of bits that can be transmitted in a run of π on any given input and choice of random strings. The* average communication complexity *of a protocol π, with respect to the input distribution μ, denoted $\mathsf{ACC}_\mu(\pi)$, is the average number of bits that are transmitted in an execution of π, for inputs drawn from μ. The* worst-case number of rounds *of π, $\mathsf{RC}(\pi)$, is the maximum depth reached in the protocol tree by a run of π on any given input. The* average number of rounds *of π, w.r.t. μ, denoted $\mathsf{ARC}_\mu(\pi)$, is the average depth reached in the protocol tree by an execution of π on input distribution μ.*

Definition 2. *The* (internal) information cost *of protocol π with respect to μ is:*[2]

$$\mathsf{IC}_\mu(\pi) = I(Y : R, \Pi | X) + I(X : R, \Pi | Y).$$

[2]It should be noted that the same quantity can be defined as

$$\mathsf{IC}_\mu(\pi) = I(Y : R, \Pi, R^{(a)} | X) + I(X : R, \Pi, R^{(b)} | Y)$$
$$= I(Y : R, \Pi | X, R^{(a)}) + I(X : R, \Pi | Y, R^{(b)}) = I(Y : \Pi | X, R, R^{(a)}) + I(X : \Pi | Y, R, R^{(b)}).$$

Definition 3. *A protocol π is said to compute function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ with error probability ε over distribution μ if*

$$\Pr_{\mu,R,R^{(a)},R^{(b)}}[\mathrm{Out}_\pi(x,y,r,r^{(a)},r^{(b)}) = f(x,y)] \geq 1-\varepsilon.$$

Many of our technical results require that the protocol uses a limited amount of randomness at each step. This should not be surprising—this is also a requirement of Newman's theorem. This motivates the following definition.

Definition 4. *A protocol π is an* ℓ-discrete protocol *if $|\mathcal{R}_v| \leq 2^\ell$ at every node of the protocol tree.*

When a protocol is ℓ-discrete, we say that it uses ℓ bits of randomness for each message; when ℓ is clear from context, we omit it. While the standard communication model allows players to use an infinite amount of randomness at each step, this is almost never an issue, since one may always "round the message probabilities" to a finite precision. This intuition is captured in the following observation.

Observation 1. *Suppose π is a private-coin protocol. Then, there exists an ℓ-discrete protocol π' with $\ell = O(\log(|\mathcal{X}|) + \log(|\mathcal{Y}|) + \mathsf{CC}(\pi))$ such that (i) $\mathsf{CC}(\pi') \leq \mathsf{CC}(\pi)$, (ii) $\mathsf{RC}(\pi') \leq \mathsf{RC}(\pi)$, and (iii) for all x, y we have*

$$\Delta\left(\Pi'(x,y,R^{(a)},R^{(b)}), \Pi(x,y,R^{(a)},R^{(b)})\right) \leq 2^{-\Omega(\ell)}.$$

Furthermore, for any input distribution μ, the error of π' is at most the error of π plus $2^{-\ell}$. Equally small differences hold between $\mathsf{ACC}_\mu(\pi')$, $\mathsf{ARC}_\mu(\pi')$, and their π equivalents, and $\mathsf{IC}_\mu(\pi')$ is within an additive constant of $\mathsf{IC}_\mu(\pi)$.

We provide a full proof of Observation 1 in Appendix A.2. Hence, while working exclusively with discretized protocols, our theorems will also hold for non-discretized protocols, except with an additional exponentially small error term. We consider this error negligible, and hence avoid discussing it beyond this point; the reader should bear in mind, though, that when we say that we are able to simulate a discretized protocol exactly, this will imply that we can simulate any protocol with sub-inverse-exponential $2^{-\Omega(\ell)}$ error.

We are particularly interested in the case of one-way protocols. In a one-way protocol, Alice sends a single message to Bob, who must determine the output. A one-way protocol π is thus given by a function $M_\pi : \mathcal{X} \times \mathcal{R} \mapsto \mathcal{M}$; on input x Alice randomly generates r and sends $M_\pi(x,r)$. Note that if π is private-coin, then $\mathsf{IC}_\mu(\pi) = I(X : M(X,R^{(a)})|Y)$, and similarly, if π is public-coin, then $\mathsf{IC}_\mu(\pi) = I(X : R, M(X,R)|Y)$.

Finally, we close this section with a further restriction on protocols, which we call 1–1. Proving an R.N.T. result for 1–1 protocols will be a useful intermediate step in the general R.N.T. proof.

Definition 5. *A one-way protocol π is a 1–1 protocol if $M_\pi(x,\cdot)$ is 1–1 for all x.*

3 Towards a Reverse Newman's Theorem

Our main result is the following:

Theorem 3.1 (Reverse Newman's Theorem, bounded-round version)**.** *Let π be an arbitrary, ℓ-discretized, mixed-coin, q-round protocol, and let $C = \mathsf{CC}(\pi)$, $n = \log|\mathcal{X}| + \log|\mathcal{Y}|$. Suppose that π's public randomness R is chosen from the uniform distribution over the set $\mathcal{R}$, and π's private randomness $R^{(a)}$ and $R^{(b)}$ is chosen from uniform distributions over the sets $\mathcal{R}^{(a)}$ and $\mathcal{R}^{(b)}$, respectively.*

Then there exists a public-coin, q-round protocol $\tilde{\pi}$, whose public randomness R' is drawn uniformly from $\mathcal{R} \times \mathcal{R}^{(a)} \times \mathcal{R}^{(b)}$, and that has the exact same transcript distribution, i.e., for any input pair x, y and any message transcript t,

$$\Pr[\pi(x, y, R, R^{(a)}, R^{(b)}) = t] = \Pr[\tilde{\pi}(x, y, R') = t],$$

and for any distribution μ giving the input (X, Y),

$$\mathsf{IC}_\mu(\tilde{\pi}) \leq \mathsf{IC}_\mu(\pi) + O\left(q \log\left(2n\ell\right)\right). \tag{1}$$

We seem to have very little room for changing π, but actually there is one change that we are allowed to make. If Alice, for instance, wishes to send a message $M = M(x, r^{(a)})$ in protocol π, and noticing that $r^{(a)}$ is picked uniformly, she might instead send message $M(x, \phi_x(r^{(a)}))$, where ϕ_x is some permutation of $\mathcal{R}^{(a)}$. The permutation ϕ_x will somehow "scramble" the formerly-private now-public randomness $R^{(a)}$ into some new string $\tilde{r}^{(a)} = \phi_x(r^{(a)})$ about which Bob hopefully knows nothing. This "scrambling" keeps the protocol exactly as it was, changing only which $R^{(a)}$ results in which message. We will see that this can be done in such a way that, in spite of knowing $r^{(a)}$, Bob has no hope of knowing $\tilde{r}^{(a)} = \phi_x(r^{(a)})$, unless he already knows x to begin with.

As suggested by the $O(q \log(2n\ell))$-term of (1), Theorem 3.1 will be derived from the following one-way version, to be proven in Sections 4 and 5. A complete proof of Theorem 3.1 from Theorem 3.2 is given in Appendix B.

Theorem 3.2 (R.N.T. for one-way protocols)**.** *For any one-way private-coin discretized protocol π there exists a one-way public-coin discretized protocol π' such that π and π' generate the same message distributions, and for any input distribution $(X, Y) \sim \mu$, we have*

$$\mathsf{IC}_\mu(\pi') \leq \mathsf{IC}_\mu(\pi) + O(\log(\log|\mathcal{X}| \cdot \log|\mathcal{R}|) + 1).$$

We conjecture, furthermore, that a fully general R.N.T. holds:

Conjecture 3.3. *Theorem 3.1 holds with (1) replaced by*

$$\mathsf{IC}_\mu(\tilde{\pi}) \leq \tilde{O}(\mathsf{IC}_\mu(\pi)),$$

where $\tilde{O}(\cdot)$ suppresses terms and factors logarithmic in $\mathsf{IC}_\mu(\pi)$ and $\mathsf{CC}(\pi)$.

In Sections 6 and 7, we show that R.N.T.s imply fully general compression of interactive communication, and hence the resulting direct-sum theorems in information complexity. This results in new compression and direct-sum theorems for the bounded-round case. We believe that attacking Conjecture 3.3, perhaps with an improvement of our techniques, is a sound and new approach to proving these theorems.

4 R.N.T. for 1–1, one-way protocols

Before we proving Theorem 3.2, let us prove the special case of 1–1 protocols.

