Electronic Colloquium on Computational Complexity, Report No. 78 (2014)

Zero-Information Protocols and Unambiguity in Arthur–Merlin Communication

Mika Göös Toniann Pitassi Thomas Watson
Department of Computer Science, University of Toronto

Abstract

We study whether information complexity can be used to attack the long-standing open problem of proving lower bounds against Arthur–Merlin (AM) communication protocols. Our starting point is to show that—in contrast to plain randomized communication complexity—every boolean function admits an AM communication protocol where on each yes-input, the distribution of Merlin's proof leaks no information about the input and moreover, this proof is unique for each outcome of Arthur's randomness. We posit that these two properties of *zero information leakage* and *unambiguity on yes-inputs* are interesting in their own right and worthy of investigation as new avenues toward AM.

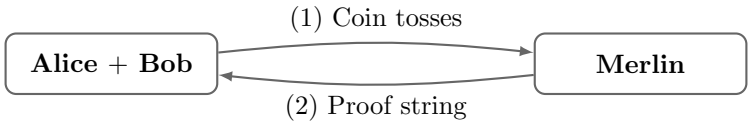
- **Zero-information protocols (ZAM).** Our basic ZAM protocol uses exponential communication for some functions, and this raises the question of whether more efficient protocols exist. We prove that all functions in the classical space-bounded complexity classes NL and $\oplus$L have polynomial-communication ZAM protocols. We also prove that ZAM complexity is lower bounded by conondeterministic communication complexity.
- **Unambiguous protocols (UAM).** Our most technically substantial result is a $\Omega(n)$ lower bound on the UAM complexity of the NP-complete *set-intersection* function; the proof uses information complexity arguments in a new, indirect way and overcomes the "zero-information barrier" described above. We also prove that in general, UAM complexity is lower bounded by the classic discrepancy bound, and we give evidence that it is *not* generally lower bounded by the classic corruption bound.

ISSN 1433-8092

1 Introduction

What is AM communication? Arthur–Merlin (AM) games [BM88] are a type of randomized proof system where a computationally-unbounded prover, Merlin, wishes to convince a skeptical and computationally-bounded verifier, Arthur, that some boolean function f evaluates to 1 on a given input. In this work, we study the communication complexity variant of AM [BFS86, Kla03, Kla11] where "Arthur" now consists of two parties, Alice and Bob, and the input is split between them: Alice holds x, Bob holds y, and they wish to verify that $f(x, y) = 1$.

In an execution of an AM communication protocol, Alice and Bob start by tossing some coins, then Merlin produces a proof string that may depend on the input and the outcomes of the coin tosses, and finally Alice and Bob independently decide whether to accept based on their own input, the outcome of the coin tosses, and Merlin's proof string.



The *completeness* criterion is that for every 1-input, with high probability over the coin tosses there exists a proof string that *both* Alice and Bob accept. The *soundness* criterion is that for every 0-input, with high probability over the coin tosses there does not exist a proof string that both Alice and Bob accept. The *communication cost* is the worst-case length of Merlin's proof string.

In short, an AM protocol is a probability distribution over nondeterministic protocols, together with a bounded-error acceptance criterion. That is, $\mathsf{AM} = \mathsf{BP} \cdot \mathsf{NP}$ in standard notation [Sch89]. The model is also robust to changes in its definition; for example, allowing Alice and Bob to communicate after Merlin's proof is published does not increase the power of the model: we can simply include the transcript of the subsequent communication in Merlin's proof.

For a more formal definition of the AM communication model, see Section 2.

Why study AM communication? The Arthur–Merlin communication model marks one of the frontiers of our understanding of communication complexity: no nontrivial lower bounds are known on the amount of communication required by AM protocols for any explicit function.

The desirability of such lower bounds stems from a variety of sources. For one, AM communication has turned out to be closely related to models of streaming delegation [CCM09, KP13, GR13a, CCGT14, CCM+13, KP14b]. Also, AM lower bounds would be a first step toward proving lower bounds against the communication polynomial hierarchy [BFS86], which is necessary for obtaining strong rank rigidity lower bounds for explicit matrices [Raz89, Lok01, Lok09, Wun12b] (as well as margin complexity rigidity lower bounds [LS09]), which in turn is related to circuit complexity [Val77]. Lower bounds against the polynomial hierarchy are also related to graph complexity [PRS88, Juk06]. Another motivation comes from the *algebrization* framework [AW09, IKK09], which converts communication lower bounds (such as for AM) into barriers to proving relations among classical, time-bounded complexity classes. The absence of and need for nontrivial AM communication lower bounds has been explicitly pointed out in [Lok01, LS09, PSS14, KP14b, KP14a].

For MA, the weaker variant where Merlin sends his proof *before* the coins are tossed, strong lower bounds are known [Kla03, RS04, GS10, Kla11, GR13a] (with applications to property testing [GR13b]). Other powerful subclasses of the polynomial hierarchy for which communication lower bounds are known include SBP (which lies between MA and AM) [GW14] and P^{NP} [IW10, PSS14].

1.1 New models UAM and ZAM

The aim of this work is to study restricted complexity measures that capture some of the difficulty of proving AM lower bounds, and to create new proof techniques and explore the power of existing ones with regard to AM communication complexity. Our results revolve around two new complexity measures UAM and ZAM that we introduce below. We proceed rather informally in this introduction; for precise definitions, see Section 2.

Unambiguous protocols (UAM). A natural restriction on any proof system is *unambiguity*, meaning that the verifier accepts at most one proof on a given input. In the context of AM one can consider three types of unambiguity: (1) *unambiguous completeness*, where for each 1-input and each outcome of the coin tosses, Arthur accepts at most one of Merlin's possible proofs; (2) *unambiguous soundness*, which is the same as above but for 0-inputs; and (3) *two-sided unambiguity*, where both unambiguous completeness and soundness hold.

