Interactive Compression to External Information

Mark Braverman[*]
Princeton University
Princeton, NJ, USA
mbraverm@cs.princeton.edu

Gillat Kol[†]
Princeton University
Princeton, NJ, USA
gillat.kol@gmail.com

ABSTRACT

We describe a new way of compressing two-party communication protocols to get protocols with potentially smaller communication. We show that every communication protocol that communicates C bits and reveals I bits of information about the participants' private inputs to an observer that watches the communication, can be simulated by a new protocol that communicates at most $poly(I) \cdot \log\log(C)$ bits. Our result is tight up to polynomial factors, as it matches the recent work separating communication complexity from external information cost.

CCS CONCEPTS

• **Theory of computation → Communication complexity**;

KEYWORDS

Interactive compression, Communication complexity, External information cost, Information theory, Correlated sampling

ACM Reference Format:
Mark Braverman and Gillat Kol. 2018. Interactive Compression to External Information. In *Proceedings of 50th Annual ACM SIGACT Symposium on the Theory of Computing (STOC'18).* ACM, New York, NY, USA, 14 pages.
https://doi.org/10.1145/3188745.3188956

1 INTRODUCTION

Seminal works by Shannon and Huffman [13, 19] considered the *data compression problem* and showed how to optimally compress one-way communication to its information content, measured by the entropy of the sent message. The corresponding problem for interactive communication, called the *interactive compression problem*, has attracted a lot of attention over the past decade. Roughly speaking, the interactive compression problem asks whether a protocol with a low information content I can be simulated by a protocol that only communicates roughly I bits [3].

1.1 Information Cost

The interactive compression problem is formalized in the setting of (distributional) two-party communication complexity. In this setting, each party gets a private input, where the inputs are sampled from a joint distribution μ. The parties engage in an interactive communication protocol in order to perform some communication task that depends on both inputs.

To measure the information content of an interactive protocol, we use the notion of *information cost*, which can be viewed as a generalization of the entropy function [1–3, 10, 14, 17]. In this work we focus on the *external information cost* measure. Roughly speaking, the external information cost of a protocol π over the distribution μ is the amount of information that an external observer, who witnesses the execution of π, learns about the parties' inputs, when the inputs are sampled from μ. More formally,

*Definition 1.1 (**External Information Cost**).* The *external information cost* of a two-party protocol π over random inputs (X, Y) that are drawn according to a joint distribution μ, is defined as $\mathsf{IC}_\mu(\pi) = \mathbb{I}(\pi(X,Y);(X,Y))$, where $\mathbb{I}$ stands for mutual information and $\pi(X, Y)$ is the transcript of π when it is run with inputs X, Y.

1.2 Our Results

In this paper we study the problem of compressing a protocol to its external information cost. This problem asks whether every protocol π with external information cost I over a distribution μ can be simulated by a protocol that only communicates roughly I bits. By "simulate" we mean that the new protocol performs the same task as π, except with some small error probability, where the probability is over μ and over the randomness used by the players.

The most relevant previous works are a compression protocol by [3] and a separation result by [11] described next. The influential [3] paper, shows how to compress every protocol that communicates at most C bits and has external information cost I over μ, to a protocol that only communicates $I \cdot \mathrm{polylog}(C)$ bits. The question of whether some dependence on C is inherent is interesting, as C may be arbitrarily larger than I, thus making the polylog(C) term the "costly" term.

This question was recently answered in the affirmative by [11], analyzing a communication task T and a distribution μ, parameterized by a parameter k, suggested by [6]. It is shown that there exists a protocol π that solves T and has $\mathsf{IC}_\mu(\pi) = O(k)$, while every protocol solving T must communicate at least $2^{\Omega(k)}$ bits. Thus, proving an exponential separation between communication complexity and external information cost.

We mention that the protocol π with low external information communicates C bits, where C is triple exponential in k. Since, in this case, $\mathrm{poly}(I) \cdot (\log\log C)^{o(1)}$ is $2^{o(k)}$, this result implies that

[*] Research supported in part by NSF Awards DMS-1128155 and CCF-1525342, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.
[†] Research supported by National Science Foundation grant No. CCF-1750443.

STOC'18, June 25–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5559-9/18/06…$15.00
https://doi.org/10.1145/3188745.3188956

there exists a communication protocol with external information I and communication complexity C that cannot be compressed to a protocol with communication complexity $\mathrm{poly}(I) \cdot (\log\log C)^{o(1)}$.

The problem of closing the gap between the upper and lower bounds for interactive compression with respect to the external information cost measure, and finding the right dependence on C, was left open. It was conjectured that a logarithmic dependence on C is essential for external compression, and that no compression to $\mathrm{poly}(I) \cdot o(log(C))$ is possible (see Open Problem 6.2 in [21]). Our main result shows that such a scheme is possible, and, in fact, one can get a double-logarithmic dependence on C. Our result is essentially tight up to polynomial factors, by [11].

Theorem 1.2. *Let $0 < \varepsilon < 1/2$ be given constant. Fix any public or private coin protocol π with input space $\mathcal{X} \times \mathcal{Y}$. Let μ be a distribution on $\mathcal{X} \times \mathcal{Y}$. Let $I = \mathsf{IC}_\mu(\pi)$ and C be the maximum number of bits communicated by π. Then, there is a public-coin protocol τ that simulates π with error ε under μ and communicates at most $poly(I) \cdot \log\log C$ bits.*

Sketch of the proof of Theorem 1.2 can be found in Section 2. The rest of the paper is devoted to proving Theorem 1.2, however, due to space constraints, some of the proofs are omitted from this version.

1.3 Additional Related Works

Internal Information Cost. A related information cost measure studied in the literature is the *internal information cost* measure. Roughly speaking, the internal information cost of a protocol π over the distribution μ, is the amount of information that the *parties* learn about each other's input by running π, when the inputs are sampled from μ.

The definition of internal information cost by the theoretical computer science community was motivated by the quest for good communication complexity lower bounds, and by fascinating relations to the direct sum problem in communication complexity [2, 3]. A similar definition appeared earlier in the information theory literature, in the context of interactive communication [14, 17].

Interactive Compression Protocols. The interactive compression problem with respect to the internal information cost measure was the focus of many recent works, and beautiful compression protocols were suggested [3–5, 7, 9, 12, 15, 18, 20]. The general case of compressing any protocol over any distribution is considered in [5], where a $2^{O(I)}$ compression is given, and in [3], where a $\tilde{O}(\sqrt{C \cdot I})$ compression is given. Here I stands for the *internal* information cost of the protocol.

The important special case of compressing interactive protocols over a product distribution μ (the inputs to the players are independent), was considered by a recent line of works [3, 15, 20], resulting in an almost optimal $O(I \cdot \mathrm{polylog}(I))$ compression scheme. It is not hard to show that the external information cost is always an upper bound on the internal information cost, as the observer is less informed than the parties to begin with, thus may only learn more from the interaction. Over a product distribution, the internal and external information costs coincide, because the first party is as informed as the observer about the second party's input, and vise versa.

2 PROOF SKETCH

Let π be a randomized communication protocol between two players, Alice and Bob. Alice has a private input x and Bob has a private input y, where (x, y) is chosen according to some publicly known joint distribution μ. We next sketch a public coin protocol τ that simulates π, and has communication complexity $\mathrm{poly}(I) \cdot \log\log(C)$, where $I = \mathsf{IC}_\mu(\pi)$ and C is the maximum number of bits communicated by π. Our simulation protocol builds over the works of [3, 15].

2.1 Communication Tree

Consider the (directed) binary tree associated with π. Each vertex v of the binary tree corresponds to a possible transcript of π. The two edges going out of v are labeled by 0 and 1, corresponding to the next bit to be transmitted. We think of each non-leaf vertex as owned by one of the parties. The protocol π proceeds as follows: Starting from the root, when π reaches a non-leaf vertex v, the player who owns v sends a bit to the other player. The players follow the edge indicated by the sent bit and reach a new vertex. Note that since π communicates at most C bits, the depth of this tree is at most C.

We denote the set of vertices of this tree by $\mathcal{V}$ and the set of leaves by $\mathcal{L}$. For $v, u \in \mathcal{V}$, we write $v \preceq u$ if v is an ancestor of u. We write $v \prec u$ if $v \preceq u$ and $v \neq u$. We write $u \succeq v$ if $v \preceq u$. We write $u \succ v$ if $u \succeq v$ and $v \neq u$. We also use $\preceq$ as a unary operator and denote by $\preceq v$ the set $\{v' \in \mathcal{V} : v' \preceq v\}$.

Let $v, w \in \mathcal{V}$ such that $w \succeq v$. Denote by $P_{v,x,y}(w)$ the probability that π reached w, conditioned on reaching v and on players' inputs being x and y. Denote by $P_{v,x}(w)$ the probability that π reached w, conditioned on reaching v and on Alice's input being x. Denote by $P_{v,y}(w)$ the probability that π reached w, conditioned on reaching v and on Bob's input being y. Denote by $P_v(w)$ the probability that π reached w, conditioned on reaching v. When we omit v, we mean that v is the root. We also view $P_{x,y}, P_x, P_y, P$ as distributions over the leaves. For example, $P_{x,y}$ assigns the probability $P_{x,y}(u)$ to the leaf $u \in \mathcal{L}$. We say that τ simulates π if τ samples a leaf u according to the "correct" distribution $P_{x,y}$.

2.2 Information Frontiers

A *frontier* or a *cut* in the transcript tree is any subset of the tree's vertices that intersects every root-to-leaf path in exactly one vertex. For example, the leaves of the tree form a frontier.

We define the information frontier $\mathcal{F}_x$ to be the set of all vertices $w \in \mathcal{V}$ for which $\log(P_x(w)/P(w)) \notin (-1, 1)$, but for w', the parent of w, it is the case that $\log(P_x(w')/P(w')) \in (-1, 1)$. Intuitively, $w \in \mathcal{F}_x$ if the bits communicated by Alice from the beginning of the protocol until reaching w give roughly 1 bit of information to an external observer about x. We mention that we define the information frontiers differently than in all previous papers, where a divergence based definition was used. Our new definition helps simplify the proof and can also result in some savings when analysing the communication complexity of simulating protocols.

An important observation by [3] is that Alice knows the frontier $\mathcal{F}_x$, as she knows x and can compute $\log(P_x(w)/P(w))$ for every $w \in \mathcal{V}$. This means that Alice can measure the amount of information she gives the observer, as he starts with no information about x. Similarly, we can define the frontier $\mathcal{F}_y$ and claim that Bob knows $\mathcal{F}_y$.

For simplicity (and with loss of generality), let us assume that the frontier $\mathcal{F}_y$ is always above the frontier $\mathcal{F}_x$ (in particular, $\mathcal{F}_x$ and $\mathcal{F}_y$ do not "intersect").

2.3 A General Scheme for τ

Let us consider the following rough scheme for the protocol τ simulating π, based on [3, 15]:

(1) **Correlated Sampling:** Players jointly sample a leaf u according to the probability distribution P_y.
(2) **Finding Separation:** Alice computes $a = \mathcal{F}_x \cap (\preceq u)$ and Bob computes $b = \mathcal{F}_y \cap (\preceq u)$. Players find a vertex w such that $b \preceq w \preceq a$.
(3) **Rejection Sampling:** Alice accepts w with probability $P_x(w)/P(w)$. If Alice rejects w, players go back to the first step.
(4) **Iterate:** Players simulate the protocol induced by the subtree rooted at w by going back to the first step with w as the root (and possibly changing the roles of Alice and Bob).

We next explain how this protocol can be implemented and why it is working.

2.3.1 Implementing the First and Third Steps. To implement the first step of τ we first note that if both players know P_y then they can sample u from the public randomness with no communication. However, Alice does not know y, and therefore, does not know P_y. To still be able to jointly sample from P_y with low communication cost, the players run the *CorrelatedSampling* protocol suggested by [7]. *CorrelatedSampling* assumes that Alice knows a distribution P' over some domain Ω, and Bob knows (a possibly different) distribution Q' over Ω. After executing $CorrelatedSampling(P', Q')$, both players know an element $\omega \in \Omega$ sampled according to P'. The communication required by this protocol is roughly the distance between P' and Q' measured by the KL-divergence $\mathbb{D}(P' \| Q')$. To implement the first step of τ, players run $CorrelatedSampling(P' = P_y, Q' = P)$. It can be shown that $\mathbb{D}(P_y \| P)$ is upper bounded by roughly I.

We discuss the implementation of the second step of τ later. For now assume that the second step was implemented and that a vertex $b \preceq w \preceq a$ was found with little communication. Since u was sampled according to P_y, the vertex w is obtained with probability $P_y(w)$. During the third step of τ, Alice accepts w with probability $P_x(w)/P(w)$. This means that w is the new root with probability $\frac{P_y(w) \cdot P_x(w)}{P(w)}$. Due to cancelations, this probability is actually $P_{x,y}(w)$. This means that w was sampled according to the correct distribution.

We can then repeat the process to correctly sample deeper and deeper vertices $w_1 \preceq w_2 \preceq \ldots$ in the communication tree, until reaching a leaf. Since the external information cost is I, it can be shown that we only need to cross roughly I information frontiers. Since we cross an information frontier in every iteration, τ ends after at most I iterations.