Theorem 4.1. *Given a one-way private-coin 1–1 protocol π, there is a one-way public-coin protocol π' generating the same message distribution, and such that for any input distribution μ, we have*

$$\mathsf{IC}_\mu(\pi') \leq \mathsf{IC}_\mu(\pi) + O(\log\log|\mathcal{X}|) \ .$$

We sketch the proof in the case that μ is uniform[3], leaving the proof to Appendix D. Let $M_\pi : \mathcal{X} \times \mathcal{R} \mapsto \mathcal{M}$ be the function Alice uses to generate her message. It will be helpful to think of M_π as a table, with rows corresponding to possible inputs x, columns corresponding to possible choices of the private random string r, and the (x, r) entry being the message $M_\pi(x, r)$. Note that

$$\begin{aligned}\mathsf{IC}_\mu(\pi) = I(X : M_\pi(X, R^{(a)})|Y) &= H(X|Y) - H\left(X|Y, M_\pi(X, R^{(a)})\right) \\ &= H(X) - H\left(X|M_\pi(X, R^{(a)})\right) \ .\end{aligned}$$

Similarly, we have $\mathsf{IC}_\mu(\pi') = H(X) - H(X|R, M_{\pi'}(X, R))$. Thus, it suffices to compare $H(X|M_\pi(x, R^{(a)}))$ and $H(X|R, M_{\pi'}(X, R))$. Suppose that Bob has received message m. In the scenario where he does not know Alice's (private) randomness, and because the protocol is 1–1, the remaining entropy of X is

$$H(X|M_\pi(X, R^{(a)}) = m) = \log|S_m|, \text{ where } S_m = \{x|\ \exists r \text{ such that } M_\pi(x, r) = m\} \ .$$

On the other hand, in π' Bob *does know* r; in this case the remaining entropy is

$$H(X|R = r, M_{\pi'}(X, R) = m) = \log|S_{m,r}|, \quad \text{where} \quad S_{m,r} = \{x|M(x, r) = m\}.$$

It could happen that $S_{m,r}$ has on average fewer elements than S_m, which would result in a smaller uncertainty about x and hence a larger information cost. However, if we could ensure that every set S_m gets broken into at most d different sets $S_{m,r}$, then we intuitively expect the entropy $\log|S_{m,r}|$ to be only $\log d$ bits smaller than $\log|S_m|$.

Our strategy to prove Theorem 4.1 will be to find a way of permuting each row of our table, in such a way that any message m is split among few columns. This can be approximately achieved by a combinatorial construction which we call a *matching graph*.

Definition 6. *An (m, n, d, δ)-matching graph is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, \mathcal{E})$ such that $|\mathcal{A}| = m$, $|\mathcal{B}| = n$, $\deg(u) = d$ for each $u \in \mathcal{A}$, and such that for all $\mathcal{A}' \subseteq \mathcal{A}$ with $|\mathcal{A}'| = n$, $G_{\mathcal{A}' \cup \mathcal{B}}$ has a matching of size at least $n(1 - \delta)$.*

In Appendix D, we use the Probabilistic Method to prove the following lemma, which shows that matching graphs with sufficiently good parameters always exist.[4]

[3] To prove Theorem 3.1, it is crucial that our technical results work for any input distribution. In the complete analysis, we compare the conditional entropy terms in $\mathsf{IC}_\mu(\pi)$ and $\mathsf{IC}_\mu(\pi')$ by conditioning on $Y = y$; this induces a possibly-arbitrary distribution on X. Fortunately, we are able to handle arbitrary distributions. Again, this is discussed in more detail in Appendix D.

[4] In the full version of the paper we will include an almost-tight lower bound of $d = \Omega(\frac{\ln k}{\delta})$, and investigate whether explicit constructions of expander graphs can be used.

Lemma 4.2. *A (kN, N, d, δ)-matching graph exists with $d = \frac{2+\ln k}{\delta^2} + \frac{\ln(1/\delta)}{\delta}$.*

Now the proof of Theorem 4.1 proceeds as follows. Let $\mathcal{A} = \mathcal{M}$ be the set of all messages and $\mathcal{B} = \mathcal{R}$ be the set of all random choices. For each row of our table, let $\mathcal{A}'_x = M(x, \mathcal{R})$ be the set of messages on that row, and extend the partial matching between $\mathcal{A}'_x$ and $\mathcal{R}$ to a perfect matching $\phi_x : \mathcal{R} \to \mathcal{A}'_x$. The new protocol π' sets $M_{\pi'}(x, r) = \phi_x(r)$ for each $r \in \mathcal{R}$. Then it will be seen that, at least approximately, no message m can be sent to more than d different columns, where d is the degree of our graph, and that this is enough to ensure that the uncertainty about X is preserved, even when the randomness is made public. As our strategy is the same regardless of the distribution we have on Alice's input X, one should intuitively expect it to work for general non-product distributions. The proofs for results in this section appear in Appendix D.

5 R.N.T. for general one-way protocols

Theorem 3.2 follows naturally from Theorem 4.1, together with the following theorem, which makes a protocol 1–1 by adding a small amount of communication.

Theorem 5.1 (Making the protocol 1–1 while revealing little information)**.** *Given a one-round discrete private-coin protocol π, there is a one-round 1–1 discrete private-coin protocol π' whose message is of the form*[5]

$$M_{\pi'}(x, r) = (M_\pi(x, r), J(x, r)),$$

and such that

$$\mathsf{IC}_\mu(\pi') \leq \mathsf{IC}_\mu(\pi) + \log\log|\mathcal{R}| + 1.$$

The proof of this theorem is left for Appendix E. We now derive Theorem 3.2 as a corollary.

Proof of Theorem 3.2. Suppose π is a one-way discrete protocol. Let π_2 be the 1–1 protocol guaranteed by Theorem 5.1, and let π_3 be the protocol guaranteed by Theorem 4.1 applied to π_2. Note that π_3's message is of the form $M_{\pi_3}(X, R) = (M_\pi(X, R), J(X, R))$, since it is equidistributed with M_{π_2}. Furthermore, we have $\mathsf{IC}_\mu(\pi_3) \leq \mathsf{IC}_\mu(\pi) + \log(\log|\mathcal{X}| \cdot \log|\mathcal{R}|) + 1$. Now, create a protocol π_4, which is identical to π_3, except that Alice omits $J(X, R)$. It is easy to see that π_4 and π are equidistributed and that

$$\mathsf{IC}_\mu(\pi_4) \leq \mathsf{IC}_\mu(\pi_3) \leq \mathsf{IC}_\mu(\pi) + O(\log(\log|\mathcal{X}| \cdot \log|\mathcal{R}|) + 1)\,.$$

This completes the proof. □

[5] On any input x and any choice of randomness r, $M_{\pi'}(x, r)$ is obtained by taking $M_\pi(x, r)$ and adding some additional communication $J(x, r)$.

6 Compression for public-coin protocols

We present in this section two results of the following general form: we will take a public-coin protocol π that reveals little information, and "compress" it into a protocol ρ that uses little communication to perform the same task with about the same error probability. It turns out that the results in this setting are simpler and give stronger compression than in the case where Alice and Bob have private randomness (such as in [BBCR10, BR11]). We present two bounds, one that is dependent on the number of rounds of π, but which is also round-efficient, in the sense that ρ will not use many more rounds than π; and one that is independent of the number of rounds of π, but where the compression is not as good when the number of rounds of π is small.

Theorem 6.1. *Suppose there exists a public-coin protocol π to compute f over the distribution μ with error probability ε, and let $C = \mathsf{CC}(\pi)$, $I = \mathsf{IC}_\mu(\pi)$. Then there is a public-coin protocol ρ computing f over μ with error $\varepsilon + \delta$, and with $\mathsf{ACC}_\mu(\rho) = O\left(I \log C/\delta\right)$.*

Let us ignore the public randomness of π for now, and explain how we will show that any deterministic protocol π can be simulated with communication roughly:

$$I(Y : \Pi|X) + I(X : \Pi|Y) = H(\Pi|X) + H(\Pi|Y)$$

(the last equality follows because $H(\Pi|X, Y) = 0$, since the transcript Π is a function of X and Y). Given her input x, Alice knows the distribution of $\Pi|x$, and she can hence organize the set $\Pi^{(a)} = \pi(x, \mathcal{Y})$ into a weighted binary tree $T^{(a)}$, as follows:

1. The root is the largest common prefix (lcp) of the transcripts in $\Pi^{(a)}$, and the remaining nodes are defined inductively.

2. If we have node τ, then we let its children be $\tau 0 \tau_0$ and $\tau 1 \tau_1$, where τ_i is the lcp of the transcripts in $\Pi^{(a)}$ beginning with τi.

3. The leaves t of $T^{(a)}$ have weight $w(t) = \Pr[\pi(X, Y) = t|X = x]$.

In this way, the leaves of $T^{(a)}$ are exactly $\Pi^{(a)}$, and are weighted according to the distribution of $\Pi(x, Y|_x)$. The weight of a non-leaf node in the tree is the sum of the weights of its descendant leaves. Bob forms a similar tree $T^{(b)}$ from the image $\Pi^{(b)} = \pi(\mathcal{X}, y)$.

Now it must hold that $\pi(x, y)$ is the unique leaf that is in both $T^{(a)}$ and $T^{(b)}$. Alice and Bob then proceed in stages to find the common leaf: at a given stage they have agreed that certain *partial transcripts*, which are nodes in their respective trees, are prefixes of $\pi(x, y)$. They then each choose a *candidate transcript*, which is a leaf extending their partial transcript, and find the lcp of their two candidate transcripts, i.e., find the first bit at which their candidate transcripts disagree. Now, because one of the players actually knows what that bit should be (that bit depends either on x or on y), the player who got it wrong can change her/his bit to its correct value, and this will give either Alice or Bob a new partial transcript in her/his tree which extends the previous partial transcript she/he had. They proceed this way until they both know $\pi(x, y)$. It will be seen that there is a way of picking the candidate transcripts so that the total probability mass under the nodes they have agreed upon halves at every correction, and this will be enough to

show that Alice will only need to correct her candidate transcript $H(\Pi|X)$ times (and Bob $H(\Pi|Y)$ times) on average. Efficient protocols for finding the lcp of two strings will then give us the required bounds. We give the full proof in Appendix C.1.