Lower bounds for models (2) and (3) can be proved using known techniques. In case of (2) it is not difficult to show that lower bounds follow from the classic corruption bound, which is known to characterize the complexity class SBP [GW14]. In case of (3) the complexity class corresponding to the model is BP$\cdot$UP. Klauck [Kla10] showed that even the smooth rectangle bound (introduced in [JK10]) suffices to lower bound BP$\cdot$UP communication complexity.

In this work we study the remaining case of unambiguous completeness, henceforth simply called *unambiguous*. We call an unambiguous AM protocol a UAM protocol for short, and we let UAM(f) denote the minimum cost of a UAM protocol for f. As is customary, we also use UAM to denote the class of two-party functions that admit polylog cost UAM protocols. We will see that UAM exhibits new phenomena not captured by SBP or BP$\cdot$UP. For starters, the model supports *zero-information* protocols as introduced next.

Zero-information protocols (ZAM). One very successful approach for proving communication lower bounds against randomized protocols is the *information complexity* methodology [CSWY01, BYJKS04, JKS03, CKS03, Gro09, Jay09, DKS12, BM13, BGPW13, BEO$^+$13, GW14]. In this approach one argues that the transcripts of correct protocols must necessarily "leak" information about the input; the amount of information leaked automatically lower bounds the communication.

A natural question is whether information complexity has any bearing on AM. One of the main conceptual contributions of this work is that information complexity, in its standard form, cannot be used to prove lower bounds against UAM protocols (much less against AM protocols). Specifically, we show that every boolean function admits a private-coin UAM protocol satisfying the following:

> **Zero-information:** *The distribution of Merlin's unique proof (which serves as the protocol transcript) is identical across all* 1*-inputs.*

We call a zero-information UAM protocol a ZAM protocol for short, and we let ZAM(f) denote the minimum cost of a ZAM protocol for f. We posit that ZAM protocols are interesting in their own right, both combinatorially and as a natural model of private computation in which Alice and Bob enlist Merlin's help in computing a function but do not want an external observer to learn anything about their inputs.

When talking about information complexity, it is most natural to consider *private-coin* protocols, where Alice and Bob only know the outcomes of their own coins (and Merlin sees everything), rather

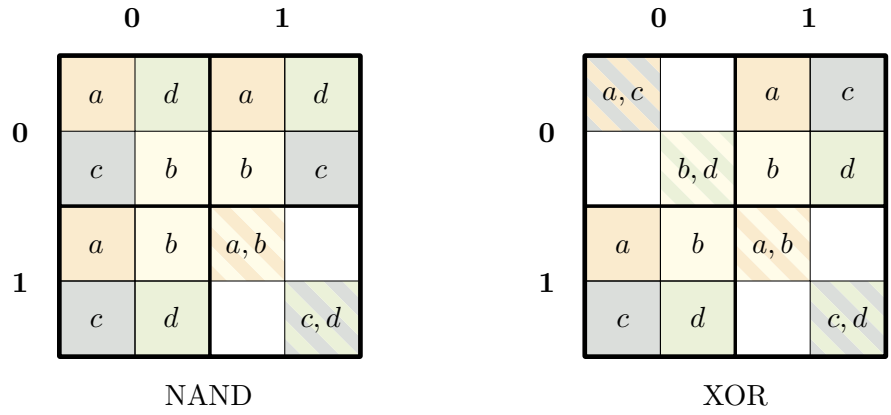


Figure 1: Two examples of ZAM protocols.

than *public-coin* protocols, where Alice and Bob share a source of randomness. Indeed, private-coin protocols arise naturally in the direct sum methodology of information complexity.

1.2 Two examples of ZAM protocols

For the sake of concreteness, let us get acquainted with zero-information protocols by studying two basic examples. Figure 1 defines private-coin AM protocols for the 2-bit functions NAND and XOR. The outer 2×2 grids correspond to the inputs x and y, while the 2×2 grid within each input block corresponds to the outcomes of the private coins (each party uses 1 bit of randomness). Both protocols use four different proofs with labels a, b, c, and d; each proof corresponds to a rectangle in the figures.

To execute such an AM protocol on an input $(x, y) \in \{0,1\}^2$ we first choose outcomes for the private coins: Alice chooses $r \in \{0,1\}$ at random and Bob chooses $q \in \{0,1\}$ at random. The input and the coin tosses now define a point (i.e., a smallest square) $P = ((x, r), (y, q))$ inside our figure. If the point P is covered by some rectangle $R \in \{a, b, c, d\}$, then Merlin can make Alice and Bob accept: he provides the label of the rectangle R as proof and both Alice and Bob can verify that $P \in R$ by checking that this holds from their own perspective. If the point P is not covered by any rectangle, then there is no way for Merlin to make both Alice and Bob accept simultaneously.

The two protocols are unambiguous since no two rectangles intersect inside a 1-block (block corresponding to a 1-input). The protocols make no errors on 1-inputs, i.e., they achieve *perfect completeness*, since they cover each 1-block fully. They are also zero-information, because all rectangles appear with the same "area" (i.e., same probability) inside each of the 1-blocks; hence, for each 1-input, Merlin's unique proof will be uniformly distributed over the set $\{a, b, c, d\}$ (though the definition of zero-information does not require the distribution to be uniform). On 0-inputs, the protocols can erroneously accept with probability $1/2$, i.e., their *soundness* is $1/2$, since in each 0-block the protocols cover half of the points. On uncovered points, Alice or Bob will reject, regardless of which proof Merlin sends. Some points are covered multiple times; e.g., in the case of $(1, 1) \in \text{NAND}^{-1}(0)$ the rectangles a and b intersect, as do c and d.