An important issue regarding the third step, is that a-priori it may be the case that $P_x(w)/P(w)$ is very small (w was over-sampled by the first step), causing Alice to almost always reject, or worst, $P_x(w)/P(w)$ may be greater than 1 (w was under-sampled). To show that $P_x(w)/P(w)$ is close to 1, we use the fact that w is above the frontier $\mathcal{F}_x$ (as $w \preceq a$), which means that $\log(P_x(w)/P(w)) \in (-1, 1)$, and thus $P_x(w)/P(w) \in (1/2, 2)$. Intuitively, less than one bit of information about x was revealed when reaching w, so $P_x(w)$ cannot be very different than $P(w)$.

2.3.2 Implementing the Second Step. It remains to reason about the implementation of the second step. The simplest implementation is to set $w = a$ by having Alice send a to Bob. Since the tree is of depth C, Alice needs to communicate roughly $\log(C)$ bits. Unfortunately, this is exactly the $\log(C)$ factor we are trying to avoid.

The Protocol Pred. Another possible implementation of this step is by recalling that given integers $c, d \in [n]$, players can agree on an integer k between c and d, by communicating only $\log\log(n)$ bits, as follows: Consider the binary representations $bin(c)$ and $bin(d)$ of c and d (respectively), and find the first coordinate i on which they disagree, by running a binary search. To do that, hash the first halves of $bin(c)$ and $bin(d)$ and compare the hashes. If the hashes match, i should be in the second half of the binary representations. Otherwise, i is in the first half. Recurse until finding i, and use i to compute k. Note that since the binary representation of a number in $[n]$ is $log(n)$ bits long, this protocol only requires roughly $\log\log(n)$ communication bits.

We can now construct the protocol $Pred(a, b)$ that outputs the desired w: We denote by c and d the levels of a and b in the tree (respectively). We find k between c and d as explained above, and return vertex number k on the path from the root to a. Note that since the tree is of depth C, it holds that $c, d \in [C]$ and the overall communication is about $\log\log(C)$.

The Double Counting (Oversampling) Problem. The main problem with this $\log\log(C)$ bits implementation of the second step (as opposed, for example, to the $\log(C)$ implementation by setting $w = a$), is that the set $\mathcal{F}$ of all vertices w that are produced for all possible leaves u selected by the first step, may not be a frontier. To show that $\mathcal{F}$ may not be a frontier, consider leaves $u, u' \in \mathcal{L}$ with lowest common ancestor v that is between the frontier $\mathcal{F}_y$ and the frontier $\mathcal{F}_x$. Let $a = \mathcal{F}_x \cap (\preceq u)$, $b = \mathcal{F}_y \cap (\preceq u)$ and let $a' = \mathcal{F}_x \cap (\preceq u')$, $b' = \mathcal{F}_y \cap (\preceq u')$. Since v is below $\mathcal{F}_y$, it is the case that $b = b'$. However, since v is above $\mathcal{F}_x$ it is possible that, say, a' is much higher in the tree than a. In this case, $Pred(a', b')$ may return a vertex w' such that $w' \prec v$, while $Pred(a, b)$ returns a vertex w such that $v \prec w \preceq a$. Since $w, w' \in \mathcal{F}$, but $w' \prec w$, $\mathcal{F}$ is not a frontier.

The problem with $\mathcal{F}$ not being a frontier is that it can lead to "double counting" or oversampling. What is the probability that τ reaches w? One way of reaching w is to reach it directly from the root, as discussed above. Another way of reaching w is to reach w' from the root, and then in the second iteration of τ, designated to simulating the subtree rooted at w', reach w from the new root w'. The fact that there are various ways of getting to w distorts the probability of reaching w.

Therefore, in order for τ to work, we either have to implement the second step in a way that induces a frontier, or deal with the double counting problem. When the distribution μ is a product distribution, the first option turned out to be possible [15, 20]. For general distributions, it is not clear how to construct such frontiers, and we are instead taking the second approach. As far as we know, this work is the first to handle double counting.

2.4 Our Compression Protocol

The Protocol Mid. Let us first consider all of Bob's information frontiers $\mathcal{F}_{i,y}$. Informally, the frontier $\mathcal{F}_{i,y}$ is the set of vertices where roughly i bits of information about y are revealed to the observer (in particular, $\mathcal{F}_{1,y} = \mathcal{F}_y$). Let us again assume for simplicity that Alice's and Bob's frontiers do not intersect. Observe that there may be any number of frontier $t \le I$ such that $\mathcal{F}_{1,y}, \ldots, \mathcal{F}_{t,y}$ are all above $\mathcal{F}_x$.

Our protocol implements the second step of τ in a way that ensures that w does not only separate $\mathcal{F}_y$ and $\mathcal{F}_x$, but also separates all the frontiers $\mathcal{F}_{i,y}$ that are above $\mathcal{F}_x$ from $\mathcal{F}_x$. That is, let u be the leaf obtained by the correlated sampling step. Denote $b_i = \mathcal{F}_{i,y} \cap (\preceq u)$ and let $a = \mathcal{F}_x \cap (\preceq u)$. Then w is such that $b_1 \preceq b_2 \preceq \ldots \preceq b_t \preceq w \preceq a$.

To find such a w, we run the following protocol $Mid(u) = Mid(b_1, \ldots, b_t, a)$: The protocol Mid first executes $Pred$ to find a w_1 that separates b_1 from a. Let j be such that $b_j \preceq w \preceq b_{j+1}$. Then, Mid executes $Pred$ to find a w_2 that separates b_{j+1} from a. This process proceeds in at most I iterations.

Consistent Leaves. Assume that the first iteration of τ reaches the vertex w. Then, the second iteration of τ is aimed at simulating the protocol induced by the subtree rooted at w. In the case where the second step of the first iteration produces a frontier $\mathcal{F}$ (with $w \in \mathcal{F}$), the second iteration does not need to know the transcript of the first iteration that led to the vertex w in the first place in order to simulate the sub-protocol. However, if the set $\mathcal{F}$ produced by the first iteration is not a frontier, then, to avoid double counting, it will be important that the second iteration of τ "remembers" the transcript of the first iteration and only samples leaves that are "consistent" with it.

More formally, suppose that in the first iteration of τ, the leaf u was sampled by the correlated sampling step, and suppose that the transcript of the $Mid(u) = Mid(b_1, \ldots, b_t, a)$ protocol run by this iteration is m, and its output is w. Then, in the second iteration of τ, simulating the subtree rooted at w, the correlated sampling step will only sample leaves u' that are consistent with m. A leaf u' is *consistent* with m if running $Mid(u')$ yields the transcript m. That is, we want the transcript of the execution $Mid(u') = Mid(b'_1, \ldots, b'_t, a')$ to be m, where $b'_i = \mathcal{F}_{i,y} \cap (\preceq u')$ and $a' = \mathcal{F}_{i,x} \cap (\preceq u')$. The intuition is that m contains all the information that τ "knew" about the leaf u that was sampled by the first iteration, and therefore u and u' look the same to τ. Hence, we may as well replace u by u'.

Let $\mathcal{U}$ be the set of all leaves u' that are consistent with m. We now need to revise the correlated sampling step of the second iteration of τ to sample leaves $u' \in \mathcal{U}$ according to the distribution $P_x|\mathcal{U}$ (the distribution P_x restricted to $\mathcal{U}$). Recall, however, that the correlated sampling procedure requires one of the players to know the correct distribution $P_x|\mathcal{U}$, while the other can have some good estimate of it. In our case, since the set $\mathcal{U}$ may depend on both x and y, it is not clear whether there is a player that knows the distribution $P_x|\mathcal{U}$.

We will next show that Alice "almost" knows this distribution. To do that, we first claim that $b'_1 = (\mathcal{F}_y \cap \preceq u') = (\mathcal{F}_y \cap \preceq u) = b_1$, as $b_1 \preceq w$ and $w \preceq u, u'$. The reason that $u \succeq w$ is that the vertex w obtained by the first iteration of τ is an ancestor of the leaf u sampled by its correlated sampling step. The reason that $u' \succeq w$ is that u' is sampled by the second iteration of τ, thus it is in the subtree rooted at w.

Now, assume that we have reached the second iteration of τ, and Alice wishes to compare the transcript of $Mid(u')$ to m, to decide if $u' \in \mathcal{U}$. (Note that m is known to Alice, as both players know the transcript of the first iteration). To do that, she runs $Mid(u')$ by herself, with no help from Bob, as described next. Alice first needs to execute $Pred(b'_1, a')$. Assume that the players take alternating turns in communicating bits while running $Pred(b'_1, a')$, and that Alice sends that first bit. Alice knows x, thus also knows a' and can compute the first communicated bit and compare it to the first bit of m. If she finds a mismatch, she concludes that $u' \notin \mathcal{U}$. Otherwise to continues to compare the next communicated bits.

The second bit should be sent by Bob, and Alice does not know y, so she cannot trivially compute it. Recall, however, that $b'_1 = b_1$, thus Bob has the same input in $Pred(b'_1, a')$ and $Pred(b_1, a)$. Because the transcript of $Pred(b_1, a)$ is induced by m, Alice knows what Bob's next message is (assuming that the first bits communicated by $Pred(b'_1, a')$ and $Pred(b_1, a)$ were the same). She can then use it to compute the third bit in the transcript of $Pred(b'_1, a')$, as this bit is sent by her. Alice continues this process and compares all the bits.

Intersecting Information Frontiers. The general case, where Alice's and Bob's frontiers may intersect, poses additional challenges. One is due to the fact that in this case it is possible that t, the number of Bob's frontiers that intersect the path to u above a, is smaller than t', the number of Bob's frontiers that intersect the path to u' above a'. In this case, the number of executions of $Pred$ by $Mid(u)$, denoted s, may be smaller than the number of executions of $Pred$ by $Mid(u')$. Let $\mathcal{U}'$ be the set of all leaves u' such that the transcript of the first s executions of $Pred$ by $Mid(u')$ is m. By the above argument, Alice can check if $u' \in \mathcal{U}'$. However, since the m may be a strict prefix of the transcript of $Mid(u')$, $\mathcal{U}$ may be a strict subset of $\mathcal{U}'$.

To overcome this problem, we note that given $u' \in \mathcal{U}'$, the players can check if $u' \in \mathcal{U}$ as follows: Let $\mathcal{F}_{j,y}$ be Bob's first information frontier after w and let $b'' = \mathcal{F}_{j,y} \cap (\preceq u')$. The leaf $u' \in \mathcal{U}'$ is not in $\mathcal{U}$ (call u a *bad* leaf) if and only if $b'' \preceq a$ (in this case, more executions of $Pred$ would have been required to separate all of Bob's frontiers from Alice's frontier).

This suggests the following protocol: Players will first jointly sample a leaf according to the distribution $P_x|\mathcal{U}'$ known to Alice, and then check if u' is bad by executing $Pred(a, b'')$ to check if $b'' \preceq a$. (Recall that $Pred$ only requires $\log\log(C)$ bits of communication, and thus is affordable.) If this is indeed the case, the leaf u' is bad and will be rejected. The the players will jointly sample a new leaf and check it. This will repeat until a good leaf is found.

Protocol Analysis. One problem that we need to deal with in order to ensure that our simulation is indeed communication efficient, arises if the leaf u' sampled from $P_x|\mathcal{U}'$ turns out to almost always be bad. In this case, we keep rejecting the sampled u', blowing up the communication substantially. To handle this case we note that u, the vertex sampled in the first iteration, is good. Thus, if good leaves are rare, it is very unlikely that u will be sampled by the first iteration.

Another issue that has to do with the communication cost of τ is the fact that we are now having the players jointly sample from the distribution $P_x|\mathcal{U}'$. While we were able to claim that $\mathbb{D}(P_x\|P) \approx I$, the same bound may not hold for $\mathbb{D}((P_x|U')\|P)$. However, it can be shown that $\mathbb{D}((P_x|U')\|P) \leq \mathbb{D}(P_x\|P) + |m| \lesssim I + I\log\log(C)$.

We also mention that we are not able to prove the accuracy of our simulation protocol τ using the techniques developed by previous papers. Those techniques basically show that every iteration of τ samples correctly form some frontier $\mathcal{F}$. However, we do not construct such frontiers. Instead, we develop a new analysis method based on the unique decomposition of leaves. This method crucially exploits the fact that for every leaf u, there is a single way for τ to reach u (assuming that the public randomness was fixed). That is, if w_i is the w vertex computed by iteration i of π, then there is a single sequence $w_1 \preceq w_2 \preceq \ldots \preceq w_\ell$ with $w_\ell = u$.

3 OUR COMPRESSION SCHEME

In this section we present the pseudo-code for our compression protocol. A more detailed explanation of the protocol, as well as the protocol's analysis, are deferred to the Appendix.