This procedure offers no guarantee on the number of rounds of the compressed protocol ρ. It is possible to compress a public-coin protocol on a round-by-round basis while preserving, up to a multiplicative constant, the total number of rounds used.

Theorem 6.2. *Suppose there exists a public-coin protocol π to compute f over the input distribution μ with error probability ε, and let $I = \mathsf{IC}_\mu(\pi)$ and $q = \mathsf{RC}(\pi)$. Then there exists a public-coin protocol ρ that computes f over μ with error $\varepsilon+\delta$, and with $\mathsf{ACC}_\mu(\rho) = O(I + q \log \frac{2q}{\delta})$ and $\mathsf{ARC}_\mu(\rho) = O(q)$.*[6]

The idea of the proof is to show the result one round at a time. In round i, Alice, say, must send a certain message m_i to Bob. From Bob's point of view, this message is drawn according to the random variable $M_i = M_i(\tilde{X}, y, r, m_1, \ldots, m_{i-1})$ where $\tilde{X}$ is Alice's input conditioned on Bob's input being y, on the public randomness r, and on the messages $m_1, \ldots, m_{i-1}$ that were previously exchanged. We will show that there is a sub-protocol σ_i that can simulate round i with small error by using constantly-many rounds and

$$O(H(M_i|y, r, m_1, \ldots, m_{i-1})) = I(X : M_i|y, r, m_1, \ldots, m_{i-1})$$

bits of communication *on average*. Then putting these sub-protocols together, and truncating the resulting protocol whenever the communication or the number of rounds is excessive, we obtain the protocol ρ which simulates π.

The procedure to compress each round is achieved through an interactive variant of the Slepian-Wolf theorem [SW73, RW05, BKV08]. We could not apply the known theorems directly, however, since they were made to work in different settings. We give the full proof in Appendix C.2.

7 Applications

From the combination of Theorems 3.1 and 6.2, and Observation 1, we can obtain a new compression result for general protocols.

Corollary 7.1. *Suppose there exists a mixed-coin, q-round protocol π to compute f over the input distribution μ with error probability ε, and let $C = \mathsf{CC}(\pi)$, $I = \mathsf{IC}_\mu(\pi)$, $n = \log|\mathcal{X}| + \log|\mathcal{Y}|$. Then there exists a public-coin, $O(q)$-round protocol ρ that computes f over μ with error $\varepsilon + \delta$, and with*

$$\mathsf{CC}(\rho) \leq O\left(I + q \log\left(\frac{qnC}{\delta}\right)\right). \tag{2}$$

As we will see in the following sub-section, this will result in a new direct sum theorem for bounded-round protocols. In general, given that we have already proven Theorem 6.1, and given that this approach shows promise in the bounded-round case, it becomes worthwhile to investigate whether we can prove Conjecture 3.3 with similar techniques.

[6] Note that this compression does not depend on the communication complexity at all.

7.1 Direct-sum theorems for the bounded-round case

The following theorem was proven in [BBCR10, Theorem 12]:

Theorem 7.2. *Suppose that there is a q-round protocol π^k that computes k copies of f with communication complexity C and error ε, over the k-fold distribution μ^k. Then there exists a q-round mixed-coin protocol π that computes a single copy of f with communication complexity C and the same error probability ε, but with information cost $\mathsf{IC}_\mu(\pi) \leq \frac{2C}{k}$ for any input distribution μ.*

As a consequence of this theorem, and of Corollary 7.1, we will be able to prove a direct sum theorem. The proof is a simple application of Theorem 7.2, and Corollary 7.1.

Theorem 7.3 (Direct sum theorem for the bounded-round case)**.** *There is some constant d such that, for any input distribution μ and any $0 < \varepsilon < \delta < 1$, if f requires, on average,*

$$C + q \log \left(\frac{qnC}{\delta - \varepsilon} \right)$$

bits of communication, to be computed over μ with error δ in dq (average) rounds, then $f^{\otimes k}$ requires kC bits of communication, in the worst case, to be computed over $\mu^{\otimes k}$ with error ε in q rounds.

7.2 Comparison with previous results

We may compare Theorem 7.1 with the results of [BR11]. In that paper, the nC factor is missing inside the log of equation (2), but the number of rounds of the compressed protocol is $O(q \log I)$ instead of $O(q)$. A similar difference appears in the resulting direct-sum theorems.

We remark that the compression of Jain et al. [JPY12] is also achieved with a round-by-round proof. Our direct-sum theorem is incomparable with their more ambitious direct-product result. It is no surprise, then, that the communication complexity of their compression scheme is $O(\frac{qI}{\delta})$, i.e., it incurs a factor of q, whereas we pay only an additive term of $\tilde{O}(q)$. However, their direct-product result also preserves the number of rounds in the protocol, whereas in our result the number of rounds is only preserved within a constant factor.

8 Acknowledgements

The authors thank Pavel Pudlák for fruitful discussions. Part of the research for this paper was made at Schloss Dagstuhl during the workshop "Algebraic and Combinatorial Methods in Computational Complexity."

References

[BBCR10] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *Proceedings of the 42nd STOC*, pages 67–76, 2010.

[BKV08] Harry Buhrman, Michal Koucký, and Nickolay Vereshchagin. Randomized individual communication complexity. In *Proceedings of the 23rd CCC*, pages 321–331, 2008.

[BR11] Mark Braverman and Anup Rao. Information equals amortized communication. In *Proceedings of the 52nd STOC*, pages 748–757, 2011.

[Bra12] Mark Braverman. Interactive information complexity. In *Proceedings of the 45th STOC*, pages 505–524, 2012.

[BYJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004. Special issue of FOCS 2002.

[CSWY01] Amit Chakrabarti, Yaoyun Shi, Antony Wirth Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message passing. In *Proceedings of the 42nd FOCS*, 2001.

[FPR94] Uriel Feige, David Peleg, and Prabhakar Raghavan. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

[JPY12] Rahul Jain, Attila Pereszlenyi, and Penghui Yao. A direct product theorem for bounded-round public-coin randomized communication complexity. In *Proceedings of the 53rd FOCS*, 2012.

[JRS03] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In *Proceedings of the 30th ICALP*, 2003.

[KS89] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1989.

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

[Pan12] Denis Pankratov. Direct sum questions in classical communication complexity. Master's thesis, University of Chicago, 2012.

[RW05] Renato Renner and Stefan Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In *Advances in Cryptology - ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 199–216, 2005.

[SW73] David Slepian and Jack K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, 19:471–480, 1973.

A More Preliminary Definitions

Definition 7 (Statistical Distance)**.** *The* statistical distance *between random variables A, A' (also known as total variation distance) is*

$$\Delta\left(A, A'\right) = \sum_{a \in \mathcal{A}} \left|\Pr[A = a] - \Pr[A' = a]\right|.$$

A.1 Facts and definitions of information theory

For a given probability random variable A distributed over the support $\mathcal{A}$, its entropy is

$$H(A) = \sum_{a \in \mathcal{A}} p_a \log \frac{1}{p_a},$$

where $p_a = \Pr[A = a]$. Given a second random variable B, the conditional entropy $H(A|B)$ equals

$$\mathbb{E}_{b \in B}[H(A|B = b)].$$

In this paper, and when clear from the context, we denote a conditional distribution $A|B = b$ more succinctly by $A|b$.

We let $I(A : B)$ denote the Shannon mutual information between A and B:

$$I(A : B) = H(A) - H(A|B) = H(B) - H(B|A).$$

Fact A.1 (Chain rule)**.**

$$I(A_1 \dots A_k : B|C) = I(A_1 : B|C) + \sum_{i=2}^{k} I(A_i : B|C, A_1, \dots, A_{i-1})$$

Fact A.2. *For any two random variables A, B over the same universe $\mathcal{U}$, it holds that*

$$|H(A) - H(B)| \leq \log(|\mathcal{U}|)\Delta\left(A, B\right) + 1,$$

A.2 Proof of Observation 1

Proof. Let π be given by its protocol tree; for each node v, let its corresponding function be $M_v : \mathcal{X} \times \mathcal{R} \to \mathcal{C}(v)$ (if it is Alice's node) or $M_v : \mathcal{Y} \times \mathcal{R}_v \to \mathcal{C}(v)$.

We let π' be given by the same protocol tree but where the functions M_v are restricted to a finite set $\mathcal{R}'_v$ of size $\leq k = 2^{10\ell}$, with $\ell = \log|\mathcal{X}||\mathcal{Y}| + \mathsf{CC}(\pi)$. Hence by construction π' has the same worst-case communication and number of rounds as π.

Let R_v be a random variable uniformly distributed over $\mathcal{R}_v$ and R'_v be a random variable uniformly distributed over $\mathcal{R}'_v$.

Claim 1. *For any node v of Alice's there is a choice of $\mathcal{R}'_v$ of given size such that*

$$\left|\Pr[M_v(x, R_v) = m] - \Pr[M_v(x, R'_v) = m]\right| \leq 2^{-4\ell}$$

for every x and m. The obvious analogue holds for Bob's nodes.