If we want to obtain protocols with soundness $1/2^k$ we can repeat the protocols independently k times in parallel and require that all k iterations accept. In a k-fold protocol the proofs are labeled with k-tuples from $\{a, b, c, d\}^k$. Note also that the iterated protocols retain their unambiguity, zero-information, and perfect completeness properties.

Function f	(notation)	Bounds on $\mathsf{ZAM}(f)$	Bounds on $\mathsf{UAM}(f)$
equality	EQ	$\Theta(\log n)$	$\Theta(\log n)$
non-equality	NEQ	$\Theta(n)$	$\Theta(\log n)$
greater-than	GT	$\Theta(n)$	$\Theta(\log n)$
set-disjointness	DISJ	$\Omega(\log n)$ and $O(n)$	$\Omega(\log n)$ and $O(n)$
set-intersection	INTER	$\Theta(n)$	$\Theta(n)$
inner-product	IP	$\Theta(n)$	$\Theta(n)$
random functions		$\Omega(n)$ and $O(2^n)$	$\Theta(n)$
functions in NL or $\oplus\mathsf{L}$		$O(\mathrm{poly}(n))$	

Table 1: Bounds on the ZAM and UAM complexities of basic problems.

1.3 Results for ZAM

Our starting point is to show that there exists a ZAM protocol for any two-party function $f\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$. Unlike most communication complexity measures, it is not obvious that linear communication suffices for a ZAM protocol, and in fact, our general ZAM protocol uses exponential communication, i.e., we only obtain $\mathsf{ZAM}(f)\le O(2^n)$ in general when $\mathcal{X}=\mathcal{Y}=\{0,1\}^n$. We can improve on this general upper bound in case f can be computed in *small space*. To express this result we use a certain measure of parity branching program size $\oplus\mathsf{BP}(f)$ that is tailored for two-party functions; we postpone the precise definition until the proof.

Theorem 1. $\mathsf{ZAM}(f)\le O(\oplus\mathsf{BP}(f))$ *for all* f. *(Section 4)*

In particular, Theorem 1 implies $O(n)$-communication ZAM protocols for all the natural functions listed in Table 1. All functions in the classical space-bounded nonuniform complexity class $\oplus\mathsf{L/poly}$ have polynomial-size parity branching programs by definition. It is known that $\mathsf{NL}\subseteq\oplus\mathsf{L/poly}$ [GW96] (moreover, NL/poly equals its unambiguous analogue UL/poly [RA00]), and thus all functions in the classical classes NL and $\oplus\mathsf{L}$ have polynomial-communication ZAM protocols. Although Theorem 1 does not seem to yield interesting examples of ZAM protocols with sublinear communication, we show that such a protocol exists at least for the equality function: $\mathsf{ZAM}(\mathrm{EQ})\le O(\log n)$.

As for lower bounds, we prove the following.

Theorem 2. $\mathsf{ZAM}(f)\ge\Omega(\mathsf{coNP}(f))$ *for all* f. *(Section 5)*

In particular, Theorem 2 allows us to derive matching lower bounds on the ZAM complexity of almost all the functions listed in Table 1; the exceptional DISJ function is discussed shortly. Interestingly, NEQ and GT demonstrate that privacy can come at a huge cost, since $\mathsf{UAM}(\mathrm{NEQ})\le\mathsf{BPP}(\mathrm{NEQ})\le O(\log n)$ and similarly for GT, and thus there is an exponential separation between ZAM and UAM.

It remains open to show that there exists a function (even a random one!) that requires superlinear ZAM communication, or prove that all functions have subexponential-communication ZAM protocols. This situation is similar to [FKN94], which studies a different model of private two-party computation, and where the best upper and lower bounds are also exponential and linear. In a similar spirit, [ACC+14] proves that in a communication model of approximate privacy called PAR (based on [Kus92]), privacy can come at an exponential cost.

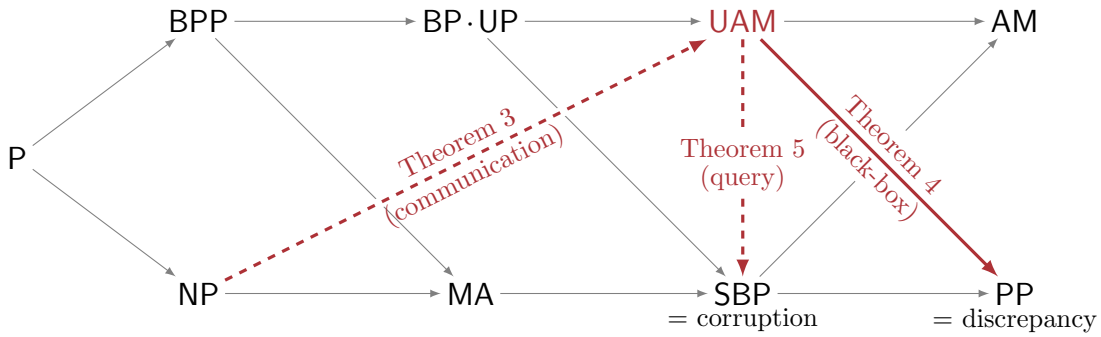


Figure 2: Our results for UAM at a glance. In this diagram, solid arrows $\mathsf{A} \longrightarrow \mathsf{B}$ indicate class inclusions $\mathsf{A} \subseteq \mathsf{B}$, and dashed arrows $\mathsf{A} \dashrightarrow \mathsf{B}$ indicate non-inclusions $\mathsf{A} \not\subseteq \mathsf{B}$.