Algorithm 1: $Pred(v^A, v^B)$

```
$D \leftarrow \lceil \log(C) \rceil + 1$
/* fdiff finds the first difference between two bit strings */
Run $\mathrm{fdiff}_D(\mathrm{level}(v^A), \mathrm{level}(v^B))$, $T = O(\log(I)/\varepsilon^2)$ times
$i \leftarrow$ most common return value of fdiff
$p \leftarrow \begin{cases} 1 & \text{if } v_i^A = 0 \\ 0 & \text{if } v_i^A = 1 \end{cases}$
$W \leftarrow$ the vertex at level $(\mathrm{level}(v^A))_{1,i-1} \circ 1 \circ 0^{D-i}$ on the path from the root that goes through $v^A$ and $v^B$
return $(p, W)$
```

Algorithm 2: $Mid^A(X, Y, V, U)$

```
1 $W \leftarrow V$
2 repeat
3     $(p, W) \leftarrow Pred(\mathcal{F}_{W,X} \cap \preceq U, \mathcal{F}_{V,Y} \cap \preceq U)$
4 until $p = 0$
5 $W \leftarrow$ the second to last value of $W$
6 $M \leftarrow$ the transcript of all the executions of $Pred$ by Line 3, excluding the last one
7 return $(W, M)$
```

Algorithm 3: $\tau(X, Y)$

```
1  $V, V' \leftarrow root$, $M \leftarrow \varepsilon$, $leader \leftarrow Alice$, $\mathcal{U} \leftarrow \mathcal{L}$
2  repeat
3      if $leader = Alice$ then
           /* $U \leftarrow$ random vertex distributed according to $P_{\mathcal{L}(V),X}|M$ */
4          repeat
5              $P^A \leftarrow$ the probability distribution $\forall u \in \mathcal{U}: \; P^A(u) = \frac{P_{V,X}(u)}{P_{V,X}(\mathcal{U})}$
6              $U \leftarrow CorrelatedSampling(P = P^A, Q = P_{\mathcal{L}(V)})$
7          until $\left(Pred(\mathcal{F}_{V',X} \cap \preceq U, \mathcal{F}_{V,Y} \cap \preceq U)\right)_1 = 1$
           /* $W \leftarrow$ candidate for new $V$ */
8          $R \leftarrow$ new shared random string
9          $(W, M) \leftarrow Mid_R^A(X, Y, V, U)$
           /* rejection sampling */
10         $Z \leftarrow \mathcal{F}_{V,Y} \cap \preceq U$
11         Bob accepts w.p. $\frac{P_{V,Y}(Z)}{8P_V(Z)}$, in this case
12             $\mathcal{U} \leftarrow \{u \in \mathcal{L}(W) \; : \; (Mid_R^A(X, Y, V, u))_2 \geq M\}$
13             $leader \leftarrow Bob$
14             $V' \leftarrow V$
15             $V \leftarrow W$
16     else
17         Same, switching the roles of Alice and Bob and of $X$ and $Y$. In addition, Lines 9 and 12 refer to the protocol $Mid^B$ instead of $Mid^A$. The protocol $Mid^B$ is obtained from the protocol $Mid^A$ by switching the roles of Alice and Bob and of $X$ and $Y$.
18 until $V$ is a leaf
19 return $V$
```

4 NOTATION

We note that some of our notation is borrowed from [20], including the very nice string notation described below.

4.1 Strings

Recall that $\{0,1\}^*$ refers to the set of all binary strings. The empty string is denoted ε. Let $v, w \in \{0,1\}^*$. We denote by $|v|$ the length of v. For $i \leq j \in [|v|]$, we denote v_i the i^{th} bit of v and by $v_{i,j}$ the substring of v given by $v_i v_{i+1} \ldots v_j$. The concatenation of v and w is denoted $v \circ w$.

Consider the standard partial order $\preceq$ on $\{0,1\}^*$, where $v \preceq w$ if and only if $v \circ v' = w$ for a (possibly empty) string v'. If $v \preceq w$, we say that v is an ancestors of w and that w is a descendant of v. We write $v \prec w$ if $v \preceq w$ and $v \neq w$. We write $v \succeq w$ if $w \preceq v$. We write $v \succ w$ if $w \prec v$.

Let $\mathcal{S} \subseteq \{0,1\}^*$ and let $v \in \mathcal{S}$. We say that v is *minimal* in $\mathcal{S}$ if there does not exist $w \neq v \in \mathcal{S}$ such that $w \preceq v$. We say that v is *maximal* in $\mathcal{S}$ if there does not exist $w \neq v \in \mathcal{S}$ such that $v \preceq w$. Let $m \in \mathbb{N}$ and let $\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_m \subseteq \{0,1\}^*$. The floor of the sets $\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_m$, denoted $\lfloor \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m \rfloor$, is the set of minimal elements of $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_m$. Analogously, the ceiling of the sets $\mathcal{S}_1, \mathcal{S}_2, \dots \mathcal{S}_m$, denoted $\lceil \mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_m \rceil$, is the set of maximal elements of $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \cup \mathcal{S}_m$.

Let $v \in \{0,1\}^*$ and let $\mathcal{S} \subseteq \{0,1\}^*$. We write $v \preceq \mathcal{S}$ if there exists $w \in \mathcal{S}$ such that $v \preceq w$. Similarly, we write $v \succeq \mathcal{S}$ if there exists $w \in \mathcal{S}$ such that $v \succeq w$.

In addition to its role as a relational operator, we use $\preceq$ as the unary operator given by $\preceq v = \{u : \ u \preceq v\}$.

Let $\mathcal{S} \subseteq \{0,1\}^*$. Let $f : \mathcal{S} \to \mathbb{R}^+$ and let $\mathcal{S}' \subseteq \mathcal{S}$. We define $f(\mathcal{S}') = \sum_{v \in \mathcal{S}'} f(v)$.

4.2 Frontiers

We identify the vertices of a tree with binary strings in the usual manner, namely, the root corresponds to the empty string ε, and inductively the left child and right child of a vertex v correspond to the strings $v \circ 0$ and $v \circ 1$, respectively. A *frontier* or a *cut* in a (finite) binary tree is any subset of the tree's vertices that intersects every root-to-leaf path in exactly one vertex. For example, the leaves of the tree form a frontier. More generally, by truncating a given tree arbitrarily and considering the resulting set of leaves, one obtains a frontier in the original tree. Given our identification of tree vertices with binary strings, we view frontiers as subsets of $\{0,1\}^n$.

Let v be a vertex in a (finite) binary tree. A frontier with respect to v is any subset of descendants of v that intersects every v-to-leaf path in exactly one vertex.

4.3 The Communication Tree

Consider a randomized protocol with input space $\mathcal{X} \times \mathcal{Y}$. Assume for simplicity that it is a private-coin protocol, meaning that Alice and Bob do not have access to a shared source of random bits. They communicate by sending one bit at a time. For any given history of previously transmitted bits (a transcript), the protocol specifies which of the participants must send the next bit, which in turn is a function of the sender's private random string, the sender's input, and the history of previously transmitted bits.

For the rest of this section, let π be an arbitrary but fixed private-coin protocol and μ be a distribution over $\mathcal{X} \times \mathcal{Y}$. We denote by $I = \mathsf{IC}_\mu(\pi)$ the external information of π over μ. We denote by C be the worst case communication complexity of π.

We identify the set of possible (partial) transcripts communicated by π with a binary tree called the *communication tree*. We denote by $\mathcal{V}$ the set of vertices of the communication tree, and by $\mathcal{L}$ the leaves of this tree. For a vertex $v \in \mathcal{V}$, we denote by $\mathcal{V}(v)$ the set of vertices in the subtree rooted at v (the set $\mathcal{V}(v)$ includes v itself). The root of the tree is denoted ε.

Distributions Over Vertices. Let (X, Y) be a pair of random variables distributed according to μ, representing the players' inputs. Let Π be a random variable representing the transcript of the protocol π when it is run with the inputs (X, Y). Let $v \in \mathcal{V}$ and let $w \in \mathcal{V}(v)$. We define

$$\begin{aligned} P_v(w) &= \Pr[\Pi \succeq w \mid \Pi \succeq v], \\ P_{v,x}(w) &= \Pr[\Pi \succeq w \mid X = x, \Pi \succeq v], \\ P_{v,y}(w) &= \Pr[\Pi \succeq w \mid X = y, \Pi \succeq v], \\ P_{v,x,y}(w) &= \Pr[\Pi \succeq w \mid X = x, Y = y, \Pi \succeq v]. \end{aligned}$$

Let $v \in \mathcal{V}$, and let $\mathcal{F}$ be a frontier with respect to v. Observe that $\mathcal{F}$ induces v. We define the following probability distributions defined over $\mathcal{F}$ by $P_{\mathcal{F}}(w) = P_v(w)$, $P_{\mathcal{F},x}(w) = P_{v,x}(w)$, $P_{\mathcal{F},y}(w) = P_{v,y}(w)$, $P_{\mathcal{F},x,y}(w) = P_{v,x,y}(w)$. When we omit the vertex v, we mean that v is the root. E.g., $P_{x,y}(w) = P_{\varepsilon,x,y}(w)$.

4.4 Information Frontiers

Let $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We define Alice's information frontiers $\mathcal{F}_{i,x}$ and Bob's information frontiers $\mathcal{F}_{i,y}$ recursively. The frontiers $\mathcal{F}_{0,x}$ and $\mathcal{F}_{0,y}$ contain only the root. For $i \geq 1$, we define

$$\begin{aligned} \mathcal{F}_{i,x} = \Big\lfloor \Big\{ w \in \mathcal{V} :\ & \exists v \in \mathcal{F}_{i-1,x} \ s.t.\ v \preceq w \\ & \textit{and} \ \log\left(\tfrac{P_{v,x}(w)}{P_v(w)}\right) \notin (-1,1) \Big\}, \mathcal{L} \Big\rfloor, \end{aligned} \tag{1}$$

$$\begin{aligned} \mathcal{F}_{i,y} = \Big\lfloor \Big\{ w \in \mathcal{V} :\ & \exists v \in \mathcal{F}_{i-1,y} \ s.t.\ v \preceq w \\ & \textit{and} \ \log\left(\tfrac{P_{v,y}(w)}{P_v(w)}\right) \notin (-1,1) \Big\}, \mathcal{L} \Big\rfloor. \end{aligned} \tag{2}$$

Note that $\mathcal{F}_{i,x}$ and $\mathcal{F}_{i,y}$ are indeed frontiers of the communication tree. As was done by previous papers, we assume that the protocol π is smooth (see [3, 8]). It therefore holds that

$$\begin{aligned} \mathcal{F}_{i,x} = \Big\lfloor \Big\{ w \in \mathcal{V} :\ & \exists v \in \mathcal{F}_{i-1,x} \ s.t.\ v \preceq w \\ & \textit{and} \ \tfrac{P_{v,x}(w)}{P_v(w)} \in [1/4, 1/2] \cup [2,4] \Big\}, \mathcal{L} \Big\rfloor, \end{aligned} \tag{3}$$

$$\begin{aligned} \mathcal{F}_{i,y} = \Big\lfloor \Big\{ w \in \mathcal{V} :\ & \exists v \in \mathcal{F}_{i-1,y} \ s.t.\ v \preceq w \\ & \textit{and} \ \tfrac{P_{v,y}(w)}{P_v(w)} \in [1/4, 1/2] \cup [2,4] \Big\}, \mathcal{L} \Big\rfloor. \end{aligned} \tag{4}$$

For $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $v \in \mathcal{V}$ we define

$$\begin{aligned} \mathcal{F}_{v,x} &= \mathcal{V}(v) \cap \mathcal{F}_{\min\{i \geq 0:\ v \preceq \mathcal{F}_{i,x}\},x}, \\ \mathcal{F}_{v,y} &= \mathcal{V}(v) \cap \mathcal{F}_{\min\{i \geq 1:\ v \preceq \mathcal{F}_{i,y}\},y}. \end{aligned}$$

Observe that for any $u \in \mathcal{L}$ it is the case that $(\mathcal{F}_{\varepsilon,x} \cap \preceq u) \prec (\mathcal{F}_{\varepsilon,y} \cap \preceq u)$. That is, Alice's first frontier is above Bob's first frontier.

The following claim bounds the amount of information about x revealed between a vertex v and the closest frontier.

Claim 1. *Let $v \in \mathcal{V}$ and $v \preceq w \preceq \mathcal{F}_{v,x}$. Then $\frac{P_{v,x}(w)}{P_v(w)} \in (1/8, 8)$. A similar claim holds when we switch the roles of x by y.*

Proof. If $v \in \mathcal{F}_{j,x}$ for some $j \in \mathbb{N}$, then, $\mathcal{F}_{v,x} = \mathcal{V}(v) \cap \mathcal{F}_{j,x}$. Since $v \preceq w \preceq \mathcal{F}_{v,x}$ it is also the case that $w = v$, thus $\frac{P_{v,x}(w)}{P_v(w)} = 1$, and the assertion follows. Now assume that $v \notin \mathcal{F}_{j,x}$, for every j. Let $i = \max\{j \in \mathbb{N} :\ \mathcal{F}_{j,x} \preceq v\}$. Let $v' = \mathcal{F}_{i,x} \cap (\preceq v)$. Since $v \preceq w \preceq \mathcal{F}_{v,x}$, it means that $w \preceq \mathcal{F}_{i+1,x}$. By Equation (3), $\frac{P_{v',x}(w)}{P_{v'}(w)} \in [1/4, 4]$. If $\frac{P_{v',x}(v)}{P_{v'}(v)} \in (-\infty, 1/2] \cup [2, \infty)$, then, by Equation (2), $v \succeq \mathcal{F}_{i+1,x}$. Since $v \preceq w \preceq \mathcal{F}_{i+1,x}$, we get that $v \in \mathcal{F}_{i+1,x}$. But this is impossible, as we assume that $v \notin \mathcal{F}_{j,x}$ for every $j \in \mathbb{N}$. Conclude that $\frac{P_{v',x}(v)}{P_{v'}(v)} \in (1/2, 2)$. Therefore, $\frac{P_{v,x}(w)}{P_v(w)} = \frac{P_{v',x}(w) \cdot P_{v'}(v)}{P_{v',x}(v) \cdot P_{v'}(w)} \in (1/4 \cdot 1/2, 4 \cdot 2) = (1/8, 8)$, and the assertion follows. □

Let $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $u \in \mathcal{L}$. The number of Alice's frontiers crossed by the path from the root to u is denoted $F_x(u)$, and the number of Bob's frontiers crossed by the path from the root to u is denoted $F_y(u)$. Formally, $F_x(u)$ and $F_y(u)$, are defined by:

$$F_x(u) = \min\{i \ : \ \mathcal{F}_{i,x} \cap (\preceq u) = u\}, \tag{5}$$

$$F_y(u) = \min\{i \ : \ \mathcal{F}_{i,y} \cap (\preceq u) = u\}. \tag{6}$$

The following useful lemma shows that the expected number of frontier crossed by a path to a leaf is roughly bounded by the external information cost.