We prove that $\mathcal{R}'_v$ exists by the probabilistic method. Let $\tilde{\mathcal{R}} = \{r_1, \ldots, r_k\}$ be a random variable which is a multiset obtained by picking k elements uniformly from $\mathcal{R}_v$, and define R'_v as the random variable which picks an element $r_i \in \tilde{\mathcal{R}}$ uniformly at random (counting multiplicities). Let P_m denote the random variable that is

$$P_m = \Pr[M_v(x, R'_v) = m] = \frac{\sum_{i=1}^{k} [M_v(x, r_i) = m?]}{k}.$$

By linearity of expectation we find that:

$$\mathbb{E}[P_m] = \frac{\sum_{i=1}^{k} \mathbb{E}[M_v(x, r_i) = m?]}{k} = \Pr[M_v(x, R_v) = m].$$

And hence by Hoeffding's inequality we conclude that:

$$\Pr[|P_m - \Pr[M_v(x, R_v) = m]| > 2^{-4\ell}] \leq 2 \exp\left(-2k2^{-8\ell}\right) \ll 2^{-\ell}.$$

Hence by a union bound there must exist a choice for $\tilde{\mathcal{R}}$ such that

$$|P_m - \Pr[M_v(x, R_v) = m]| \leq 2^{-4\ell}$$

holds for every x and m; this choice is $\mathcal{R}'_v$.

Now fix x, y; from the claim it follows that for any transcript t,

$$|\Pr[\pi(x, y) = t] - \Pr[\pi'(x, y) = t]| \leq 2^{-3\ell},$$

which in turn implies that

$$\Delta\left(\Pi(x, y, R^{(a)}, R^{(b)}), \Pi'(x, y, \mathcal{R}'^{(a)}, \mathcal{R}'^{(b)})\right) \leq 2^{-2\ell}.$$

This results in a difference of $\leq 2^{-\ell}$ in success probability, average communication complexity, and average number of rounds, for any given input distribution.

To prove that there is a small difference in information cost, note that:

$$I(\Pi : X|Y) = H(\pi(X, Y, R)|Y) - H(\pi(X, Y, R)|X, Y),$$

and then use Fact A.2 conclude that

1. $|H(\pi(X, Y, R)|Y = y) - H(\pi'(X, Y, R')|Y = y)| = O(1)$ for all y, and
2. $|H(\pi(X, Y, R)|X = x, Y = y) - H(\pi'(X, Y, R')|X = x, Y = y)| = O(1)$ for any x, y, and hence
3. $|I(\Pi : X|Y) - I(\Pi' : X|Y)| = O(1)$,

Now, by a symmetric reasoning for Bob, we find that $|\mathsf{IC}_\mu(\pi) - \mathsf{IC}_\mu(\pi)| = O(1)$. □

B Proofs of Section 3

Let us prove Theorem 3.1 as a consequence of Theorem 3.2.

Proof of Theorem 3.1. The new protocol $\tilde{\pi}$ first picks public randomness r according to R and then proceeds by simulating $\pi|_r$ on a round-per-round basis. Since π is ℓ-discrete, the private randomness used in each round is a binary string of length $\leq \ell$, picked uniformly at random from some set, and independently from the remaining rounds; i.e., $R^{(a)} = R^{(a,1)} \times \ldots \times R^{(a,q)}$ for random variables $R^{(a,j)}$ uniformly distributed over $\mathcal{R}^{(a,j)} \subseteq \{0,1\}^\ell$. The same will hold for $R^{(b)}$, and the message π_j for round j depends only on x (or y), $r^{(a,j)}$ (or $r^{(b,j)}$), and on the previous messages. Now suppose we wish to simulate round $j+1$ of π, and that we have already successfully simulated π up to round j, which cost us at most

$$I(X : \Pi_{\leq j}|Y, r) + I(Y : \Pi_{\leq j}|X, r) + j \log(2n\ell)$$

bits of information. Suppose w.l.o.g. that it is Alice's turn to communicate, and that messages $m_1, \ldots, m_j$ have been exchanged up to this point. Then round $j+1$ is simulated thus: we consider the one-way private-coin protocol $M_v(x, r^{(a,j+1)})$ where v is the node in π's protocol tree after the path $m_1, \ldots, m_j$, and we run M'_v instead, as given by Theorem 3.2. As per that theorem, for the distribution of the inputs $X|r, m_{\leq j}$ and $Y|r, m_{\leq j}$, the information revealed will be

$$I(X : M'(X, R^{(a,j+1)}), R^{(a,j+1)}|r, Y, m_{\leq j}) \\ \leq I(X : M(X, R^{(a,j+1)})|r, Y, m_{\leq j}) + \log(2n\ell).$$

When averaged over $\Pi_1, \ldots, \Pi_j$ (recall that the message distribution is always preserved), the left-hand side becomes the information cost of our new protocol on round $j+1$, and the right-hand side becomes $I(X : \Pi_{j+1}|Y, r, \Pi_{\leq j}) + \log(2n\ell)$, which when added to the previous information cost is exactly

$$I(X : \Pi_{\leq j+1}|Y, r) + I(Y : \Pi_{\leq j+1}|X, r) + (j+1) \log(2n\ell).$$

□

C Proofs of Section 6

C.1 Proof of Theorem 6.1

We prove Theorem 6.1 by constructing an interactive protocol that makes use of a special device, which we call *lcp box*. This is a conceptual interactive device with the following behavior: Alice takes a string x and puts it in the lcp box, Bob takes a string y and puts it in the lcp box, then a button is pressed, and Alice and Bob both learn the least index j such that $x_j \neq y_j$; if no such j exists, they both learn that $x = y$. We will charge them $O(\log n)$ bits of communication for each use of the lcp box, where $n = \max(|x|, |y|)$.

The use of an lcp box can be simulated with an error-prone implementation:

Lemma C.1 ([FPR94])**.** *There is a randomized public-coin protocol such that on input two n-bit strings x, y, it outputs the first index j such that $x_j \neq y_j$ with probability at least $1 - \varepsilon$, if such an j exists, and otherwise outputs that the two strings are equal, with worst-case communication complexity $O(\log(n/\varepsilon))$.*

Corollary C.2. *Any protocol $\tilde{\rho}$ that uses an lcp box ℓ times on average can be simulated with error δ by a protocol ρ that does not use an lcp box, and communicates an average of $O(\ell \log(\frac{\ell}{\delta}))$ extra bits.*

Proof. The protocol ρ simulates $\tilde{\rho}$ by replacing each use of the lcp box with the protocol given by Lemma C.1 with error parameter $\varepsilon = \frac{\delta^2}{4\ell}$. The $\log n$ bits of communication have been accounted for, and hence the ρ's extra communication is $O(\ell \log \frac{\ell}{\delta})$ on average. To bound the error, notice that by Markov's inequality the probability that the number of calls to the lcp box (in a random run of $\tilde{\rho}$) is greater than $\frac{2}{\delta}\ell$ is less than $\delta/2$. But conditioned on not having such a large number of calls, the probability that we make an error in any of our simulations of the lcp box is at most $\delta/2$ by a union bound. Hence, ρ will correctly simulate $\tilde{\rho}$ with error probability at most δ. □

Using an lcp box will allow us to ignore error events until the very end of the proof, avoiding an annoying technicality that offers no additional insight. We are now ready to prove Theorem 6.1; our compression scheme is similar, but not identical, to that of [BBCR10] — the absence of private randomness allows for a more elementary proof.

Proof of Theorem 6.1. We first define a protocol $\tilde{\rho}$ using an lcp box as follows. On inputs x and y, Alice and Bob first pick the shared randomness r in the same way as in protocol π. Then, Alice organizes $\Pi^{(a)} = \pi(x, \mathcal{Y}, r)$ into the prefix-tree $T^{(a)}$, as follows:

1. The root is the largest common prefix (lcp) of the transcripts in $\Pi^{(a)}$, and the remaining nodes are defined inductively.

2. If we have node τ, then we let its left child be the lcp of those transcripts in $\Pi^{(a)}$ beginning with $\tau 0$, and its right child the lcp of those beginning with $\tau 1$.

3. The leaves t of $T^{(a)}$ have weight $w(t) = \Pr[\pi(X, Y, r) = t | X = x]$.

In this way, the leaves of $T^{(a)}$ are exactly $\Pi^{(a)}$, and are weighted according to the distribution of $\Pi(x, Y|_x, r)$. We use the convention that the weight of a non-leaf node in the tree is the sum of the weights of its descendant leaves.

Bob forms a similar tree $T^{(b)}$ from the image $\Pi^{(b)} = \pi(\mathcal{X}, y, r)$. It can be seen that $\Pi^{(a)} \cap \Pi^{(b)} = \{\pi(x, y, r)\}$, because every leaf in $T^{(a)}$ is of the form $\pi(x, y', r)$ for some y', and every leaf in $T^{(b)}$ is of the form $\pi(x', y, r)$ for some x', and clearly if $t = \pi(x, y', r) = \pi(x', y, r)$, then it must also hold that $t = \pi(x, y, r)$.