1.4 Results for UAM

Our results for UAM are summarized in Figure 2. Our most technically substantial contribution is a linear lower bound on the UAM complexity of *set-intersection* $\textsc{Inter}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ defined by $\textsc{Inter}(x,y) = 1$ iff x and y intersect when viewed as subsets of $[n]$. Recall that INTER is the canonical NP-complete problem in communication complexity.

Theorem 3. $\mathsf{UAM}(\textsc{Inter}) = \Theta(n)$. *(Section 6)*

Because of the existence of ZAM protocols for INTER, it is not possible to prove Theorem 3 by lower bounding the standard measure of information complexity. Nevertheless, we develop a technique for employing information complexity tools in an *indirect* way, which is inspired by our ZAM lower bounds. Theorem 3 strengthens a result of Klauck [Kla10], who proved that $\mathsf{BP{\cdot}UP}(\textsc{Inter}) = \Theta(n)$ using the smooth rectangle bound of [JK10] (recall that BP·UP is *two-sided* unambiguous AM). By the methods of [AW09], a corollary to Theorem 3 is that proving $\mathsf{NP} \subseteq \mathsf{UAM}$ in the classical time-bounded world would require non-algebrizing techniques. We remark that Theorem 3 also holds for the promise problem where the input sets x and y are guaranteed to intersect in at most two coordinates. This is tight: the UAM complexity of INTER is $O(\log n)$ under the promise that x and y intersect in at most a single coordinate.

One of the most classical lower bound techniques in communication complexity is *discrepancy*, which characterizes PP [Kla07]. Klauck [Kla11] showed that discrepancy does not yield AM lower bounds in general, in other words $\mathsf{AM} \not\subseteq \mathsf{PP}$ (for promise problems). We prove that discrepancy *can* be used for UAM.

Theorem 4. $\mathsf{UAM}(f) \geq \Omega(\mathsf{PP}(f))$ *for all* f. *(Section 7)*

The proof of Theorem 4 is via a general "black-box" simulation, in the sense that it does not exploit any specific properties of communication complexity and works equally well for other models such as time-bounded computation. Note that Theorem 3 does not follow from Theorem 4, since $\mathsf{PP}(\textsc{Inter}) = \Theta(\log n)$. However, all the other UAM lower bounds in Table 1 can be derived as corollaries of Theorem 4; see Section 7 for details.

Since discrepancy can be used for UAM lower bounds, it is natural to ask whether the similarly-prominent *corruption* bound (a one-sided version of discrepancy) can also be used. Since corruption characterizes SBP [GW14], this is equivalent to asking whether $\mathsf{UAM}(f)$ can be reasonably lower

bounded in terms of $\mathsf{SBP}(f)$, for all f. If so, this would be very significant as it would lead to a lower bound on the UAM complexity of *set-disjointness* $\textsc{Disj} = \neg\textsc{Inter}$ (which is conjectured to require linear AM communication) by the corruption bound for $\textsc{Disj}$ due to [Raz92]. Currently we cannot even prove that $\mathsf{ZAM}(\textsc{Disj}) \geq \omega(\log n)$. However, we conjecture that corruption alone is *not* sufficient to lower bound UAM complexity, i.e., we conjecture that $\mathsf{UAM} \not\subseteq \mathsf{SBP}$. While we are unable to prove this separation for communication complexity, we prove it for *query complexity* (which is a necessary step in order for the communication complexity separation to hold).

In general, an SBP computation (introduced in [BGM06]) is a randomized computation where the acceptance probability is at least α on 1-inputs and at most $\alpha/2$ on 0-inputs, for some arbitrarily small threshold $\alpha > 0$ which may depend on the input size. We provide formal definitions of the query complexity measures $\mathsf{UAM}^{\mathsf{dt}}(f)$ and $\mathsf{SBP}^{\mathsf{dt}}(f)$ in Section 8, but for now it suffices to say that they are natural decision tree analogues of the corresponding communication measures. We define a partial function called $\textsc{Gut}$ (short for *gap-unique-tribes*) and prove the following separation.

Theorem 5. $\mathsf{UAM}^{\mathsf{dt}}(\textsc{Gut}) \leq O(1)$ *and* $\mathsf{SBP}^{\mathsf{dt}}(\textsc{Gut}) \geq \Omega(n^{1/4})$. *(Section 8)*

We note that this result yields (by standard techniques) an alternative proof that there exists an oracle separating the classical time-bounded complexity classes MA and AM, which was first proved in [San89]. We also note that the composed two-party function $\textsc{Gut} \circ \textsc{And}^n$ is a natural candidate to witness our conjectured communication separation $\mathsf{UAM} \not\subseteq \mathsf{SBP}$.

2 Definitions

We let $\mathbb{P}$ denote probability, $\mathbb{E}$ denote expectation, $\mathbb{H}$ denote Shannon entropy, $\mathbb{I}$ denote mutual information, and $[k]$ denote $\{1, 2, \ldots, k\}$.

2.1 Communication complexity

We assume some familiarity with basic definitions of communication complexity; see [KN97, Juk12]. We consider two-party functions $f\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, where Alice is given the first part of the input $x \in \mathcal{X}$ and Bob is given the second part of the input $y \in \mathcal{Y}$. For $b \in \{0,1\}$, a b-input is a pair $(x,y) \in f^{-1}(b)$. We adopt the convenient notation of using complexity class names as complexity measures. For example:

- $\mathsf{P}(f)$ is the minimum over all deterministic protocols for f of the maximum number of bits communicated on any input.
- $\mathsf{coNP}(f)$ is the ceiling of the log of the minimum number of rectangles needed to cover the 0's of the communication matrix of f.
- $\mathsf{PP}(f)$ is the minimum over all $\epsilon > 0$ and all randomized protocols computing f with error $\leq 1/2 - \epsilon$ of the communication cost of the protocol plus $\log(1/\epsilon)$; see also [BFS86].
- $\mathsf{SBP}(f)$ is the minimum over all $\alpha > 0$ and all randomized protocols that accept 1-inputs with probability $\geq \alpha$ and 0-inputs with probability $\leq \alpha/2$ of the communication cost of the protocol plus $\log(1/\alpha)$; see also [GW14].