Lemma 4.1. *It holds that*

$$\mathop{\mathbf{E}}_{U \leftarrow P_{\mathcal{L},X,Y}} [F_X(U) + F_Y(U)] \leq 10I.$$

4.5 The Chopped Protocol π'

Let $N = 100I/\varepsilon$. Given a communication protocol π, we define the *chopped protocol* π'. Informally speaking, until reaching the N^{th} frontier, the parties running π' follow the protocol π. After the N^{th} frontier was reached, the parties ignore their inputs and communicate according to the distribution of an external observer. Formally, let Π' be the transcript of π'. Let $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $v \in \mathcal{V}$, and $b \in \{0,1\}$. Assume that when reaching v, Alice communicates a bit. Define

$$\Pr[\Pi' \succeq v \circ b \mid X = x, \Pi' \succeq v] = \begin{cases} \Pr[\Pi \succeq v \circ b \mid X = x, \Pi \succeq v] & \text{if } v \preceq \mathcal{F}_{N,x} \\ \Pr[\Pi \succeq v \circ b \mid \Pi \succeq v] & \text{if } v \succ \mathcal{F}_{N,x}. \end{cases}$$

The definition for the case that Bob is communicating a bit when reaching v is obtained from the above by switching the roles of X, x and Y, y.

Observe that for every distribution μ, it holds that

$$\mathsf{IC}_\mu(\pi') = O(\mathsf{IC}_\mu(\pi)) = O(I).$$

In this section we will use the superscript π or π' to indicate the underlying protocol assumed. E.g., $\mathcal{F}^{\pi'}_{i,x}$ is Alice's i^{th} frontier when the protocol π' is run.

The following claim shows that in π', the number of frontier crossed by a path to a leaf is at most $N+1$.

Claim 2. *Let $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $u \in \mathcal{L}$. It holds that $F^{\pi'}_x(u), F^{\pi'}_y(u) \leq N+1$.*

The next claim shows that π and π' are close. It follows directly from Markov's inequality and Lemma 4.1.

Claim 3. *It holds that*

$$\mathop{\mathbf{E}}_{X,Y} \left[|\pi(X,Y) - \pi'(X,Y)| \right] \leq 0.1\varepsilon.$$

5 DESCRIPTION OF OUR COMPRESSION SCHEME

Our simulation protocol, τ', is presented in Section 5.4. We first survey the (sub)protocols *Pred*, Mid^A, Mid^B, τ that are used to construct τ'.

We assume to be given a protocol π and denote by π' the chopped version of π (see Section 4.5). The protocol τ' aims to simulate π', and thus, due to Claim 3, also simulate π. The protocols Mid^A, Mid^B, τ, τ' work with the underlying protocol π'. That is, in all that follows, when we use $P_{v,x}, P_{v,y}, P_{v,x,y}, \mathcal{F}_{i,x}, \mathcal{F}_{i,y}, \mathcal{F}_{v,x}, \mathcal{F}_{v,y}, F_x, F_y$ etc. we mean $P^{\pi'}_{v,x}, P^{\pi'}_{v,y}, P^{\pi'}_{v,x,y}, \mathcal{F}^{\pi'}_{i,x}, \mathcal{F}^{\pi'}_{i,y}, \mathcal{F}^{\pi'}_{v,x}, \mathcal{F}^{\pi'}_{v,y}, F^{\pi'}_x, F^{\pi'}_y$ etc.

5.1 The Protocol *Pred*

The protocol $Pred(v^A, v^B)$ is given in Algorithm 1. The protocol *Pred* assumes that Alice has a vertex $v^A \in \mathcal{V}$ and Bob has a vertex $v^B \in \mathcal{V}$, such that v^A and v^B are on some path from the root to a leaf. That is, $v^A \preceq v^B$ or $v^A \succeq v^B$. It is assumed that $v^A \neq v^B$.

The protocol *Pred* uses the protocol fdiff_D (first difference) suggested in the solution to Exercise 3.18 in [16]. The protocol fdiff_D communicates at most $O(\log(D))$ bits and computes the function $f : \{0,1\}^D \times \{0,1\}^D \to [D]$ given by $f(a,b) = \min\{i \in [D] : a_i \neq b_i\}$, with constant error probability. For any $v \in \mathcal{V}$ denote by $\text{level}(v) \in \{0,1\}^{\lceil \log(C) \rceil + 1}$, the binary representation of the level in the tree that contains v (the root is in level 0).

An execution of $Pred(v^A, v^B)$ returns a pair $(p, W) \in \{0,1\} \times \mathcal{V}$, where:

- If $v^A \preceq v^B$ then $p = 1$, otherwise $p = 0$.
- W is between v^A and v^B. Formally, if $v^A \preceq v^B$ then $v^A \prec W \preceq v^B$. Otherwise, $v^B \prec W \preceq v^A$.

The protocol *Pred* communicates at most $O(\log\log(C) \cdot \log(I)/\varepsilon^2)$ bits, and errs with probability at most $\varepsilon^2 \cdot I^{-10}$. The protocol *Pred* is randomized. However, whenever *Pred* returns a correct vertex W satisfying the above condition, it returns the same vertex W (when it is run with the same inputs).

5.2 The Protocol Mid^A

The protocol $Mid^A(x, y, v, u)$ is given in Algorithm 2. It assumes that Alice has an input $x \in \mathcal{X}$, Bob has an input $y \in \mathcal{Y}$, and that both players agree on a vertex $v \in \mathcal{V}$ and a leaf $u \in \mathcal{L}(v)$. The protocol returns a vertex w on the path from v to u. The vertex w separates Alice's and Bob's frontiers, in the sense that no frontier of Bob's intersects the path from v to w, and the first of Bob's frontiers to intersect the path from w to u, intersects it above Alice's next frontier. It holds that

$$w \preceq (\mathcal{F}_{v,y} \cap \preceq u) \preceq (\mathcal{F}_{w,x} \cap \preceq u). \tag{7}$$

By Claim 2, Mid^A runs *Pred* at most $O(N) = O(I)$ times. This holds as each time *Pred* is called (with the exception of the first call), W crosses at least one Alice's frontiers $\mathcal{F}_{i,X}$. Therefore the following holds:

Claim 4. *The protocol Mid^A communicates at most $T = O(I \log(I) \cdot \log\log(C)/\varepsilon^2)$ bits, and errs with probability at most $\varepsilon \cdot I^{-5}$.*

Remark 1. *The protocol Mid^A is randomized. However, whenever Mid^A returns a correct vertex W satisfying the above condition, it returns the same vertex W.*

When we want to emphasis that the random string used by a specific execution is of Mid^A is R, we write Mid^A_R.

The protocol Mid^B is obtained from Mid^A by switching the roles of X and Y. Claims analogous to the above hold for Mid^B.

5.3 The Protocol τ

The protocol τ is given in Algorithm 3. The protocol τ executes the correlated sampling protocol, *CorrelatedSampling*, suggested in [7]. The protocol *CorrelatedSampling* assumes that Alice knows a distribution P over $\mathcal{L}$, and Bob knows (a possibly different) distribution Q over $\mathcal{L}$. After executing $CorrelatedSampling(P, Q)$, Alice gets a leaf $u \in \mathcal{L}$ sampled according to P. Bob gets a leaf $u' \in \mathcal{L}$ such that $u' = u$, except with probability $\varepsilon^2 \cdot I^{-10}$.

We next show that τ can be implemented by Alice and Bob. In particular, we claim the correlated sampling step (Line 6) and the rejection step (Line 11) can be carried out.

The rejection step assumes that whenever Line 11 is called, it holds that $\frac{P_{V,Y}(Z)}{8P_V(Z)} \leq 1$. This is indeed the case by Claim 1, as $Z \in \mathcal{F}_{V,Y}$.

To show that correlated sampling step can be carried out, we first observe that Bob knows the distribution $P_{\mathcal{L}(V)}$. Alice knows the distribution P^A, as, as shown by the next lemma, she can calculate the set $\mathcal{U}$ (although she does not know Y).

For a possible (partial) transcript M of Mid^A or Mid^B, we denote by $\|M\|$ the number of iterations of the loop in Line 2 of Mid in the execution that resulted in the transcript M.

Claim 5. *Whenever Line 12 of the protocol π is called, Alice can compute the set $\mathcal{U}$ by herself.*

Proof. Assume that we have reached a specific iteration of the loop in Line 2 of τ. Let $V, V', U, W, M, \mathcal{U}, R, leader$ be the values of the variables $V, V', W, U, M, \mathcal{U}, R, leader$ after V was updated for the last time before this iteration. Assume without loss of generality that $leader = Alice$.

Fix $u \in \mathcal{L}(V)$. Consider the first $\|M\|$ iterations of the loop in Line 2 of Mid when running $Mid^B_R(X, Y, V', u)$ (if indeed Mid^B_R runs for at least this many iterations). By Line 12 of τ, $u \in \mathcal{U}$ if and only if the transcript of these $\|M\|$ iterations is M. We will show that Alice can simulate Bob's part in these iterations, and thus can compute $\mathcal{U}$. Let $j \in [\|M\|]$. Let W'_j be the value of the variable W at the beginning of iteration j. Let $(M')^A_j$ and $(M')^B_j$ be Alice's and Bob's messages (respectively) in iteration j.

Now consider the first $\|M\|$ iterations of the loop in Line 2 of Mid when running $Mid^B_R(X, Y, V', U)$. Let W_j be the value of the variable W at the beginning of iteration j. Let M^A_j and M^B_j be Alice's and Bob's messages (respectively) for iteration j. By Line 9 of τ, $M = (M^A_j, M^B_j)_{j \in [\|M\|]}$. Define $(M')^A_0 = M^A_0 = (M')^B_0 = M^B_0 = \varepsilon$. We next prove the following claim

Claim 6. *Let $j' \in \{0, \ldots, \|M\|\}$. If $(M')^A_j = M^A_j$ for every $j \leq j'$, then also $(M')^B_j = M^B_j$ for every $j \leq j'$.*

Proof of Claim 6. We prove by induction. For $j' = 0$, the claim holds trivially. To prove the claim for $j' \geq 1$, we observe that by the induction hypothesis, since $(M')^A_j = M^A_j$ for $j \leq j' - 1$, then also $(M')^B_j = M^B_j$ for $j \leq j' - 1$, which implies that $W'_j = W_j$. Since $j' \leq \|M\|$, it holds that j' is not the last iteration of the loop in Line 2 of Mid^B when running $Mid^B_R(X, Y, V', U)$ (by Line 6 of Mid^B, the last iteration is not recorded to M). Therefore, $(\mathcal{F}_{W_{j'}, Y} \cap \preceq U) \prec W_{j'+1} \preceq (\mathcal{F}_{V,X} \cap \preceq U)$. Since $W_{j'+1} \preceq W = V$, we get $(\mathcal{F}_{W_{j'}, Y} \cap \preceq U) \preceq V$. As $u, U \in \mathcal{L}(V)$, it holds that $(\mathcal{F}_{W_{j'}, Y} \cap \preceq u) = (\mathcal{F}_{W_{j'}, Y} \cap \preceq V) = (\mathcal{F}_{W'_{j'}, Y} \cap \preceq U)$, thus Bob's input is the same when running $Pred$. Since we assume that $(M')^A_j = M^A_j$ for every $j \leq j'$, it also holds that $(M')^B_{j'} = M^B_{j'}$, as Bob messages are a function of his input and the messages communicated by Alice. □

The assertion follows from the claim for the following reason: Alice knows M^B_j for every $j \in [\|M\|]$, as they are induced by M. To decide whether $u \in \mathcal{U}$, Alice does the following: Alice compares $(M')^A_1$ to M^A_1. If they are not the same, then clearly $u \notin \mathcal{U}$ and she halts. If $(M')^A_1 = M^A_1$, then by Claim 6 it is also the case that $(M')^B_1 = M^B_1$. Thus, Alice knows that the transcript of the first iteration of the loop in Line 2 when running $Mid^B_R(X, Y, V', u)$ is consistent with M. Alice can repeat the process (compare $(M')^A_2$ and M^A_2 etc.) to check that transcript of the first $\|M\|$ iterations of the loop in Line 2 of Mid when running $Mid^B_R(X, Y, V', u)$ is indeed M. □

5.4 The Protocol τ'

Our final simulation protocol τ' is obtained from τ by making the following changes:

(1) The loop in Line 2 of the protocol τ is never executed more than $N' = 10^{10}N/\varepsilon = O(I)$ times. If the loop does not halt after this many iterations, τ' returns failure.
(2) The *total* number of times that the loop in Line 7 of the protocol τ is executed during an execution of τ' is at most $N'' = 10^{10}N' \cdot N/\varepsilon = O(I^2)$. If τ attempts to execute this loop more than N'' times, τ' returns failure.