This suggests that we might find $\pi(x, y, r)$, by running a generic protocol to uncover the common leaf of $T^{(a)}$ and $T^{(b)}$. This is harder than set intersection, and such a generic protocol would hence require $\Omega(|\Pi^{(a)}| + |\Pi^{(b)}|)$ communication [KS89]. So we must necessarily exploit the fact that our trees arise from a protocol. We will now conclude the proof by showing that Alice and Bob can determine the intersection $\Pi^{(a)} \cap \Pi^{(b)}$ by using an lcp box $O(\mathsf{IC}_\mu(\pi))$ many times on average.

In the descriptions below, we will use $t^{(a)}$ (and $t^{(b)}$) to designate a leaf of $T^{(a)}$ (resp. $T^{(b)}$), and $\tau^{(a)}$ ($\tau^{(b)}$) to designate an arbitrary node of $T^{(a)}$ (resp. $T^{(b)}$). Alice and Bob will proceed in stages $s = 1, 2, \ldots$ to find a common leaf of $T^{(a)}$ and $T^{(b)}$:

(1) At the beginning of stage s, they will have agreed that certain nodes in their respective trees — $\tau^{(a)}(s) \in T^{(a)}$ and $\tau^{(b)}(s) \in T^{(b)}$ — are prefixes of $\pi(x, y, r)$. In the first stage, $\tau^{(a)}(1)$ and $\tau^{(b)}(1)$ will be the roots of the trees.

(2) Then Alice picks a *candidate leaf* $t^{(a)}$ that is a successor of $\tau^{(a)}(s)$, and Bob picks a candidate leaf $t^{(b)}$ that is a successor of $\tau^{(b)}(s)$.

(3) They then use an lcp box to find the least position j for which $t^{(a)}_j \neq t^{(b)}_j$. If there is no such j then they both have $t^{(a)} = t^{(b)} = \pi(x, y, r)$, and the simulation terminates.

(4) Now, if in the protocol π it was Alice's turn to communicate bit j, then — in protocol $\tilde{\rho}$ — Alice sets $\tau^{(a)}(s+1) = \tau^{(a)}(s)$, and Bob will set $\tau^{(b)}(s+1)$ as explained in the next item. Then they proceed to the next stage.

(5) Bob looks at the node $\hat{\tau}^{(b)}$ given the first $j-1$ bits of his candidate leaf $t^{(b)}$. Now suppose that $t^{(b)}$ is a left successor of $\hat{\tau}^{(b)}$. Then Bob will set $\tau^{(b)}(s+1)$ to be the right child of $\hat{\tau}^{(b)}$. The node $\tau^{(b)}(s+1)$ is called the "flip" of $t^{(b)}$ at position j.

Note that $\hat{\tau}^{(b)}$ must be a node in the tree $T^{(b)}$, since $\pi(x, y, r)$ is prefixed by $\hat{\tau}^{(b)}$ and has a different bit than $t^{(b)}$ on the j-th position (namely, the j-th bit of $t^{(a)}$). Because every left child of $\hat{\tau}^{(b)}$ will have the same bit as $t^{(b)}$ at position j, we see that $\pi(x, y, r)$ can not be a left successor of $\hat{\tau}^{(b)}$. Hence $\pi(x, y, r)$ must be a successor of $\tau^{(b)}(s+1)$, and the property required by item (1) is preserved for stage $s+1$. Of course, a symmetric argument holds when $t^{(b)}$ is a right successor of $\hat{\tau}^{(b)}$, and a similar reasoning is applied if in protocol π it was Bob's turn to communicate bit j (then it will be Alice who changes her prefix).

The property required by item (1) ensures that Alice and Bob will eventually agree on π itself. All that is left for us to specify is how Alice and Bob pick their candidate. We will show that for any weighted binary prefix tree T, and for any prefix $\tau \in T$, there is a way of picking a candidate leaf t extending τ such that any "flip" of t will have at most half the weight of τ. Then this implies that weight of either $\tau^{(a)}(s)$ or $\tau^{(b)}(s)$ will halve at each stage.

The candidate is picked as follows: we let $\tau_1 = \tau$ to begin with; if we have already defined τ_i, then we choose τ_{i+1} to be whichever child (left or right) has higher weight, (arbitrarily) opting for the left child if both weights are equal. When we reach a leaf, this is our candidate t. Define w_i to be the weight of τ_i. Suppose τ' is a flip of our candidate, i.e., τ' is the one child of τ_i, for some i for which τ_{i+1} is the other child. Then by our choice of τ_{i+1}, τ' has weight at most $w_i/2 \leq w_1/2$, as intended.

Now we can finish the analysis. For a given inputs x, y, and a given transcript t, let

$$p^{(a)}(x, t) = \Pr[\pi(x, Y|_x, r) = t] \qquad p^{(b)}(y, t) = \Pr[\pi(X|_y, y, r) = t].$$

On inputs x and y, with $t = \pi(x, y, r)$, Alice will correct her prefix no more than $\log \frac{1}{p^{(a)}(x,t)}$ times, because the probability of the descendant leaves of her current prefix halves with

each correction. A similar thing happens to Bob, and hence the total number of stages that Alice and Bob run on input x, y is bounded by

$$\log \frac{1}{p^{(a)}(x,t)} + \log \frac{1}{p^{(b)}(y,t)}.$$

Taking the average over μ, this is exactly $\mathsf{IC}_\mu(\pi_r)$ stages. For instance, for a fixed x, the left-hand term, averaged over $Y|X = x$, is

$$\mathbb{E}_{y \in Y|_x} \left[\log \frac{1}{p^{(a)}(x,t)} \right] = \sum_{t \in \pi(x,\mathcal{Y},r)} p^{(a)}(x,t) \log \frac{1}{p^{(a)}(x,t)} = H(\Pi|x,r).$$

Now averaging over X, we get exactly $H(\Pi|X,r) = I(Y : \Pi|X,r)$. An averaging of the right-hand term will give us $I(X : \Pi|Y,r)$. Finally, averaging over the choice of r, we get a global average of $\mathsf{IC}_\mu(\pi)$ stages.

The protocol $\tilde{\rho}$ simulates π perfectly; Alice and Bob make one use of the lcp box per stage, on two strings whose length is at most $\mathsf{CC}(\pi)$, hence $\mathsf{ACC}_\mu(\tilde{\rho}) = O(\mathsf{IC}_\mu(\pi) \log \mathsf{CC}(\pi))$. Then, by Corollary C.2, we can replace each use of the lcp box with the protocol given by Lemma C.1. This is our protocol ρ, which simulates π with error δ and average communication:

$$\mathsf{ACC}_\mu(\rho) = O\left(\mathsf{IC}_\mu(\pi) \log \frac{\mathsf{CC}(\pi)\mathsf{IC}_\mu(\pi)}{\delta} \right) = O\left(\mathsf{IC}_\mu(\pi) \log \frac{\mathsf{CC}(\pi)}{\delta} \right).$$

□

C.2 Proof of Theorem 6.2

In a similar fashion to the proof of Theorem 6.1, we will make use of a special interactive device, which we call a *membership box*. Its behavior is as follows: one player takes a string z and puts it in the membership box, the other player takes a family of sets of strings $\mathcal{Z}_1 \subseteq \ldots \subseteq \mathcal{Z}_K$ and puts it in the box, a button is pressed, and then both players will know an index k such that $z \in \mathcal{Z}_k$ (with $k = \infty$ if $z \notin \mathcal{Z}_k$ for any natural number k), and whenever $z \in \mathcal{Z}_k$, then the second player will additionally know which string z is. Each use of the box costs the players $2\log|\mathcal{Z}_k| + 2k$ bits of communication and $3k$ rounds, and $2\log|\mathcal{Z}_K| + 2K$ bits and $3K$ rounds if $z \notin \mathcal{Z}_k$ for any natural number k.

As before, there exists a procedure to simulate the use of a membership box:

Lemma C.3. *Suppose that Alice is given a string z, and Bob is given a family of finite sets $\mathcal{Z}_1 \subseteq \cdots \subseteq \mathcal{Z}_K$. Then there exists a randomized public-coin protocol which outputs k whenever $z \in \mathcal{Z}_k$, and in that case Bob will know what z is, except with a failure probability of at most ε. A run of this protocol uses at most $3k$ rounds and $2\log|\mathcal{Z}_k| + \log\frac{1}{\varepsilon} + 2k$ bits of communication.*

Proof. The protocol is divided into stages and works as follows. On the first stage, Bob begins by sending the number $\ell_1 = \log|\mathcal{Z}_1|$ in unary to Alice, and Alice responds by picking $L_1 = \ell_1 + \log\frac{1}{\varepsilon} + 1$ random linear functions $f_1^{(1)}, \ldots, f_{L_1}^{(1)} : \mathbb{Z}_2^{|z|} \to \mathbb{Z}_2$ using public randomness, and sending Bob the hash values $f_1^{(1)}(z), \ldots, f_{L_1}^{(1)}(z)$. Bob then looks for a

string $z' \in \mathcal{Z}_1$ that has the same hash values he just received; if there is such a string, then Bob says so, and the protocol is finished with Bob assuming that $z' = z$.