We let $\textsc{Eq}_n, \textsc{Neq}_n, \textsc{Gt}_n, \textsc{Disj}_n, \textsc{Inter}_n, \textsc{Ip}_n\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ denote the equality, non-equality, greater-than, set-disjointness, set-intersection, and inner-product modulo 2 functions, respectively. We have that $\textsc{Disj}_n(x,y) = \bigwedge_i \neg(x_i \wedge y_i) = \textsc{And}_n \circ \textsc{Nand}^n(x,y)$ is coNP-complete,

$\textsc{Inter}_n(x,y) = \bigvee_i (x_i \wedge y_i) = \mathrm{OR}_n \circ \mathrm{AND}^n(x,y)$ is NP-complete, and $\textsc{Ip}_n(x,y) = \bigoplus_i (x_i \wedge y_i) = \mathrm{XOR}_n \circ \mathrm{AND}^n(x,y)$ is $\oplus\mathsf{P}$-complete. In all cases, we may omit the subscript n when there is no confusion.

2.2 AM, UAM, and ZAM

We work exclusively with *private-coin* AM protocols. Private-coin protocols are essentially equally powerful to their *public-coin* counterparts; see Remark 1 below. We stress that Merlin can always see all the outcomes of coin tosses, and "private" and "public" refer only to whether Alice and Bob can see each other's randomness. (This is in contrast to classical time-bounded complexity where private and public often refer to whether Merlin can see Arthur's randomness.)

In an AM protocol Π, Alice is given a uniform sample from some finite set $\mathcal{R}$ and Bob is independently given a uniform sample from some finite set $\mathcal{Q}$, and there is a collection of rectangles $\tau_1, \ldots, \tau_m \subseteq (\mathcal{X} \times \mathcal{R}) \times (\mathcal{Y} \times \mathcal{Q})$. (Recall that a rectangle τ_i is of the form $A \times B$ for some $A \subseteq \mathcal{X} \times \mathcal{R}$ and $B \subseteq \mathcal{Y} \times \mathcal{Q}$, or equivalently that for all u and u' in $\mathcal{X} \times \mathcal{R}$ and all v and v' in $\mathcal{Y} \times \mathcal{Q}$, if (u, v') and (u', v) are in τ_i, then so are (u, v) and (u', v').) The acceptance probability of Π on input (x, y) is defined to be $\mathbb{P}_{r \in \mathcal{R}, q \in \mathcal{Q}}[\exists i : ((x, r), (y, q)) \in \tau_i]$, and we refer to the set $(\{x\} \times \mathcal{R}) \times (\{y\} \times \mathcal{Q})$ as the *block* corresponding to input (x, y). The index i of a rectangle τ_i can be thought of as a message sent from Merlin to Alice and Bob, who then independently decide whether they accept. The output of the protocol is 1 iff they both accept. We use the terminology "rectangles", "transcripts", and "proofs" interchangeably (τ stands for "transcript"). We define the *communication cost* of Π to be $|\Pi| := \lceil \log m \rceil$, the length of Merlin's proof. The protocol has *completeness* c and *soundness* s if the acceptance probability is at least c on 1-inputs and at most s on 0-inputs. *Perfect completeness* means $c = 1$. We define $\mathsf{AM}_{c,s}(f)$ to be the minimum of $|\Pi|$ over all AM protocols Π for f with completeness c and soundness s.

We say an AM protocol Π is *unambiguous* (more precisely: has unambiguous completeness), or that Π is a UAM protocol, if for every 1-input and every outcome of the randomness, there is at most one proof of Merlin that causes Alice and Bob to accept (and on 0-inputs, in the unlikely event that there exists a proof that is accepted, there can be any number of such proofs). In other words, rectangles do not overlap on 1-inputs; more formally, $((x, r), (y, q)) \notin \tau_i \cap \tau_j$ holds for all $(x, y) \in f^{-1}(1)$, $r \in \mathcal{R}$, $q \in \mathcal{Q}$, and $i \neq j$. We define $\mathsf{UAM}_{c,s}(f)$ to be the minimum of $|\Pi|$ over all UAM protocols Π for f with completeness c and soundness s.

On 1-inputs, a UAM protocol Π can be viewed as a function that maps each $((x, r), (y, q))$ with $(x, y) \in f^{-1}(1)$ to the unique $i \in \{1, \ldots, m\}$ such that $((x, r), (y, q)) \in \tau_i$, or to $\perp$ if no such i exists. We say a UAM protocol Π is *zero-information*, or that Π is a ZAM protocol, if the distribution of the output of this function over random $r \in \mathcal{R}, q \in \mathcal{Q}$ is the same for all $(x, y) \in f^{-1}(1)$. We define $\mathsf{ZAM}_{c,s}(f)$ to be the minimum of $|\Pi|$ over all ZAM protocols Π for f with completeness c and soundness s.