The proof of Theorem 1.2, showing that τ' simulates π with low communication cost, is given by Lemma 8.1 (accuracy lemma proved in Section 8) and by Lemma 13 (communication cost lemma proved in Section 9).

6 DEFINITIONS FOR PROTOCOL ANALYSIS

Fixing. For the rest of the text, fix the inputs x, y.

Notation. Let $v \in \mathcal{V}$. Let $F(v)$ be the set containing all intersections between the path from the root to v and one of the information frontiers. Formally,

$$F(v) = \left(\cup_{i \in \mathbb{N}}(\mathcal{F}_{i,x} \cap \preceq v)\right) \cup \left(\cup_{i \in \mathbb{N}}(\mathcal{F}_{i,y} \cap \preceq v)\right).$$

Let $t(v) = |F(v)|$ be the number of information frontiers intersecting the path from the root to v. Let $t \in [t(v)]$. Let $f_t(v) \in F(v)$ be the intersection between the path from the root to v and the t^{th} information frontier. Formally, $f_t(v)$ is a vertex in $F(v)$ for which there are exactly t distinct vertices v' in $F(v)$ satisfying $v' \preceq v$.

Let $v \in \mathcal{V}$. Note that $f_{t(v)}(v)$ is the last intersection between the path from the root to v and the information frontiers. Define $leader(v) = A$ if $f_{t(v)}(v) = (\mathcal{F}_{i,y} \cap \preceq v)$ for some $i \in [t(v)]$. Define $leader(v) = B$ if $f_{t(v)}(v) = (\mathcal{F}_{j,x} \cap \preceq v)$ for some $j \in [t(v)]$.

For all that follows, $T \in \mathbb{N}$ is the upper bound on the length of the transcript of the protocols Mid^A and Mid^B obtained in Claim 4. Observe that T is also an upper bound on the binary representation size of the M variable returned by Mid^A and Mid^B. We will assume that $M \in \{0,1\}^T$. Let $T' \in \mathbb{N}$ be an upper bound on the number of random bits used by Mid^A and Mid^B. Let $\mathcal{R}$ be the set of all sequences $\bar{r} = r_1, r_2, \ldots,$ such that $r_t \in \{0,1\}^{T'}$.

Useful Sets. Let $\mathcal{S} \subseteq \{0,1\}^{T'} \times \mathcal{V}^2 \times \{0,1\}^T$ be the set of all (r, v, w, m) such that

$$\exists u \in \mathcal{V}(v): \; Mid_r^{leader(v)}(x, y, v, u) = (w, m).$$

For $(r, v, w, m) \in \mathcal{S}$, we define the sets $\mathcal{U}(r, v, w, m)$, $\mathcal{U}^*(r, v, w, m)$, $\mathcal{L}^*(r, v, w, m)$, $\mathcal{V}^*(r, v, w, m)$ as follows: If $leader(v) = B$,

$$\mathcal{U}(r, v, w, m) = \{u \in \mathcal{L}(w) : \ (Mid_r^B(x, y, v, u))_2 \geq m\},$$

$$\mathcal{U}^*(r, v, w, m) = \{u \in \mathcal{L}(w) : \ (Mid_r^B(x, y, v, u))_2 = m\},$$

$$\mathcal{L}^*(r, v, w, m) = \left\{u \in \mathcal{U}(r, v, w, m) \ : \ (\mathcal{F}_{v,x} \cap \leq u) \leq (\mathcal{F}_{w,y} \cap \leq u)\right\},$$

$$\mathcal{V}^*(r, v, w, m) = \{v^* = (\mathcal{F}_{v,x} \cap \leq u) : \ u \in \mathcal{U}(r, v, w, m), v^* \leq \mathcal{F}_{w,y}\}.$$

If $leader(v) = A$, the definitions of $\mathcal{U}(r, v, w, m)$, $\mathcal{U}^*(r, v, w, m)$, $\mathcal{L}^*(r, v, w, m)$, $\mathcal{V}^*(r, v, w, m)$ are obtained from the above definitions by switch the roles of A and B and of x and y.

Random Variables. In all that follows, we denote by V_t, V'_t, M_t, P_t^A, $\mathcal{U}_t, R_t, W_t$ the last values attained by the variables $V, V', M, P^A, \mathcal{U}$, R, W (respectively) in the iteration of the loop in Line 2 of τ in which V is updated for the t^{th} time (the initialization of V in Line 1 of τ is the 0^{th} update of V).

Unique Decomposition. Let $u \in \mathcal{L}$ and let $\bar{r} \in \mathcal{R}$. We define the *unique decomposition* $\{(v_t, m_t)\}_t$ of u with respect to $x, y, \bar{r}$ in a recursive manner: Let $(v_0, m_0) = (\varepsilon, \varepsilon)$. For $t \geq 1$, define

$$(v_{t+1}, m_{t+1}) = Mid_{\bar{r}_t}^{leader(v_t)}(x, y, v_t, u).$$

Let $u, u' \in \mathcal{L}$, and let $v \in \mathcal{V}$ be the lowest common ancestor of u and u'. Let $\{(v_t, m_t)\}_t$ and $\{(v'_t, m'_t)\}_t$ be the unique decomposition of u and u' (respectively) with respect to $x, y, \bar{r}$. Let i be the maximal such that $v_i \leq v$. Then, $(v_t, m_t) = (v'_t, m'_t)$ for every $t \leq i$.

Let $v \in \mathcal{V}$. Let $u \in \mathcal{L}$ such that $u \geq v$. Let $\{(v_t, m_t)\}_t$ be the unique decomposition of u with respect to $x, y, \bar{r}$. Let i be the maximal such that $v_i \leq v$. We define the unique decomposition of v with respect to $x, y, \bar{r}$ to be $\{(v_t, m_t)\}_{t \leq i}$. Observe that the unique decomposition of v is well defined.

7 ACCURACY OF τ

We next show that τ simulates the chopped protocol π'. For the rest of the section we will be assuming that the implementations of the *Pred* and *CorrelatedSampling* protocols are error free.

Assumption 1. *The Pred and CorrelatedSampling protocols never err or fail.*

Consider the randomness used by an execution of τ. We denote by R the randomness used for the executions of Mid^A and Mid^B by τ. We denote by R' the rest of the randomness used by τ. Let $\bar{r} \in \mathcal{R}$ and recall that x, y are fixed. Let $\tau_{\bar{r}}(x, y)$ be the distribution of τ's outputs when it is run on inputs x, y with randomness $R = \bar{r}$. Note that we did not fix R'. Let $\pi'(x, y)$ be the distribution over the leaves of the communication tree of π' obtained by running π' with inputs x, y.

Lemma 7.1. *Under Assumption 1, for every $\bar{r} \in \mathcal{R}$, the distribution $\tau_{\bar{r}}(x, y)$ is identical to the distribution $\pi'(x, y)$.*

The rest of the section is devoted to proving Lemma 7.1.

Lemma 7.2. *Let $v \in \mathcal{V}$, $u' \in \mathcal{L}(v)$, and $r \in \{0, 1\}^{T'}$. Assume that $leader(v) = B$. Assume that $Mid_r^B(x, y, v, u') = (w', m')$ for some $w' \in \mathcal{V}(v)$ and $m' \in \{0, 1\}^T$. Let $u \in \mathcal{L}(w')$ and assume that $Mid_r^B(x, y, v, u) = (w, m)$, for some $w \in \mathcal{V}(v)$ and $m \geq m' \in \{0, 1\}^T$.*

Let $i \in [\|m'\| + 1]$. Define W'_i to be the value of the variable W at the beginning of the i^{th} iteration of Line 2 of Mid^B when running $Mid_r^B(x, y, v, u')$. Define W_i to be the value of the variable W at the beginning of the i^{th} iteration of Line 2 of Mid^B when running $Mid_r^B(x, y, v, u)$.

Then, $W'_i = W_i$ for every $i \in [\|m'\| + 1]$. In particular, if $m' = m$ then $w' = w$.

Proof. The execution of $Mid_r^B(x, y, v, u')$ results in (w', m'), which means that the execution of $Pred_r(\mathcal{F}_{W'_{\|m'\|}, y} \cap \leq u', \mathcal{F}_{v,x} \cap \leq u')$ by $Mid_r^B(x, y, v, u')$ returned $(p = 1, w')$. This implies that $(\mathcal{F}_{W'_{\|m'\|}, y} \cap \leq u') \leq w'$. Since $W_1 = W'_1 = v$, since $v = W'_1 \leq W'_2 \leq \ldots \leq W'_{\|m'\|}$, and since $u \in \mathcal{L}(w')$, it holds that $(\mathcal{F}_{W'_1, y} \cap \leq u') = (\mathcal{F}_{W_1, y} \cap \leq u)$. Therefore, Bob's input for the first execution of *Pred* by both $Mid_r^B(x, y, v, u')$ and $Mid_r^B(x, y, v, u)$ are the same. Since the output of *Pred* is known to both parties, it is determined by Bob's input and the transcript. Since $m' \leq m$ and since Bob's inputs are the same, the output is also the same and $W_2 = W'_2$. We continue this process and show that $W_i = W'_i$ for every $i \in [\|m'\| + 1]$. □

Lemma 7.3. *Let $(r, v, w, m) \in \mathcal{S}$. It holds that*

$$\mathcal{U}^*(r, v, w, m) = \mathcal{L}^*(r, v, w, m).$$

Proof. Let $\mathcal{U}^* = \mathcal{U}^*(r, v, w, m)$, $\mathcal{U} = \mathcal{U}(r, v, w, m)$, and $\mathcal{L}^* = \mathcal{L}^*(r, v, w, m)$. Assume without loss of generality that $leader(v) = B$. Recall that $(r, v, w, m) \in \mathcal{S}$ implies

$$\exists u' \in \mathcal{V}(v) : \ Mid_r^B(x, y, v, u') = (w, m). \tag{8}$$

The direction $\mathcal{L}^* \subseteq \mathcal{U}^*$. Let $u \in \mathcal{U}^*$. Thus, $u \in \mathcal{L}(w)$ and $(Mid_r^B(x, y, v, u))_2 = m$. By Lemma 7.2 and Equation (8), it is also the case that $(Mid_r^B(x, y, v, u))_1 = w$. This means that the loop in Line 2 of Mid^B ended with $p = 0$, where $p = (Pred_r(\mathcal{F}_{w,y} \cap \leq u, \mathcal{F}_{v,x} \cap \leq u))_1$. That is, it ended because the last execution of *Pred* indicated that $(\mathcal{F}_{v,x} \cap \leq u) \leq (\mathcal{F}_{w,y} \cap \leq u)$. Since it is also the case that $u \in \mathcal{U}^* \subseteq \mathcal{U}$, conclude that $u \in \mathcal{L}^*$.
The direction $\mathcal{L}^* \subseteq \mathcal{U}^*$. Let $u \in \mathcal{L}^*$. As $u \in \mathcal{L}^* \subseteq \mathcal{U}$, it holds that $u \in \mathcal{L}(w)$ and that $(Mid_r^B(x, y, v, u))_2 \geq m$. By Lemma 7.2 and Equation (8), the value of the variable W at the beginning of the $(\|m\| + 1)^{st}$ iteration of Line 2 of Mid^B when running $Mid_r^B(x, y, v, u')$ and $Mid_r^B(x, y, v, u)$ is the same. The value of the variable W at the beginning of the $(\|m\| + 1)^{st}$ iteration of Line 2 of Mid^B when running $Mid_r^B(x, y, v, u')$ is w, thus value of the variable W at the beginning of the $(\|m\| + 1)^{st}$ iteration of Line 2 of Mid^B when running $Mid_r^B(x, y, v, u)$ is also w. Since $u \in \mathcal{L}^*$ it holds that $(\mathcal{F}_{v,x} \cap \leq u) \leq (\mathcal{F}_{w,y} \cap \leq u)$. Therefore, $(Pred_r(\mathcal{F}_{w,y} \cap \leq u, \mathcal{F}_{v,x} \cap \leq u))_1 = 0$. This implies that iteration $\|m\| + 1$ is the last iteration of the execution $Mid_r^B(x, y, v, u)$, and that $u \in \mathcal{U}^*$. □

Claim 7. *Let $(r, v, w, m) \in \mathcal{S}$. It holds that*

$$\mathcal{L}(\mathcal{V}^*(r, v, w, m)) = \mathcal{U}^*(r, v, w, m).$$

Proof. Denote $\mathcal{U}^* = \mathcal{U}^*(r, v, w, m)$, $\mathcal{U} = \mathcal{U}(r, v, w, m)$ and $\mathcal{V}^* = \mathcal{V}^*(r, v, w, m)$. Assume without loss of generality that $leader(v) = B$. By Lemma 7.3, it holds that

$$\begin{aligned}\mathcal{U}^* &= \mathcal{L}^*(r, v, w, m) && (9)\\ &= \left\{u \in \mathcal{U}(r, v, w, m) \ : \ (\mathcal{F}_{v,x}\cap \le u) \le (\mathcal{F}_{w,y}\cap \le u)\right\}. && (10)\end{aligned}$$

The direction $\mathcal{U}^* \subseteq \mathcal{L}(\mathcal{V}^*)$. Let $u \in \mathcal{U}^*$ and denote $v^* = \mathcal{F}_{v,x}\cap \le u$. Then, since $u \in \mathcal{U}^*$ it holds that $u \in \mathcal{U}$ and $v^* = (\mathcal{F}_{v,x}\cap \le u) \le (\mathcal{F}_{w,y}\cap \le u)$. Conclude that $v^* \in \mathcal{V}^*$, thus $u \in \mathcal{L}(\mathcal{V}^*)$.