Otherwise, the protocol continues. At stage k, Bob computes the number $\ell_k = \log |\mathcal{Z}_k|$, and sends the number $\ell_k - \ell_{k-1}$ in unary to Alice; Alice responds by picking $L_k = \ell_k - \ell_{k-1} + 1$ random linear functions $f_1^{(k)}, \ldots, f_{L_k}^{(k)}$, whose evaluation on z she sends over to Bob. Bob then looks for a string $z' \in \mathcal{Z}_k$ that has the same hash values for all the hash functions which were picked in this and previous stages; if there is such a string, then Bob says so, and the protocol is finished with Bob assuming that $z' = z$.

An error will occur whenever a $z' \neq z$ is found that has the same fingerprint as z. The probability that this happens at stage k for a specific $z' \in \mathcal{Z}_k$ is 2^{-L}, where $L = \ell_k + k + \log \frac{1}{\varepsilon}$ is the total number of hash functions picked up to this stage. By a union bound, the probability that such a z' exists is at most $|\mathcal{Z}_k| 2^{-\ell_k} \frac{\varepsilon}{2^k} \leq \frac{\varepsilon}{2^k}$. Again by a union bound, summing over all stages k we get a total error probability of ε.

To bound the communication, notice that sending every ℓ_i costs Bob at most $\log |\mathcal{Z}_k|$ bits of total communication (when $x \in \mathcal{Z}_k$), that the total number of hash values sent by Alice is at most $\log |\mathcal{Z}_k| + k + \log \frac{1}{\varepsilon}$, and that Bob's reply (saying whether the protocol should continue) costs him at most k bits. □

From this we get an analogue of Corollary C.2.

Corollary C.4. *Any protocol $\tilde{\rho}$ that uses a membership box ℓ times can be simulated with error δ by a protocol ρ that does not use a membership box, and communicates an average of $O(\ell \log(\frac{\ell}{\delta}))$ more bits than $\tilde{\rho}$.*

In order to compress one round of a given protocol, we will use the following theorem, which in itself can be seen as a variant of the Slepian–Wolf coding theorem.

Theorem C.5 (Constant-round average-case one-shot Sleepian–Wolf)**.** *Suppose that Alice and Bob are given inputs (X, Y) drawn according to the distribution μ. Then there is a protocol that makes use of a membership box and such that Alice successfully sends X to Bob using $O(H(X|Y) + 1)$ bits of communication and $O(1)$ rounds on average.*

Contrast this to the classical Slepian–Wolf theorem, where Alice and Bob are given a stream of i.i.d. pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$, and Alice gets to transmit $X_1, \ldots, X_n$ by using only one-way communication, and with an *amortized* communication of $H(X|Y)$.

Proof. Let y be Bob's given input. For a given x in the support of X, let $p(x) = \Pr[X = x|Y = y]$, and for a given subset $\mathcal{X}$ of the same support, let $p(\mathcal{X}) = \Pr[X \in \mathcal{X}|Y = y]$. Then Bob begins by arranging the x's in the support of X by decreasing order of the probability $p(x)$. He then defines the two sets

$$\mathcal{X}_1 = \{x_1, \ldots, x_{i(1)}\}, \qquad \mathcal{Z}_1 = \mathcal{X}_1,$$

where $i(1)$ is the minimal index which makes $p(\mathcal{X}_1) \geq 1/2$. Inductively he then defines:

$$\mathcal{X}_{k+1} = \{x_{i(k)+1}, \cdots, x_{i(k+1)}\}, \qquad \mathcal{Z}_{k+1} = \mathcal{Z}_k \cup \mathcal{X}_{k+1},$$

where $i(k+1) > i(k)$ is the minimal index which makes $p(\mathcal{X}_{k+1}) \geq \frac{1-p(\mathcal{Z}_k)}{2}$. I.e. $\mathcal{X}_{k+1}$ is the smallest set which takes the remaining highest-probability x's so that they total at least half of the remaining probability mass.

Because at least one new x_i is added at every step, this inductive procedure gives Bob a finite number of sets $\mathcal{Z}_1, \ldots, \mathcal{Z}_K$; in fact it can be seen that $\mathcal{X}_K$ is the singleton set $\mathcal{X}_K = \{x_{i(K)}\}$ which contains only the lowest probability element of the support (because by construction $x_{i(K)}$ will never be paired with $x_{i(K)-1}$). Then the protocol consists of Alice putting her input x into the membership box, and Bob putting $\mathcal{Z}_1 \subseteq \ldots \subseteq \mathcal{Z}_K$, and pushing the button. This will cost them $3k$ rounds and $2k+2\log \mathcal{Z}_k$ bits of communication, where k is the index such that $x \in \mathcal{X}_k$.

To see the correctness, notice that x must be in one of the $\mathcal{Z}_k$ sets, hence by the end of the protocol Bob knows what x is.

Now let us bound the average number of rounds and communication complexity. First notice that $p(\mathcal{X}_k) \leq 2^{-k}$, and hence, taking the average over Alice's inputs, we find that

$$\sum_{k=1}^{K} p(\mathcal{X}_k) 3k = O(1)$$

must upper bound the average number of rounds, as well as the contribution of the $2k$ term to the average communication. To upper-bound the contribution of the $2\log \mathcal{Z}_k$ term, we first settle that:

(i) $p(\mathcal{X}_k) \leq 2p(\mathcal{X}_{k+1}) + 2p(x_{i(k)})$, which can be seen by summing the two inequalities that follow from the definition of $\mathcal{X}_k$ and $\mathcal{X}_{k+1}$:

$$p(\mathcal{X}_k) - p(x_{i(k)}) \leq \frac{1 - p(\mathcal{Z}_{k-1})}{2}, \qquad \frac{1 - p(\mathcal{Z}_k)}{2} \leq p(\mathcal{X}_{k+1}),$$

after which we get

$$\frac{p(\mathcal{X}_k)}{2} - p(x_{i(k)}) \leq p(\mathcal{X}_{k+1}).$$

(ii) $|\mathcal{Z}_k| \leq \frac{1}{p(x)}$ for any $x \in \mathcal{X}_{k+1} \cup \{x_{i(k)}\}$, which follows since every $x' \in \mathcal{Z}_k$ has a higher-or-equal probability than the x's in $\mathcal{X}_{k+1} \cup \{x_{i(k)}\}$, but the sum of all the $p(x')$ still adds up to less than 1; and

Now we are ready to bound the remaining term in the average communication:

$$\sum_{k=1}^{K} p(\mathcal{X}_k) \log |\mathcal{Z}_k| \leq 2 \sum_{k=1}^{K-1} p(\mathcal{X}_{k+1}) \log |\mathcal{Z}_k| + p(\mathcal{X}_K) \log |\mathcal{Z}_K| + 2 \sum_{k=1}^{K} p(x_{i(k)}) \log |\mathcal{Z}_k|$$

$$\leq 5 \sum_x p(x) \log \frac{1}{p(x)} = O(H(X|Y=y));$$

above, the first inequality follows from (i), and the second from (ii). □

Now suppose that Alice and Bob are given inputs $(\tilde{X}, \tilde{Y})$, and Alice wishes to send message $m = M(\tilde{x})$ to Bob which is the result of applying the function M to her input $\tilde{x}$. Suppose that this reveals I bits of information, i.e., that

$$I(\tilde{X} : M(\tilde{X})|\tilde{Y}) = H(M(\tilde{X})|Y) = I.$$

Then using the previous theorem (with $X = M(\tilde{X})$), we get a protocol that succeeds in communicating message m to Bob, using only, on average, a constant number of rounds and $O(I+1)$ bits of communication. More formally:

Corollary C.6. *Let $M : \mathcal{X} \to \mathcal{M}$ be any deterministic one-way protocol, and let μ be the distribution of the inputs $(\tilde{X}, \tilde{Y})$. Then there exists a deterministic protocol σ that makes use of a membership box, in a way such that:*

1. *The average communication of σ w.r.t. the distribution μ is*

$$\mathsf{ACC}_\mu(\sigma) = O(I(\tilde{X} : M(\tilde{X})|\tilde{Y}) + 1);$$

2. *The protocol σ uses the membership box once; and*

3. *After σ is run on the inputs x, y, both players know $M(x)$.*

Now we proceed to show compression for a general protocol π with q rounds. Let us begin by assuming that π is deterministic.

Theorem C.7. *Let π be any deterministic q-round protocol, and let μ be the distribution of the inputs (X, Y). Then there exists a randomized protocol $\tilde{\rho}$ that makes use of a membership box to achieve the following properties.*

1. *The average communication of $\tilde{\rho}$ is $\mathsf{ACC}_\mu(\tilde{\rho}) = O(\mathsf{IC}_\mu(\pi) + q)$;*

2. *The average number of rounds of $\tilde{\rho}$ is $\mathsf{ARC}_\mu(\tilde{\rho}) = O(q)$;*

3. *$\tilde{\rho}$ uses the membership box q times; and*

4. *After $\tilde{\rho}$ is run on the inputs x, y, both players know $\pi(x, y)$.*

Proof. Let π be a given deterministic q-round protocol. We define a protocol $\tilde{\rho}$ that simulates π by using the one-round compression of Corollary C.6 on a round-per-round basis. For a tuple $m_{<j} = (m_1, \ldots, m_{j-1})$ of $j-1$ strings, let $\pi_{j,m_{<j}}(x, y)$ denote the message communicated on the j-th round of π, for inputs x and y and supposing that $m_{<j}$ is the content of the messages communicated in the previous rounds. Then $\pi_{j,m_{<j}}(x, y)$ is either a function of x or of y, depending on whose turn of communicating it is. Let $\pi_{<j}(x, y) = (\pi_1(x, y), \ldots, \pi_{j-1,m_{<j-1}}(x, y))$ be the content of the first $j-1$ rounds of π when run on inputs x, y.