The connection to information complexity is that a protocol is zero-information iff for any or all joint random variables (X, Y) whose support is $f^{-1}(1)$, we have $\mathbb{I}(\Pi : X, Y) = 0$ (where Π stands for the unique proof function, viewed as a random variable jointly distributed with (X, Y) and with Alice's and Bob's randomness). We consider only distributions over 1-inputs (rather than over all inputs) since (i) the proof function is not uniquely defined on 0-inputs, (ii) the known communication lower bounds via information complexity only need distributions over 1-inputs, and (iii) from a privacy perspective, this is analogous to the situation in cryptographic zero-knowledge, in which the prover's zero-knowledge property is only required to hold on 1-inputs since on 0-inputs,

the prover could misbehave and send any message he wants (including ones that reveal too much information).

We assume by default that protocols have perfect completeness and soundness $1/2$, so we define $\mathsf{AM} = \mathsf{AM}_{1,1/2}$ and $\mathsf{UAM} = \mathsf{UAM}_{1,1/2}$ and $\mathsf{ZAM} = \mathsf{ZAM}_{1,1/2}$. Note that $\mathsf{AM}(f) \leq \mathsf{UAM}(f) \leq \mathsf{ZAM}(f)$ for all f. Our upper bounds all have perfect completeness, and our lower bounds all work even for imperfect completeness (as will be clarified in the proofs).

Remark 1. We note here the well-known fact that by studying a private-coin model (e.g., $\mathsf{UAM} = \mathsf{UAM}^{\mathsf{priv}}$) we lose little generality over analogous public-coin models (e.g., $\mathsf{UAM}^{\mathsf{pub}}$). In a public-coin AM protocol, the randomness is sampled uniformly from some finite set $\mathcal{R}$, and for each $r \in \mathcal{R}$ there is a collection of rectangles $\tau_1^r, \tau_2^r, \ldots, \tau_m^r \subseteq \mathcal{X} \times \mathcal{Y}$. The acceptance probability is $\mathbb{P}_{r \in \mathcal{R}}[\exists i : (x,y) \in \tau_i^r]$, and for unambiguity we require that $(x,y) \notin \tau_i^r \cap \tau_j^r$ holds for all $(x,y) \in f^{-1}(1)$, $r \in \mathcal{R}$, and $i \neq j$. For all f we have $\mathsf{UAM}^{\mathsf{priv}}_{c',s'}(f) \leq \mathsf{UAM}^{\mathsf{pub}}_{c,s}(f) + O(\log\log|\mathcal{X} \times \mathcal{Y}|)$ by standard sparsification techniques [KN97, §3.3], provided c, s, c', s' are constants such that $s < s'$ and either $c' < c$ or $c = 1$. For AM, the same sparsification fact holds, and standard amplification renders the exact values of the constants c and s immaterial. In contrast, for UAM it is not known how to amplify c while preserving the unambiguity property (though s can be amplified if $c = 1$).

3 Overview of Proofs

Before we present the formal proofs of our Theorems 1–5, we first describe here the intuitions underlying the proofs.

3.1 Theorem 1

The simple but non-obvious fact that every two-party function f *has* a ZAM protocol (moreover, one of cost $O(2^n)$) follows by combining a generic reduction from f to Disj with a ZAM protocol for Disj that runs the ZAM protocol for NAND (from Figure 1) independently for each coordinate. Details are provided in Section 4.

To prove the stronger Theorem 1, we use a two-step process: (i) we reduce the function f to the evaluation of a determinant of a matrix of a certain form, and (ii) we design a ZAM protocol for the latter task.

We recall that Valiant [Val79] showed how to reduce any function in NC^1 to the evaluation of a determinant, and subsequently it was shown that mod-2 determinant is actually complete for $\oplus\mathsf{L}$ [Dam90]. In [IK02, AIK06], a generalization of a reduction from $\oplus\mathsf{L}$ to determinant was used (employing the parity branching program model for $\oplus\mathsf{L}$ computations). To achieve (i), we use essentially the same reduction, and we describe a simple and combinatorial (as opposed to linear-algebraic as in [IK02, AIK06]) proof of the reduction.

To achieve (ii), the basic idea is to pick a random vector and challenge Merlin to find a preimage under the linear transformation of the matrix. If the matrix has nonzero determinant then its linear transformation is a bijection, which means Merlin's proof is in one-to-one correspondence with the challenge vector and is hence uniformly distributed regardless of the matrix. If the determinant is zero then the range of the linear transformation is a proper subspace, so with probability at least half, the challenge vector has no preimage. The matrix needs to be of a certain form to

enable Alice and Bob to jointly multiply the matrix by Merlin's claimed preimage without further communication.

The idea for showing $\mathsf{ZAM}(\textsc{Eq}) \leq O(\log n)$ is just the standard approach of using an error-correcting code for a downward-random-self-reduction from equality on n bits to equality on 1 bit, and then invoking a 1-bit protocol. The reduction preserves the necessary properties.

3.2 Theorem 2

Many classical lower bound methods in communication complexity (such as discrepancy and corruption) examine the properties of *individual* rectangles. A key departure here is that we consider how *pairs* of rectangles interact with each other.

We need to show how to convert an arbitrary ZAM protocol into a conondeterministic protocol without increasing the cost by more than a constant factor. Thus, we need to be able to find rectangles that cover 0-inputs. The first key idea is the observation that for any unambiguous protocol, the intersection of two different proof rectangles is contained within the blocks of 0-inputs. Thus if we take such an intersection and "project" it to the inputs, we get a 0-monochromatic rectangle in $\mathcal{X} \times \mathcal{Y}$. We call the collection of rectangles that arise in this way the *double cover*. If we knew that the double cover covered *all* the 0-inputs, then it would be a conondeterministic protocol (with cost at most twice the cost of the ZAM protocol) and we would be done.