The direction $\mathcal{L}(\mathcal{V}^*) \subseteq \mathcal{U}^*$. Let $v^* \in \mathcal{V}^*$ and let $u \in \mathcal{L}(v^*)$. We will show that $u \in \mathcal{U}^*$. Since $v^* \in \mathcal{V}^*$, there exists $u' \in \mathcal{U}$ such that $v^* = (\mathcal{F}_{v,x}\cap \le u') \le (\mathcal{F}_{w,y})$. Since $u \in \mathcal{L}(v^*)$, it holds that $(\mathcal{F}_{v,x}\cup \le u) = v^* = (\mathcal{F}_{v,x}\cup \le u')$. Since $v^* \le \mathcal{F}_{w,y}$, we get that $(\mathcal{F}_{v,x}\cup \le u) \le (\mathcal{F}_{w,y}\cup \le u)$. It remains to show that $u \in \mathcal{U}$.

Since $(\mathcal{F}_{v,x}\cup \le u) = (\mathcal{F}_{v,x}\cup \le u')$, all but the last execution of *Pred* by $Mid^B_r(x, y, v, u')$ and $Mid^B_r(x, y, v, u)$ are done with the same parameters, thus the two executions give the same result. Since $u' \in \mathcal{U}$, it holds that $(Mid^B_r(x, y, v, u))_2 = (Mid^B_r(x, y, v, u'))_2 \ge m$, thus $u \in \mathcal{U}$. Conclude that $u \in \mathcal{U}^*$. □

Claim 8. *Let $(r, v, w, m) \in \mathcal{S}$. Let $f_{t(w)}(w) \le w' \le w$. Assume $leader(v) = B$. It holds that*

$$P_{w',x,y}(\mathcal{U}^*(r, v, w, m)) \le 8P_{w',x}(\mathcal{U}^*(r, v, w, m)).$$

Proof. Denote $\mathcal{U}^* = \mathcal{U}^*(r, v, w, m)$, $\mathcal{U} = \mathcal{U}(r, v, w, m)$ and $\mathcal{V}^* = \mathcal{V}^*(r, v, w, m)$. By Claim 7, $\mathcal{L}(\mathcal{V}^*) = \mathcal{U}^*$. Let $v^* \in \mathcal{V}^*$. There exists $u' \in \mathcal{U} \subseteq \mathcal{L}(w)$ such that $v^* = (\mathcal{F}_{v,x}\cap \le u')$. By Equation (7), $w \le (\mathcal{F}_{v,x}\cap \le u')$. Hence, $w \le v^*$. By the definition of $\mathcal{V}^*$ it holds that $v^* \le \mathcal{F}_{w,y}$. Recall the assumption that $f_{t(w)}(w) \le w' \le w$, and get that $f_t(w) \le w' \le w \le v^* \le \mathcal{F}_{w,y}$. Since there are no information frontiers between $f_{t(w)}(w)$ and w, there are no frontiers between w' and w. This implies that $w' \le v^* \le \mathcal{F}_{w',y}$, and by Claim 1 and the fact that $\mathcal{V}^*$ is contained in a frontier, $\frac{P_{w',y}(\mathcal{V}^*)}{P_{w'}(\mathcal{V}^*)} < 8$. Therefore,

$$P_{w',x,y}(\mathcal{V}^*) = \frac{P_{w',x}(\mathcal{V}^*) \cdot P_{w',y}(\mathcal{V}^*)}{P_{w'}(\mathcal{V}^*)} \le 8P_{w',x}(\mathcal{V}^*).$$

By Claim 7, $P_{w',x,y}(\mathcal{U}^*) \le 8P_{w',x}(\mathcal{U}^*)$. □

Lemma 7.4. *Fix t. Let $\mathcal{U}^* = \mathcal{U}^*(R_t, V_{t-1}, V_t, M_t)$. Assume that $leader(V_t) = A$. Then, for $u \in \mathcal{U}^*$,*

$$\Pr_{R,R'}[U_{t+1} = u] = \frac{P_{V_t,x}(u)}{P_{V_t,x}(\mathcal{U}^*)}.$$

Thus, it is also the case that for $u \in \mathcal{L} \setminus \mathcal{U}^*$, $\Pr[U_{t+1} = u] = 0$.

Proof. Let $\mathcal{U} = \mathcal{U}(R_t, V_{t-1}, V_t, M_t)$, $\mathcal{L}^* = \mathcal{L}^*(R_t, V_{t-1}, V_t, M_t)$. Let K be a random variable that counts the number of iterations of loop in Line 7 of τ, during the $(t + 1)^{st}$ execution of the loop in Line 2 of τ. Let $u \in \mathcal{L}^*$. Since $\mathcal{L}^* \subseteq \mathcal{U} = \mathcal{U}_t$, it holds that $u \in \mathcal{U}_t$. It also holds that $\Pr[(U_{t+1} = u) \land (K = 1)] = P^A_{t+1}(u)$, as the correlated sampling step in Line 6 of τ returns u with probability $P^A_{t+1}(u)$ (as $u \in \mathcal{U}_t$), and the condition of the loop in Line 7 of τ is always satisfied. For $u \in \mathcal{L} \setminus \mathcal{L}^*$, it holds that $Pr[(U_t = u) \land (K = 1)] = 0$, as either $u \notin \mathcal{U} = \mathcal{U}_t$, thus $P^A_{t+1}(u) = 0$, or the condition of the loop is not satisfied.

Observe that U_{t+1} is independent of K, since for any $k, k' \in \mathbb{N}$, it holds that $U_{t+1}|(K = k)$ has the same distribution as $U_{t+1}|(K = k')$. Thus, for $u \in \mathcal{L}^*$,

$$\begin{aligned}\Pr[U_{t+1} = u] = \Pr[U_{t+1} = u|K = 1] &= \frac{\Pr[(U_{t+1} = u) \land (K = 1)]}{\Pr[K = 1]}\\ &= \frac{P^A_{t+1}(u)}{\sum_{u\in\mathcal{L}^*} P^A_{t+1}(u)} = \frac{\frac{P_{V_t,x}(u)}{P_{V_t,x}(\mathcal{U}_t)}}{\frac{\sum_{u\in\mathcal{L}^*} P_{V_t,x}(u)}{P_{V_t,x}(\mathcal{U}_t)}} = \frac{P_{V_t,x}(u)}{P_{V_t,x}(\mathcal{L}^*)}\\ &= \frac{P_{V_t,x}(u)}{P_{V_t,x}(\mathcal{U}^*)},\end{aligned}$$

where the last equality is because $(R_t, V_{t-1}, V_t, M_t) \in \mathcal{S}$, thus by Lemma 7.3, $\mathcal{L}^* = \mathcal{U}^*$. □

Lemma 7.5. *Let $v \in \mathcal{V}$ and let $\bar{r} \in \mathcal{R}$. Let $\{(v_t, m_t)\}_t$ be the unique decomposition of v with respect to $x, y, \bar{r}$. Then, for every t it holds that*

$$((V_t, M_t) \mid V = v, X = x, Y = y, R = \bar{r}) = (v_t, m_t).$$

Proof. The proof is by induction on t: For $t = 0$, it holds that $M_0 = m_0 = \varepsilon$. We assume that $((V_{t-1}, M_{t-1}) \mid V = v, X = x, Y = y, R = \bar{r}) = (v_{t-1}, m_{t-1})$, and show that $((V_t, M_t) \mid V = v, X = x, Y = y, R = \bar{r}) = (v_t, m_t)$. Assume $V = v$. Assume without loss of generality that $leader(V_{t-1}) = B$. By Lemma 7.4, $U_{t+1} \in \mathcal{U}^*(R_t, V_{t-1}, V_t, M_t)$. Therefore, $(Mid^B_{R_t}(x, y, V_{t-1}, U_{t+1}))_2 = M_t$.

The paths from the root to v and from the root to U_{t+1} agree until V_{t+1}, as $V_{t+1} \le V = v$ and $U_{t+1} \ge V_{t+1}$. By the induction hypothesis, $V_{t-1} = v_{t-1}$. Since $\mathcal{F}_{v_{t-1},x} = \mathcal{F}_{V_{t-1},x} \le V_{t+1}$, we get that $(\mathcal{F}_{V_{t-1},x}\cap \le U_{t+1}) = (\mathcal{F}_{v_{t-1},x}\cap \le v)$. This implies that all but the last execution of *Pred* by $Mid^B_{\bar{r}_t}(x, y, v_{t-1}, v)$ and $Mid^B_{R_t}(x, y, V_{t-1}, U_{t+1})$ are done with the same parameters, thus the two executions give the same result. Thus,

$$(v_t, m_t) = Mid^B_{\bar{r}_t}(x, y, v_{t-1}, v) = Mid^B_{R_t}(x, y, V_{t-1}, U_{t+1}) = (V_t, M_t),$$

and the assertion follows. □

Lemma 7.6. *Let $v \in \mathcal{V}$ and let $\bar{r} \in \mathcal{R}$. Let $\{(v_t, m_t)\}_t$ be the unique decomposition of v with respect to $x, y, \bar{r}$. We define*

$$\mathcal{U}^*(\bar{r}_0, v_{-1}, v_0, m_0) = \mathcal{L}.$$

Then, for every t,

$$\begin{aligned}&\Pr_{R,R'}[(V_{t+1}, M_{t+1}) = (v_{t+1}, m_{t+1}) \mid\\ &\qquad (V_t, M_t) = (v_t, m_t), \ldots, (V_1, M_1) = (v_1, m_1), X = x, Y = y, R = \bar{r}]\\ &\qquad = \frac{P_{v_t,x,y}(\mathcal{U}^*(\bar{r}_{t+1}, v_t, v_{t+1}, m_{t+1}))}{P_{v_t,x,y}(\mathcal{U}^*(\bar{r}_t, v_{t-1}, v_t, m_t))}.\end{aligned}$$

Proof of Lemma 7.1. Let $v \in \mathcal{V}$ and let $\bar{r} \in \mathcal{R}$. Let $\{(v_t, m_t)\}_t$ be the unique decomposition of v with respect to $x, y, \bar{r}$. Let $\ell \in \mathbb{N}$

be such that $v_{\ell-1} = v$ (note that also $v_\ell = v$). Let V be the vertex V returned by τ. It holds that

$$\begin{aligned}
&\Pr[V = v \mid X = x, Y = y, R = \bar{r}]\\
&= \Pr[(V_\ell, M_\ell) = (v_\ell, m_\ell), \ldots, (V_1, M_1) = (v_1, m_1) \mid \\
&\qquad X = x, Y = y, R = \bar{r}] \qquad \text{(by Lemma 7.5)}\\
&= \prod_{t \in \{0,1,\ldots,\ell-1\}} \Pr\left[(V_{t+1}, M_{t+1}) = (v_{t+1}, m_{t+1}) \mid \right.\\
&\qquad \left.(V_t, M_t) = (v_t, m_t), \ldots, (V_1, M_1) = (v_1, m_1), X = x, Y = y, R = \bar{r}\right]\\
&= \prod_{t \in \{0,1,\ldots,\ell-1\}} \frac{P_{v_t,x,y}(\mathcal{U}^*(\bar{r}_{t+1}, v_t, v_{t+1}, m_{t+1}))}{P_{v_t,x,y}(\mathcal{U}^*(\bar{r}_t, v_{t-1}, v_{t-1}, m_t))}\\
&\qquad \text{(by Lemma 7.6)}\\
&= \prod_{t \in \{0,1,\ldots,\ell-1\}} \frac{P_{v_t,x,y}(v_{t+1}) \cdot P_{v_{t+1},x,y}(\mathcal{U}^*(\bar{r}_{t+1}, v_t, v_{t+1}, m_{t+1}))}{P_{v_t,x,y}(\mathcal{U}^*(\bar{r}_t, v_{t-1}, v_t, m_t))}\\
&\qquad \text{(as } \mathcal{U}^*(\bar{r}_{t+1}, v_t, v_{t+1}, m_{t+1}) \subseteq \mathcal{L}(v_{t+1}))\\
&= \frac{P_{v_\ell,x,y}(\mathcal{U}^*(\bar{r}_\ell, v_{\ell-1}, v_\ell, m_\ell))}{P_{v_0,x,y}(\mathcal{U}^*(\bar{r}_0, v_{-1}, v_0, m_0)))} \cdot \prod_{t \in \{0,1,\ldots,\ell-1\}} P_{v_t,x,y}(v_{t+1})\\
&= P_{v_0,x,y}(v_{\ell-1})\\
&\qquad \text{(as } \mathcal{U}^*(\bar{r}_\ell, v_{\ell-1}, v_\ell, m_\ell) = \{v_\ell\} \text{ and } \mathcal{U}^*(\bar{r}_0, v_{-1}, v_0, m_0) = \mathcal{L})\\
&= P_{x,y}(v).
\end{aligned}$$

The assertion follows as $P_{x,y}(v)$ is the probability that π' outputs v on inputs x, y. □

8 ACCURACY OF τ'

This section is devoted to showing that τ' simulates the original protocol π. Let (X, Y) be a pair of random variables distributed according to μ. Let $\tau'(x, y)$ be the distribution of the output of τ' when it is run on inputs x, y. Let $\pi(x, y)$ be the distribution over the leaves of the communication tree of π obtained by running π with inputs x, y.