Suppose that j rounds have been simulated by $\tilde{\rho}$, and that the messages $m_1 = \pi_1(x, y), \ldots, m_{j-1} = \pi_{j-1,m_{<j-1}}(x, y)$ have been agreed upon (by using the membership box). Let $\tilde{X}$ and $\tilde{Y}$ be Alice and Bob's inputs conditioned on these events; i.e.,

$$(\tilde{X}, \tilde{Y}) = (X, Y)|\pi_{<j}(X, Y) = m_{<j}.$$

Suppose that it is Alice's turn to communicate, so that $\pi_{j,m_{<j}}(x, y)$ is a function of x, say $M(x)$. Then Alice sends the message $M(x)$ to Bob by running the protocol σ of Corollary C.6, for the inputs $\tilde{X}$ and $\tilde{Y}$. This will cost her an average of

$$\mathsf{ACC}_\mu(\sigma) = O(I(\tilde{X} : M(\tilde{X})|\tilde{Y}) + 1)$$

bits of communication, $\mathsf{ARC}_\mu(\sigma) = O(1)$ rounds and a single use of the membership box. If it is Bob's turn to communicate, i.e. $\pi_{j,m_{<j}}(x, y) = M(y)$, then they reverse their positions, and the average communication cost will instead be

$$\mathsf{ACC}_\mu(\sigma) = O(I(\tilde{Y} : M(\tilde{Y})|\tilde{X}) + 1).$$

In either case, after executing σ both players will have agreed on the message $\pi_{j,m_{<j}}(x,y)$ of round j, while using an average of $O(1)$ rounds, a single call to the membership box, and

$$O(I(\tilde{X} : M(\tilde{X})|\tilde{Y}) + I(\tilde{Y} : M(\tilde{Y})|\tilde{X}) + 1)$$

bits of communication, which is (a constant multiple of) the information cost of round j of π, conditioned on $\pi_{<j}(X,Y) = m_{<j}$. The theorem now follows from the chain rule (Fact A.1). □

Theorem 6.2 now follows as an easy corollary.

Proof of Theorem 6.2. To compress a public-coin protocol π using a membership box, Alice and Bob pick public randomness according to π's public randomness distribution, and then run the protocol of the previous theorem that simulates π_r on distribution μ using a membership box. This results in a protocol $\tilde{\rho}$ that has average communication $\mathsf{ACC}_\mu(\tilde{\rho}) = O(\mathsf{IC}_\mu(\pi) + q)$, average number of rounds $\mathsf{ARC}_\mu(\tilde{\rho}) = O(q)$, and uses the membership box q times.

By Corollary C.4, $\tilde{\rho}$ can be replaced by a protocol ρ with $\mathsf{ACC}_\mu(\rho) = O(\mathsf{IC}_\mu(\pi) + q \log \frac{2q}{\delta})$, and $\mathsf{ARC}_\mu(\rho) = O(q)$, that simulates $\tilde{\rho}$, and thus π, with error δ. This concludes the proof. □

D Proofs for Section 4

Proof of Lemma 4.2. We show the existence of such a graph using a probabilistic argument. Let $A = \{u_1, \ldots, u_{kN}\}$ and $B = \{v_1, \ldots, v_N\}$. Construct a random graph G by choosing d random neighbors independently for each $u \in A$. For any $A' \subseteq A$ of size N, let $E_{A'}$ be the event that $G_{A' \cup B}$ does *not* have a matching of size $N(1-\delta)$, and let $\textsc{Bad} := \bigvee_{A'} E_{A'}$. Note that the lemma holds if $\Pr[\textsc{Bad}] < 1$.

Next, we bound $\Pr[E_{A'}]$. Let $\mathcal{N}(u)$ denote the neighborhood of a vertex u. Consider the following procedure for generating a matching for $G_{A' \cup B}$:

FIND-MATCHING
1 $M \leftarrow \emptyset$
2 $V \leftarrow \emptyset$
3 **for** $i \leftarrow 1$ **to** N
4 **if** $\mathcal{N}(u_i) \not\subseteq V$
5 pick arbitrary $v_i \in \mathcal{N}(u_i) \setminus V$
6 $M \leftarrow M \cup \{(u_i, v_i)\}$
7 $V \leftarrow V \cup \{v_i\}$
8 **return** M

Let $X_1, \ldots, X_N$ be indicator variables for the event that the matching increased at step i, and let $Y_1, \ldots, Y_N$ to be i.i.d. random coins with $\Pr[Y_i = 1] = e^{-d\delta}$. Define $\textsc{Bad}_{A'}$ to be the event that $\sum_i X_i < N(1-\delta)$. In other words, $\textsc{Bad}_{A'}$ is the event that FIND-MATCHING fails to return a large enough matching for $G_{A' \cup B}$. For any i, the matching fails to increase

at step i only when all neighbors of u_i have already been matched. It follows that

$$\Pr[X_i = 0] = \left(\frac{\sum_{j<i} X_j}{N}\right)^d .$$

Furthermore, assuming that a large matching has not been found by step i, we have

$$\Pr[X_i = 0] < (1-\delta)^d < \Pr[Y_i = 1] . \tag{3}$$

In fact, we claim the following.

Claim 2. $\Pr[E_{A'}] \le \Pr[B_{AD_{A'}}] \le \Pr[\sum_i Y_i > \delta N]$.

It remains to bound this latter probability. We use the following claim, with $p := e^{-d\delta}$.

Claim 3. *Let $Y_1, \ldots, Y_N$ be i.i.d. biased coins, with $\Pr[Y_i = 1] = p < \delta < 1$. Then,*

$$\Pr\left[\sum Y_i > \delta N\right] < \exp(\delta N(1 + \ln(p/\delta))) .$$

Next, we bound the number of subsets $A' \subset A$ of size N, with the following claim.

Claim 4. *There are at most $\exp(N(1+\ln k))$ subsets of A of size N.*

Taking the two claims together, we have

$$\begin{aligned}
\Pr[BAD] &\le \exp\left(N(1+\ln k)\right) \cdot \exp\left(\delta N(1+\ln(p/\delta))\right) \\
&= \exp\left(N + N\ln k + \delta N + \delta N \ln(1/\delta) + \delta N \ln p\right) \\
&= \exp\left(N + N\ln k + \delta N + \delta N \ln(1/\delta) - d\delta^2 N\right) \\
&< 1 ,
\end{aligned}$$

where the final inequality uses $d = (2+\ln k)/\delta^2 + \ln(1/\delta)/\delta$. □

Now let us prove the claims.

Proof of Claim 2. For a string $x \in \{0,1\}^N$, let $x_{\le i}$ denote the substring $x_1 \cdots x_i$, and call x bad if $|x| < N(1-\delta)$. For $0 \le j \le n$, consider the random variable

$$D^{(j)} = X_1 \ldots X_j (1 - Y_{j+1}) \ldots (1 - Y_N).$$

Now notice that for any string v of length i, it holds that $\Pr[D^{(i)}_{\le i} = v] = \Pr[D^{(i+1)}_{\le i} = v]$. We have two cases:

- If $|v| \ge N(1-\delta)$, then

$$\Pr[D^{(i)} \text{ is bad} | D^{(i)}_{\le i} = v] = \Pr[D^{(i+1)} \text{ is bad} | D^{(i+1)}_{\le i} = v] = 0;$$

- If $|v| < N(1-\delta)$, then from equation (3) we get

$$\begin{aligned}
\Pr[D^{(i)}_{i+1} = 1 | D^{(i)}_{\le i} = v] &= \Pr[Y_{i+1} = 0 | \vec{X}_{\le i} = v] \\
&> \Pr[X_{i+1} = 1 | \vec{X}_{\le i} = v] \\
&= \Pr[D^{(i+1)}_{i+1} = 1 | D^{(i+1)}_{\le i} = v] = 0.
\end{aligned}$$

So in either case we conclude that

$$\Pr[D^{(i+1)} \text{ is bad}] \leq \Pr[D^{(i)} \text{ is bad}],$$

and the claim follows. □

Proof of Claim 3. Let $Y := \sum Y_i$, and let $\mu := \mathbb{E}[Y]$. Note that $\mu = pN$. Also, let $\psi := \delta/p - 1$. Using the multiplicative version of the Chernoff bound, we have

$$\begin{aligned}
\Pr[\textstyle\sum Y_i > \delta N] &= \Pr[Y > pN \cdot (\delta/p)] \\
&= \Pr[Y > \mu(1+\psi)] \\
&< \left(\frac{e^{\psi}}{(1+\psi)^{(1+\psi)}}\right)^{\mu} \\
&= \exp\left(\mu\left(\frac{\delta}{p} - 1 - \frac{\delta}{p}\ln(\frac{\delta}{p})\right)\right) \\
&= \exp\left(pN\frac{\delta}{p}\left(1 - \frac{p}{\delta} - \ln\delta + \ln p\right)\right) \\
&= \exp\left(\delta N - pN + \delta N\ln(1/\delta) + \delta N \ln p\right) \\
&< \exp\left(\delta N\left(1 + \ln(1/\delta) + \ln p\right)\right) .
\end{aligned}$$

□

Proof of Claim 4. There are $\binom{kN}{N}$ subsets of A of size N. By Stirling's Formula, we have

$$\binom{kN}{N} \leq \frac{(kN)^N}{N!} \leq \left(\frac{kNe}{N}\right)^N = \exp(N(1+\ln k)) .$$

□

Proof of Theorem 4.1. Let $k = \log|\mathcal{X}|$ and $N = |\mathcal{R}|$. Assume without loss of generality that $\mathcal{M} = M(\mathcal{X}, \mathcal{R})$; then $|\mathcal{M}| \leq 2^k N$. Now let G be $(2^k N, N, d, \delta)$-matching graph having $\mathcal{M}$ as its left set and $\mathcal{R}$ as its right set, for $\delta = \frac{1}{2k^2}$. For these parameters, we are assured by Lemma 4.2 that such a matching graph exists having left-degree $d = O(k^5)$.