How can we say anything about *which* 0-inputs get covered by the double cover? This is where the zero-information assumption comes in. As a simple example, consider an arbitrary ZAM protocol for the NAND function. A proof rectangle has the same area within the $(0,0)$ and $(0,1)$ blocks, and this actually forces it to have the same *shape* (height and width) within these blocks. Similarly, the proof has the same shape within the $(0,0)$ and $(1,0)$ blocks. Hence it has the same shape in $(0,1)$ and $(1,0)$ and thus also in $(1,1)$. So every proof has the same shape (in particular, area) in all four blocks! If the proofs were pairwise disjoint in the $(1,1)$ block then we could add up their areas to find that the acceptance probability on this 0-input is the same as the acceptance probability on the 1-inputs, a contradiction.

After generalizing this idea (for an arbitrary function f), what we find is that the 0-inputs *not* covered by the double cover can be organized into a coarse "non-equality-like" structure, and can hence be covered by few rectangles, which we add to the double cover to get a low-cost conondeterministic protocol.

3.3 Theorem 3

By Theorem 2, we know that $\mathsf{ZAM}(\textsc{Inter}) \geq \Omega(n)$. The basic intuition for Theorem 3 is as follows: Building on the ideas in the proof of Theorem 2, we can prove a "robust" version for $\textsc{Inter}$, showing that $\Omega(n)$ communication is required by UAM protocols whose transcripts leak a sublinear amount of information about the input (which is a weaker assumption than zero-information). This leads to a dichotomy: Either a protocol leaks a linear amount of information, or it does not. If it does, we are done (since information cost trivially lower bounds communication cost). If it does not, we are done by the above argument. In either case, the protocol must use $\Omega(n)$ communication. However, there are several technical obstacles that need to be overcome to get this approach to work.

Similarly to Theorem 2, the basic strategy for proving that "low information implies high communication" is to find a huge number of 0-inputs that all get "double-covered" by Merlin's rectangles, and then use the fact that every 0-monochromatic rectangle is small for $\textsc{Inter}$ (hence

there must be many pairs of rectangles of Merlin). However, using information complexity techniques, what we can find is a huge number of special "windows" (which are certain submatrices of the communication matrix) such that each of these windows contains a 0-input that gets double-covered. But what if different windows share the same double-covered 0-input? Then we might not have a huge number of double-covered 0-inputs as required. This problem goes away if the special windows are disjoint. Hence, we define the distribution over inputs (with respect to which information cost is measured) in a careful, nonstandard way to enable us to get a large number of *disjoint* special windows.

Another issue is that our argument requires information cost to be measured with respect to a distribution over 1-inputs, whereas in the standard framework of [BYJKS04] the distribution is over 0-inputs of Inter. This necessitates using the (somewhat more complicated) framework of [JKS03] for analyzing the so-called *partial information cost* with respect to a distribution over 1-inputs of Inter.

3.4 Theorem 4

We first describe a way to prove a quantitatively weaker version of Theorem 4. Consider a UAM protocol, and let us say a 0-input is unambiguous if no rectangles overlap within its block, and is ambiguous otherwise. Then 1-inputs can be distinguished from ambiguous 0-inputs by a PP (indeed, coNP) protocol à la the proof of Theorem 2, and 1-inputs can be distinguished from unambiguous 0-inputs by a PP (indeed, SBP) protocol by treating the nondeterminism as randomness. Hence 1-inputs can be distinguished from all 0-inputs by using the fact that PP is closed under intersection [BRS95, Wun12a].

The disadvantages of the above proof are that it incurs a quadratic loss in efficiency (from the closure under intersection), and it relies on the somewhat-heavy machinery of [BRS95]. We provide a direct proof that overcomes both of these disadvantages.

Since the acceptance probability on any input can be expressed as the area of the union of rectangles within the block, it can also be expressed in terms of the areas of *intersections* of rectangles using the inclusion–exclusion formula. For 1-inputs, there are no nonempty intersections of two or more rectangles, so the formula can safely be truncated. For 0-inputs, truncating the formula at an *even* level (say, the second) gives an underestimate of the acceptance probability, which is fine for our purpose. Then we can use standard techniques to construct a PP protocol whose acceptance probability is related to the value of the truncated inclusion-exclusion formula. This argument automatically handles AM protocols with "bounded ambiguity" on 1-inputs, simply by truncating the inclusion-exclusion formula at an appropriate level.

3.5 Theorem 5

Our approach to lower bound the SBP decision tree complexity of Gut is analogous to a corruption-style argument in communication complexity. We consider a hard pair of distributions, one over 1-inputs and the other over 0-inputs, and we argue that for any root-to-leaf path in a deterministic decision tree, the path's acceptance probability under the 1-input distribution cannot be more than a small constant factor greater than its acceptance probability under the 0-input distribution. Arguing the latter is more-or-less a direct, technical calculation. For the corruption analogy, a root-to-leaf path in a decision tree plays the role of a transcript/rectangle in a communication protocol.

4 ZAM Upper Bounds

In this section we prove Theorem 1 as well as the upper bound $\mathsf{ZAM}(\textsc{Eq}) \leq O(\log n)$.

4.1 Existence and universal upper bound

We first show that every two-party function f has a ZAM protocol.

Theorem 6. *$\mathsf{ZAM}(f) \leq O(2^{\mathsf{coNP}(f)})$ for all f.*

Proof. It is a basic fact that any function f reduces to the set-disjointness function $\textsc{Disj}_n$ on $n = 2^{\mathsf{coNP}(f)}$ bits; see [KN97, Example 4.45]. Thus, to prove the theorem, it suffices to show that $\textsc{Disj}_n$ admits a ZAM protocol of communication cost $O(n)$.