Lemma 8.1. *It holds that*

$$\mathop{\mathbf{E}}_{X,Y}\left[|\tau'(X, Y) - \pi(X, Y)|\right] \leq \varepsilon/2.$$

The rest of the section is devoted to proving Lemma 8.1.

Claim 9. *An execution of τ ends after at most N' iterations of the loop in Line 2, except with probability at most $2^{-N'/10}$.*

Proof. Consider an iteration of the loop in Line 2 of τ. Let V be the value of the variable V at the beginning of this iteration, and let Z be the value of the variable Z after it is updated by this iteration. Assume without loss of generality that $leader(V) = A$. By Lemma 2, V needs to be updated at most $2N + 2$ times in the duration of τ's the execution. In the rejection sampling step of τ (Line 11), Bob accepts with probability $\frac{P_{V,y}(Z)}{8P_V(Z)}$, and in this event V is updated. By Line 10, it holds that $Z \in \mathcal{F}_{V,y}$. By Equation (4), $\frac{P_{V,y}(Z)}{8P_V(Z)} \geq \frac{1}{8} \cdot \frac{1}{4} = \frac{1}{32}$. Since V is updated with probability at least $\frac{1}{32}$ by any iteration of Line 2 of τ, and since the total number of updates is at most $2N + 2$, the assertion follows by the Chernoff bound. □

We next show that τ and τ' give the same output with high probability. Recall that x, y are fixed.

Lemma 8.2. *Under Assumption 1, $|\tau(x, y) - \tau'(x, y)| \leq \varepsilon/5$.*

Proof. Let $(r, v, w, m) \in \mathcal{S}$. Define

$$p_{succ}(r, v, w, m) = \frac{P_{w,x}(\mathcal{U}^*(r, v, w, m))}{P_{w,x}(\mathcal{U}(r, v, w, m))}.$$

Note that p_{succ} is well defined as for every $u \in \mathcal{U}(r, v, w, m)$ it holds that $w \preceq u$.

Fix $\delta \in (0, 1]$. We say that $(r, v, w, m) \in \mathcal{S}$ is *bad* if $p_{succ}(r, v, w, m) < \delta$. Let $\mathcal{B} \subseteq \mathcal{S}$ be the set of all bad tuples (r, v, w, m). Consider an iteration of the loop in Line 2 of τ, such that at the beginning of this iteration, $V' = v$, $V = w$ and $M = m$, and the value of the variable R when V was last updated was r. Then, by Lemma 7.4, $p_{succ}(r, v, w, m)$ is the probability that the loop in Line 7 of τ ends after its first iteration. Note that conditioned on reaching iteration i of the loop in Line 7 of τ, the probability of iteration i being the last iteration is the same for every i. Therefore, if $(r, v, w, m) \in \mathcal{S} \setminus \mathcal{B}$, the expected number of iterations of this loop is at most $1/\delta$.

Let $u \in \mathcal{L}$ and let $\bar{r} \in \mathcal{R}$. Let $\{(v_t, m_t)\}_t$ be the unique decomposition of u with respect to $x, y, \bar{r}$. We say that u is bad with respect $\bar{r}$ to if there exists a t such that (r_t, v_{t-1}, v_t, m_t) is bad. Let $\mathcal{L}^{\mathcal{B},\bar{r}} \subseteq \mathcal{L}$ be the set of all bad leaves with respect to $\bar{r}$.

Let τ'' be the protocol obtained from τ by limiting the number of iteration of the loop in Line 2 to at most N' iterations. Assume that τ'' is run with inputs x, y. Let U be the output of τ''. Let K be a random variable counting the total number of repetitions of the loop in Line 7 of τ'' during the execution of τ''. By the Chernoff bound, for every $\bar{r} \in \mathcal{R}$,

$$\Pr_{R,R'}\left[K > \tfrac{1000N'}{\delta} \mid R = \bar{r}, U \in \mathcal{L} \setminus \mathcal{L}^{\mathcal{B},\bar{r}}\right] \leq 2^{-N'/\delta},$$

where R is the randomness used for the executions of Mid^A and Mid^B by τ'', and R' is the rest of the randomness used by τ''. The reason is that the number of iteration of the loop in Line 2 of τ'' is at most N'. This implies that

$$\Pr_{R,R'}\left[K > \tfrac{1000N'}{\delta} \mid U \in \mathcal{L} \setminus \mathcal{L}^{\mathcal{B},R}\right] \leq 2^{-N'/\delta}.$$

We get that

$$\begin{aligned}
&\Pr_{R,R'}\left[K > \tfrac{1000N'}{\delta}\right]\\
&= \Pr_{R,R'}\left[K > \tfrac{1000N'}{\delta} \mid U \in \mathcal{L}^{\mathcal{B},R}\right] \cdot \Pr_{R,R'}\left[U \in \mathcal{L}^{\mathcal{B},R}\right]\\
&\quad + \Pr_{R,R'}\left[K > \tfrac{1000N'}{\delta} \mid U \in \mathcal{L} \setminus \mathcal{L}^{\mathcal{B},R}\right] \cdot \Pr_{R,R'}\left[U \in \mathcal{L} \setminus \mathcal{L}^{\mathcal{B},R}\right]\\
&\leq \Pr_{R,R'}\left[U \in \mathcal{L}^{\mathcal{B},R}\right] + 2^{-N'/\delta}.
\end{aligned}$$

The rest of the proof is devoted to showing that for every $\bar{r} \in \mathcal{R}$, $P_{x,y}(\mathcal{L}^{\mathcal{B},\bar{r}}) \leq 20\delta N$. Thus, by the accuracy claim, Lemma 7.1, it also holds that for every $\bar{r}' \in \mathcal{R}$, $\Pr_{R'}\left[U \in \mathcal{L}^{\mathcal{B},\bar{r}} | R = \bar{r}'\right] \leq 20\delta N$. In particular, $\Pr_{R'}\left[U \in \mathcal{L}^{\mathcal{B},\bar{r}} | R = \bar{r}\right] \leq 20\delta N$, implying $\Pr_{R,R'}\left[U \in \mathcal{L}^{\mathcal{B},R}\right] \leq 20\delta N$. This suggests that $\Pr\left[K > \frac{1000N'}{\delta}\right] \leq 20\delta N + 2^{-N'/\delta}$. By setting $\delta = \varepsilon/200N$, we get that

$$\Pr\left[K > N''\right] \leq \Pr\left[K > \tfrac{200000N' \cdot N}{\varepsilon}\right] \leq \varepsilon/10 + 2^{-200N \cdot N'/\varepsilon} \leq \varepsilon/9.$$

The assertion then follows as by Claim 9,

$$\begin{aligned}|\tau(x,y)-\tau'(x,y)| &\le |\tau(x,y)-\tau''(x,y)|+|\tau''(x,y)-\tau'(x,y)|\\ &\le 2^{-N'/10}+\varepsilon/9 \le \varepsilon/5.\end{aligned}$$

Let $u \in \mathcal{V}$ and $t \in \{0,1,\ldots,2N+2\}$. Let μ_t be a probability distribution over $\mathcal{L}$ given by

$$\mu_t(u)=\begin{cases} P_{x,y}(f_t(u))\cdot P_{f_t(u),x}(u) & \text{if } f_t(u)=(\mathcal{F}_{i,y}\cap \preceq u) \text{ for some } i\\ P_{x,y}(f_t(u))\cdot P_{f_t(u),y}(u) & \text{if } f_t(u)=(\mathcal{F}_{i,x}\cap \preceq u) \text{ for some } i.\end{cases}$$

Claim 10. *Let $(r,v,w,m)\in\mathcal{B}$ and denote $t=t(w)$. It holds that*

$$P_{x,y}(\mathcal{U}^*(r,v,w,m)) \le 8\delta\mu_t(\mathcal{U}(r,v,w,m)).$$

Proof. Denote $\mathcal{U}^*=\mathcal{U}^*(r,v,w,m)$ and $\mathcal{U}=\mathcal{U}(r,v,w,m)$. Assume without loss of generality that $leader(v)=B$. There exists i such that $v\preceq\mathcal{F}_{i,y}\preceq w$, but there is no j such that $v\preceq\mathcal{F}_{j,x}\preceq w$, therefore $f_t(w)=(\mathcal{F}_{i',y}\cap\preceq w)$ for some i'.

It holds that

$$\begin{aligned}\delta &> p_{succ}(r,v,w,m) && (\text{as } (r,v,w,m)\in\mathcal{B})\\ &=\frac{P_{w,x}(\mathcal{U}^*)}{P_{w,x}(\mathcal{U})}\\ &=\frac{P_{x,y}(f_t(w))\cdot P_{f_t(w),x}(w)\cdot P_{w,x}(\mathcal{U}^*)}{P_{x,y}(f_t(w))\cdot P_{f_t(w),x}(w)\cdot P_{w,x}(\mathcal{U})} && (\text{as } f_t(w)\preceq w)\\ &=\frac{P_{x,y}(f_t(w))\cdot P_{f_t(w),x}(\mathcal{U}^*)}{\mu_t(\mathcal{U})}\\ & && (\text{as } f_t(w)=(\mathcal{F}_{i',y}\cap\preceq w) \text{ for some } i')\\ &\ge\frac{P_{x,y}(f_t(w))\cdot P_{f_t(w),x,y}(\mathcal{U}^*)}{8\mu_t(\mathcal{U})} && (\text{by Claim 8})\\ &=\frac{P_{x,y}(\mathcal{U}^*)}{8\mu_t(\mathcal{U})}.\end{aligned}$$

□

Let $\bar{r}\in\mathcal{R}$. Let $\mathcal{S}_{\bar{r}}$ be the set of all $(r,v,w,m)\in\{0,1\}^{T'}\times\mathcal{V}^2\times\{0,1\}^T$ such that there exists $v\in\mathcal{V}$ with a unique decomposition $\{(v_t,m_t)\}_t$ with respect to $x,y,\bar{r}$, such that $r=\bar{r}_t$, $v=v_{t-1}$, $w=v_t$ and $m=m_t$, for some t. Observe that $\mathcal{S}_{\bar{r}}\subseteq\mathcal{S}$.

Claim 11. *Let $\bar{r}\in\mathcal{R}$. Let $(r_1,v_1,w_1,m_1)\neq(r_2,v_2,w_2,m_2)\in\mathcal{S}_{\bar{r}}$ such that $t(w_1)=t(w_2)$. Then,*

$$\mathcal{U}(r_1,v_1,w_1,m_1)\cap\mathcal{U}(r_2,v_2,w_2,m_2)=\phi.$$

Proof. Since $\mathcal{U}(r_b,v_b,w_b,m_b)\subseteq\mathcal{L}(w_b)$ for $b\in\{0,1\}$, it can only be the case that $\mathcal{U}(r_1,v_1,w_1,m_1)\cap\mathcal{U}(r_2,v_2,w_2,m_2)\neq\phi$ if w_1 and w_2 are on the same path, that is, $w_1\preceq w_2$ or $w_2\preceq w_1$. Let $b\in\{0,1\}$. Since $(r_b,v_b,w_b,m_b)\in\mathcal{S}_{\bar{r}}$, it holds that v_b is a vertex in the unique decomposition of some vertex w' with respect to $x,y,\bar{r}$. Since v_b is an ancestors of both w_b and w', the vertices in the unique decomposition of w_b and w (with respect to $x,y,\bar{r}$), that are ancestors of v_b are the same. Therefore, v_b is a vertex in the unique decomposition of w_b with respect to $x,y,\bar{r}$. Furthermore, v_b must be the last vertex (excluding w_b itself) in the unique decomposition of w_b with respect to $x,y,\bar{r}$. Therefore, since $t(w_1)=t(w_2)$ and since w_1 and w_2 are on the same path, it must be the case that $v_1=v_2$. If v_1 is the t^{th} vertex in the unique decomposition of w', then it is the case that $r_1=r_2=\bar{r}_{t+1}$. This implies that for every $u\in\mathcal{L}(v_1)$ it holds that $Mid_{r_1}^{leader(v_1)}(x,y,v_1,u)=Mid_{r_2}^{leader(v_2)}(x,y,v_2,u)$. Therefore, $u\in\mathcal{U}(r_1,v_1,w_1,m_1)\cap\mathcal{U}(r_2,v_2,w_2,m_2)$ implies that $w_1=w_2$ and $m_1=m_2$. A contradiction to the assumption that $(r_1,v_1,w_1,m_1)\neq(r_2,v_2,w_2,m_2)$. □

Let $\bar{r}\in\mathcal{R}$. To conclude the proof, we compute the following bound

$$\begin{aligned}&P_{x,y}(\mathcal{L}^{\mathcal{B},\bar{r}})\\ &\le\sum_{(r,v,w,m)\in\mathcal{B}\cap\mathcal{S}_{\bar{r}}}P_{x,y}(\mathcal{U}^*(r,v,w,m))\\ &\le 8\delta\sum_{(r,v,w,m)\in\mathcal{B}\cap\mathcal{S}_{\bar{r}}}\mu_{t(w)}(\mathcal{U}(r,v,w,m)) && (\text{by Claim 10})\\ &\le 8\delta\sum_{t\in\{0,1,\ldots,2N+2\}}\ \sum_{(r,v,w,m)\in\mathcal{B}\cap\mathcal{S}_{\bar{r}}:\ t(w)=t}\mu_t(\mathcal{U}(r,v,w,m))\\ &\le 8\delta(2N+3)\\ &\qquad(\text{by Claim 11 and as } \mu_t \text{ is a probability distribution})\\ &\le 20\delta N.\end{aligned}$$

□

Proof of Lemma 8.1. Denote by τ^* the protocol obtained from τ by implementing the *Pred* and the *CorrelatedSampling* protocols in a way that ensures that they never err or fail. In the same way, we obtain the protocol $(\tau')^*$ from τ'. Recall that *Pred* and *CorrelatedSampling* err with probability at most $\varepsilon\cdot I^{-5}$.