We construct the new protocol M' as follows:

For each $x \in \mathcal{X}$ let $\mathcal{M}_x = M(x, \mathcal{R})$ be the set of messages that might be sent on input x. Noticing that $|M_x| = N$, consider a partial G-matching between $\mathcal{M}_x$ and $\mathcal{R}$ pairing all but a δ-fraction of $\mathcal{M}_x$; then define a bijection $\phi_x : \mathcal{R} \to \mathcal{M}_x$ by setting $\phi_x(r) = m$ if $\{m, r\}$ is an edge in the matching, and pairing the unmatched m and r's arbitrarily (possibly using edges not in G). Finally, set $M'(x, r) = \phi_x(r)$.

Since $M'(x, r) = M(x, \sigma(r))$ for some permutation σ of $\mathcal{R}$, it is clear that M and M' generate the same transcript distribution for any input x.

Now we prove that M' does not reveal much more information than M. Let the input $(\tilde{X}, \tilde{Y})$ be given by an arbitrary distribution μ. It holds that

$$I(\tilde{X} : M', R|\tilde{Y}) - I(\tilde{X} : M|\tilde{Y}) = \mathbb{E}_{y \in \tilde{Y}}[I(\tilde{X} : M', R|\tilde{Y} = y) - I(\tilde{X} : M|\tilde{Y} = y)].$$

Hence the result follows if we simply let X denote the random variable $\tilde{X}|(\tilde{Y} = y)$, for an arbitrary y, and prove that $I(X : M', R) - I(X : M) = O(\log k)$. Since $I(X : M', R) - I(X : M) = H(X|M) - H(X|M', R)$, the result will follow if $H(X|M', R) \geq H(X|M) - O(\log k)$.

A triple $(x, r, M(x, r))$ will be called a *cell* of message $m = M(x, r)$. A given pair $\{m, r\}$ will be called *good* when $\{m, r\}$ is an edge of G, and a cell (x, r, m) is called *good* if $\{m, r\}$ is good. Also, let us call a message m *good* if its good cells make up at least a $1 - \frac{1}{k}$ fraction of the probability mass of m. We will say "bad" as a shorthand for "not good." The following claim will be proven later:

Claim 5. *For our choice of parameters,* $\Pr[M'(X, R)$ *is bad*$] < \frac{1}{k}$.

Now, if R_m denotes the random variable R conditioned on $M'(X, R) = m$, then

$$H(X|M', R) = \mathbb{E}_{m \sim M'(X,R)}[H(X|M' = m, R_m)]. \tag{4}$$

For each fixed m, the right-hand entropy equals

$$H(X|M' = m, R_m) = H(X|M' = m) - I(X : R|M' = m). \tag{5}$$

But $I(X : R|M' = m) = H(R_m) - H(R_m|M' = m, X) = H(R_m)$, because since M' is 1–1, if we know m and x then r is completely determined. Now because M and M' are equidistributed for every x, then from (4) and (5), we get:

$$H(X|M', R) = H(X|M) - H(R|M').$$

All that is left to do is bound $H(R|M')$. For any fixed m we have $H(R_m) \leq k$, because r is a function of m (which is given) and x (which is k-bits long). Hence,

$$H(R|M') \leq \Pr[M' \text{ is good}]\, \mathbb{E}_{\text{good } m}[H(R_m)] + \Pr[M' \text{ is bad}]k.$$

And for good m,

$$H(R_m) \leq \Pr[\{m, R_m\} \text{ is good}] H(R_m|\text{good } \{m, R_m\}] + \Pr[\{m, R_m\} \text{ is bad}]k + 1.$$

We now have that $\Pr[M'$ is bad$]$ and $\Pr[\{m, R_m\}$ is bad$]$ are both less than $\frac{1}{k}$, by Claim 5 and assuming that m is good, respectively. Furthermore, conditioned on $\{m, R_m\}$ being good, the support of R_m is at most $d = O(k^5)$, and hence $H(R_m|\text{good } \{m, R_m\}]$ is at most $\log d = O(\log k)$; hence we obtain

$$H(X|M', R) \geq H(X|M) - O(\log k).$$

□

Proof. (Claim 5) Suppose that $\Pr[M'(X, R)$ is bad$] > \frac{1}{k}$. Then the probability that $(X, R, M'(X, R))$ is a bad cell is at least

$$\Pr[M'(X, R) \text{ is bad}] \Pr[(X, R, M'(X, R)) \text{is bad} | M'(X, R) \text{ is bad}] > \frac{1}{k^2}.$$

But then there must exist a choice of x such that $\Pr[(x, R, M'(x, R))$ is bad$] > \frac{1}{k^2}$, which implies that, for this x, there is a $\frac{1}{k^2}$ fraction of the $(r, M'(x, r))$ pairs that are not edges of G, and hence not part of the matching. But this contradicts the fact that our matching gives at most a $\delta < \frac{1}{k^2}$ fraction of unmatched vertices. □

E Proofs of Section 5

Proof of Theorem 5.1. We think of $M(\cdot,\cdot)$ as a table, which we will call *the M-table*, where the inputs $x \in \mathcal{X}$ are the rows and the random choices $r \in \mathcal{R}$ are the columns, and fix some ordering $r_1 < r_2 < \ldots$ of $\mathcal{R}$. The second part $J(x,r)$ of $\tilde{M}$ will be set to the number of times $M(x,r)$ appeared in the same row up to the column r, i.e.,

$$J(x,r) = |\{r' \leq r | M(x,r') = M(x,r)\}|.$$

This ensures that $\tilde{M}$ is 1–1. From this point onwards, let us fix the message m, as well as Bob's input y, and let X_m denote the random variable $X|M(X,R) = m, Y = y$, and J_m denote $J(X,R)|M(X,R) = m, Y = y$. Notice that conditioned on a given fixed $X_m = x$, the distribution J_m is uniform (because the randomness is picked uniformly). The supports of X_m and J_m will be denoted $\mathcal{X}_m$ and $\mathcal{J}_m$, respectively. We will settle the theorem by proving that the following holds, regardless of our choice of m or y:

$$I(X_m : J_m) \leq \log\log|\mathcal{J}_m| + 1.$$

The full result will then follow by averaging over y and m. To prove this bound, let $w_x = |\{r \in \mathcal{R} | M(x,r) = m\}|$ be the number of m-entries in row x, and partition $\mathcal{X}_m$ into *blocks* $\mathcal{X}_1, \ldots, \mathcal{X}_\ell$ such that, for any x, x' in the same block, $w_x \leq 2w_{x'}$. This can be done with $\ell \leq \log|\mathcal{J}_m|$. Now let the random variable B be the block $\mathcal{X}_B$ that X_m belongs to, and for $b = 1, \ldots, \ell$, let $W_b = max\{w_x | x \in \mathcal{X}_b\}$ be the size of the support of J_m conditioned on $B = b$.

Then by the information processing inequality, and the chain rule, we find that

$$I(X_m : J_m) \leq I(X_m B : J_m) = I(B : J_m) + I(X_m : J_m | B).$$

To begin, we know that $I(B : J_m) \leq \log\ell \leq \log\log|\mathcal{J}_m|$. Now, given any block $B = b$, $I(X_m : J_m | B = b) = H(J_m | B = b) - H(J_m | B = b, X_m)$. And we also know that:

$$H(J_m | B = b, X_m = x) = \log w_x \geq \log W_b - 1 \geq H(J_m | B = b) - 1,$$

where the first equality follows because J_m is uniformly distributed on any given row $X_m = x$, the first inequality follows because if $x \in \mathcal{X}_b$, then $w_x \geq W_b/2$, and the second inequality follows because W_b is the size of the support of $J_m | B = b$. Hence we conclude that $I(X_m : J_m | B = b) \leq 1$. □

In the full version of the paper we will present a lower bound of $\log\log|J_m| - O(1)$ for a specific pair X_m, J_m, implying that no improvement of Theorem 5.1 is possible without changing $J(x,r)$.