We first claim that $\mathbf{E}\left[|\tau'(X,Y)-(\tau')^*(X,Y)|\right]<0.1\varepsilon$. Since the loop in Line 7 iterates at most N'' times when running τ', the *CorrelatedSampling* protocol is executed at most $N''=O(I^2)$ times by τ', and thus the probability of an error in at least one of these execution in at most I^{-2}. Since the loop in Line 2 of τ' iterates at most N' times when running τ', the protocols Mid^A and Mid^B are called at most $N'=O(I)$ times. Each such execution of Mid^A or Mid^B executes the *Pred* protocol at most $O(N)=O(I)$ times. Therefore, *Pred* is executed at most $O(I^2)$ times by τ', and thus the probability of an error in at least one of these execution in at most I^{-2}.

To prove the claim we use the triangle inequality,

$$\begin{aligned}&\mathbf{E}\left[|\tau'(X,Y)-\pi(X,Y)|\right]\\ &\le\mathbf{E}\left[|\tau'(X,Y)-(\tau')^*(X,Y)|\right]+\mathbf{E}\left[|(\tau')^*(X,Y)-\tau^*(X,Y)|\right]\\ &\quad+\mathbf{E}\left[|\tau^*(X,Y)-\pi'(X,Y)|\right]+\mathbf{E}\left[|\pi'(X,Y)-\pi(X,Y)|\right].\end{aligned}$$

As explained above, the first term on the right hand side is upper bounded by 0.1ε. The second term is upper bounded by 0.2ε by Lemma 8.2. The third term is zero by Lemma 7.1. The last term is upper bounded by 0.1ε by Claim 3. □

9 COMMUNICATION COST OF τ'

Claim 12. *Let x be an input. Let $i'\le i\in\mathbb{N}$. Let $\mathcal{F}_{i'-1,x}\preceq v\preceq\mathcal{F}_{i',x}$ and $w\in\mathcal{V}(v)\cap\mathcal{F}_{i,x}$. It holds that*

$$\log\left(\frac{P_{v,x}(w)}{P_v(w)}\right)\le O(I).$$

Proof. By Claim 2, $i - i' \leq O(I)$. For $j \in \{i', \ldots, i\}$, let $v_j = (\mathcal{F}_{j,x} \cap \leq w)$. By Equation (3), for $j \in \{i', \ldots, i-1\}$, it holds that $\frac{P_{v_j,x}(v_{j+1})}{P_{v_j}(v_{j+1})} \leq 4$. Using Claim 1, $\frac{P_{v,x}(v_{i'})}{P_v(v_{i'})} \leq 8$. Conclude that

$$\begin{aligned}
&\log\left(\frac{P_{v,x}(w)}{P_v(w)}\right)\\
&= \log\left(\frac{P_{v,x}(v_{i'})}{P_v(v_{i'})} \cdot \prod_{j \in \{i',\ldots,i-1\}} \frac{P_{v_j,x}(v_{j+1})}{P_{v_j}(v_{j+1})}\right)\\
&= \log\left(\frac{P_{v,x}(v_{i'})}{P_v(v_{i'})}\right) + \sum_{j \in \{i',\ldots,i-1\}} \log\left(\frac{P_{v_j,x}(v_{j+1})}{P_{v_j}(v_{j+1})}\right) \leq O(I).
\end{aligned}$$

□

Lemma 9.1. *The correlated sampling step in Line 6 of τ' communicates $O(T)$ bits in expectation.*

Proof. Let $t \in \mathbb{N}$. Assume without loss of generality that $leader(V_t) = A$. Consider any execution of the correlated sampling step in Line 6 of τ, that is after the t^{th} update of the value of V, but before the $(t+1)^{st}$ update. It holds that

$$\begin{aligned}
\mathbb{D}\left(P^A_{t+1} \| P_{\mathcal{L}(V_t)}\right) &= \mathop{\mathbf{E}}_{u \leftarrow P^A_{t+1}}\left[\log\left(\frac{P^A_{t+1}(u)}{P_{V_t}(u)}\right)\right] \quad (11)\\
&= \mathop{\mathbf{E}}_{u \leftarrow P^A_{t+1}}\left[\log\left(\frac{P_{V_t,x}(u)}{P_{V_t,x}(\mathcal{U}_t) \cdot P_{V_t}(u)}\right)\right]\\
&= \mathop{\mathbf{E}}_{u \leftarrow P^A_{t+1}}\left[\log\left(\frac{P_{V_t,x}(u)}{P_{V_t}(u)}\right) + \log\left(\frac{1}{P_{V_t,x}(\mathcal{U}_t)}\right)\right]\\
&\leq O(I) + \log\left(\frac{1}{P_{V_t,x}(\mathcal{U}_t)}\right) \quad \text{(by Claim 12)}
\end{aligned}$$

Recall that we have fixed x, y. Consider an execution of τ where $\bar{r} \in \mathcal{R}$ is the randomness used for the executions of Mid^A and Mid^B by τ. We bound the second term on the right hand side of the last inequality as follows

$$\begin{aligned}
&\mathbf{E}\left[\log\left(\frac{1}{P_{V_t,x}(\mathcal{U}_t)}\right)\right] \quad (12)\\
&= \mathop{\mathbf{E}}_{V_{t-1},V_t,M_t}\left[\log\left(\frac{1}{P_{V_t,x}(\mathcal{U}(\bar{r}_t, V_{t-1}, V_t, M_t))}\right)\right]\\
&= \mathop{\mathbf{E}}_{V_{t-1}}\left[\sum_{m \in \{0,1\}^T, w \in \mathcal{V}(V_{t-1})} \Pr[V_t = w, M_t = m | V_{t-1}] \cdot \right.\\
&\qquad \left. \log\left(\frac{1}{P_{w,x}(\mathcal{U}(\bar{r}_t, V_{t-1}, w, m))}\right)\right]\\
&= \mathop{\mathbf{E}}_{V_{t-1}}\left[\sum_{m \in \{0,1\}^T, w \in \mathcal{V}(V_{t-1})} \Pr[U \in \mathcal{U}^*(\bar{r}_t, V_{t-1}, w, m) | V_{t-1}] \cdot \right.\\
&\qquad \left. \log\left(\frac{1}{P_{w,x}(\mathcal{U}(\bar{r}_t, V_{t-1}, w, m))}\right)\right]\\
&\leq \mathop{\mathbf{E}}_{V_{t-1}}\left[\sum_{m \in \{0,1\}^T, w \in \mathcal{V}(V_{t-1})} P_{V_{t-1},x,y}(\mathcal{U}^*(\bar{r}_t, V_{t-1}, w, m)) \cdot \right.\\
&\qquad \left. \log\left(\frac{1}{P_{w,x}(\mathcal{U}(\bar{r}_t, V_{t-1}, w, m))}\right)\right]\\
&\qquad \text{(by Lemma 7.1)}\\
&\leq 8 \mathop{\mathbf{E}}_{V_{t-1}}\left[\sum_{m \in \{0,1\}^T, w \in \mathcal{V}(V_{t-1})} P_{V_{t-1},x}(\mathcal{U}^*(\bar{r}_t, V_{t-1}, w, m)) \cdot \right.\\
&\qquad \left. \log\left(\frac{1}{P_{V_{t-1},x}(\mathcal{U}(\bar{r}_t, V_{t-1}, w, m))}\right)\right]\\
&\qquad \text{(by Claim 8)}\\
&\leq 8 \mathop{\mathbf{E}}_{V_{t-1}}\left[\sum_{m \in \{0,1\}^T, w \in \mathcal{V}(V_{t-1})} P_{V_{t-1},x}(\mathcal{U}^*(\bar{r}_t, V_{t-1}, w, m)) \cdot \right.\\
&\qquad \left. \log\left(\frac{1}{P_{V_{t-1},x}(\mathcal{U}^*(\bar{r}_t, V_{t-1}, w, m))}\right)\right]\\
&\qquad \text{(as } \mathcal{U}^*(\bar{r}_t, V_{t-1}, w, m) \subseteq \mathcal{U}(\bar{r}_t, V_{t-1}, w, m))\\
&\leq O(T),
\end{aligned}$$

where the last inequality holds because $m \neq m'$ implies that

$$\mathcal{U}^*(\bar{r}_t, V_{t-1}, w, m) \cap \mathcal{U}^*(\bar{r}_t, V_{t-1}, w, m') = \phi,$$

and as for any random variable Z over k bits it holds that $\mathbb{H}(Z) \leq k$.

By [7], the expected number of bit communicated by the correlated sampling step in Line 6 of τ' is upper bounded by

$$O(\mathbf{E}\left[\mathbb{D}\left(P^A_t \| P_{\mathcal{L}(V_{t-1})}\right)\right] + I).$$

Equations (11) and (12) give a bound of $O(I + T + I) = O(T)$. □

Claim 13. *The protocol τ' communicates at most $O(I^2 \cdot T) = O(I^3 \log(I) \log\log(C))$ bits in expectation.*

Proof. During an execution of τ', the loop in Line 2 iterates at most N' times. Each such iteration executes Mid^A or Mid^B once (Line 9). By Lemma 4, Mid^A and Mid^B communicate at most $O(T)$ bits in expectation. Therefore, the total expected number of bits communicated by the executions of Mid^A and Mid^B during an execution of τ' is $O(N' \cdot T) = O(I \cdot T)$.

During an execution of τ', the loop in Line 7 iterates at most N'' times. Each such iteration executes the *CorrelatedSampling* protocol once (Line 6). By Lemma 9.1, the correlated sampling protocol communicates at most $O(T)$ bits in expectation. Therefore, the total expected number of bits communicated by the executions of *CorrelatedSampling* during an execution of τ' is $O(N'' \cdot T) = O(I^2 \cdot T)$. □

REFERENCES

[1] Reuven Bar-Yehuda, Benny Chor, Eyal Kushilevitz, and Alon Orlitsky. 1993. Privacy, Additional Information, and Communication. *IEEE Transactions on Information Theory* 39 (1993), 55–65.

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

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

[4] Balthazar Bauer, Shay Moran, and Amir Yehudayoff. 2015. Internal Compression of Protocols to Entropy. In *APPROX/RANDOM*. 481–496.

[5] Mark Braverman. 2012. Interactive information complexity. In *STOC*. 505–524.

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

[7] Mark Braverman and Anup Rao. 2011. Information Equals Amortized Communication. In *FOCS*. 748–757.

[8] Mark Braverman and Omri Weinstein. 2015. An Interactive Information Odometer and Applications. In *STOC*. 341–350.

[9] Joshua Brody, Harry Buhrman, Michal Koucký, Bruno Loff, Florian Speelman, and Nikolay K. Vereshchagin. 2013. Towards a Reverse Newman's Theorem in Interactive Information Complexity. In *CCC*. 24–33.

[10] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. 2001. Informational Complexity and the Direct Sum Problem for Simultaneous Message Complexity. In *FOCS*. 270–278.

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

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

[13] David A. Huffman. 1952. A method for the construction of minimum redundancy codes. *proc. IRE* 40, 9 (1952), 1098–1101.

[14] Amiram H. Kaspi. 1985. Two-way Source Coding with a Fidelity Criterion. *IEEE Transactions on Information Theory* 31, 6 (1985), 735–740.

[15] Gillat Kol. 2016. Interactive compression for product distributions. In *STOC*. 987–998.

[16] Eyal Kushilevitz and Noam Nisan. 1997. Communication complexity. *Cambridge University Press* (1997).

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

[18] Sivaramakrishnan Natarajan Ramamoorthy and Anup Rao. 2015. How to Compress Asymmetric Communication. In *CCC*. 102–123.

[19] Claude E. Shannon. 1948. A mathematical theory of communication. *The Bell Systems Technical Journal* 27 (1948), July 379–423, October 623–656.

[20] Alexander A. Sherstov. 2016. Compressing interactive communication under product distributions. *FOCS* (2016).

[21] Omri Weinstein. 2015. Information Complexity and the Quest for Interactive Compression. *SIGACT News* 46, 2 (2015), 41–64.