By definition, $\textsc{Disj}_n(x, y) = 1$ iff $\textsc{nand}(x_i, y_i) = 1$ for all $i \in [n]$. Thus, to verify that $\textsc{Disj}_n(x, y) = 1$, we can simply run n independent instances of the NAND protocol from Figure 1 in parallel (one for each coordinate i) and require that all of them accept. In more detail: Alice and Bob each flip n coins, and Merlin's proofs are labeled by strings from $\{a, b, c, d\}^n$. For a particular label string ℓ, the associated rectangle τ_ℓ is the intersection of the following n rectangles: the $\ell_i \in \{a, b, c, d\}$ rectangle from the NAND protocol applied to the i-th bits of the input and the i-th coin tosses, for all i.

All the claimed properties are straightforward to verify. On any 1-input, we claim that there is a bijection between Merlin's proofs and the outcomes of all the coin tosses, which immediately implies that the protocol has perfect completeness and is unambiguous and zero-information. For a 1-input of a single instance of NAND, by inspection there is a bijection between the four proof labels $\{a, b, c, d\}$ and the four outcomes of coin tosses. A 1-input of $\textsc{Disj}_n$ is a sequence of n 1-inputs for NAND, and the cartesian product of the n associated bijections yields the bijection for the whole input. The protocol has soundness $1/2$ since for any 0-input there is a coordinate i that is a 0-input for NAND, and if the i-th coin tosses land unfavorably to Merlin (which happens with probability $1/2$) then the outcome is not covered by any rectangle (since we take intersections of rectangles). The protocol has cost $\log(4^n) = 2n$. □

4.2 Upper bounds from determinant

Our main source for efficient zero-information protocols builds on a zero-information method for testing whether a given matrix M (of a suitable form) has a nonzero determinant.

Given an input x to Alice and y to Bob, let $M = M(x, y)$ be a matrix with entries from some finite field $\mathbb{F}$. We say that M is a *two-party* matrix if each row of M is "owned" by either Alice or Bob, that is, for each row either all its entries are functions of x or all its entries are functions of y. For example, if $x, y \in \{0, 1\}$ are just single bits, then

$$M = \begin{bmatrix} 1 & x \\ y & 1 \end{bmatrix} \tag{1}$$

is a two-party matrix; Alice owns the first row, and Bob owns the second.

Lemma 7. *Let $M = M(x, y)$ be a two-party $n \times n$ matrix. There is a perfect-completeness* ZAM *protocol of cost $\lceil n \cdot \log |\mathbb{F}| \rceil$ for verifying that $\det_{\mathbb{F}}(M) \neq 0$.*

Proof. The idea is to use the fact that the linear map $M\colon \mathbb{F}^n \to \mathbb{F}^n$ is a bijection iff $\det_{\mathbb{F}}(M) \neq 0$. Let $R_A \cup R_B = [n]$ be a partition of the rows of M into those owned by Alice and those owned by Bob. The protocol is as follows.

Protocol for $\det(M) \neq 0$.

1. Alice and Bob start by generating a uniformly random vector $\boldsymbol{v} \in \mathbb{F}^n$, by choosing the values v_i for $i \in R_A$ using Alice's private coins, and choosing the values v_i for $i \in R_B$ using Bob's private coins. The vector $\boldsymbol{v}$ is the challenge sent to Merlin.
2. Merlin's task is to provide a preimage for $\boldsymbol{v}$ under M. That is, Merlin's proof is an encoding of a vector $\boldsymbol{u} \in \mathbb{F}^n$ that Merlin claims satisfies $M\boldsymbol{u} = \boldsymbol{v}$.
3. To check Merlin's claim, Alice computes $(M\boldsymbol{u})_i$ for the coordinates $i \in R_A$ and accepts iff $(M\boldsymbol{u})_i = v_i$ for all $i \in R_A$. Bob does the same for the coordinates in R_B.

If $\det_{\mathbb{F}}(M) \neq 0$ then M has full rank, and so for every outcome of the randomness $\boldsymbol{v}$ there is exactly one proof $\boldsymbol{u} = M^{-1}\boldsymbol{v}$ that Alice and Bob would accept, and for every proof $\boldsymbol{u}$ there is exactly one outcome of the randomness $\boldsymbol{v} = M\boldsymbol{u}$ for which Alice and Bob would accept $\boldsymbol{u}$. The existence of this bijection immediately implies that the protocol has perfect completeness and is unambiguous and zero-information. For soundness, note that if $\det_{\mathbb{F}}(M) = 0$ then the range of M is a proper subspace of $\mathbb{F}^n$, and so with probability at least $1 - 1/|\mathbb{F}| \geq 1/2$ the challenge vector $\boldsymbol{v}$ lies outside of this subspace. In that case there is no proof that would make Alice and Bob accept. □

We can now start designing ZAM protocols for different two-party functions by reducing them to the evaluation of a determinant. For example, for the matrix in (1) we have

$$\det_{\mathbb{F}}(M) = 1 - xy = \mathrm{NAND}(x, y).$$

Thus, we get a family of ZAM protocols for NAND parametrized by the choice of $\mathbb{F}$. In fact, for $|\mathbb{F}| = 2$ we recover the protocol from Figure 1.

More generally, we can exploit a known connection between the determinant and branching programs. We describe the connection only for $|\mathbb{F}| = 2$, in which case we are dealing with *parity branching programs*. The standard definition of parity branching programs [Juk12, §1.3] is somewhat less general than the definition we use: we allow each edge to "query" an arbitrary predicate of either Alice's input or Bob's input, as opposed to just querying a single bit.

Definition 1. A *two-party parity branching program* ($\oplus$BP) is a directed acyclic graph (G, s, t) with a source node s and a target node t, such that each edge is owned by Alice and labeled with a function $\mathcal{X} \to \{0,1\}$, or is owned by Bob and labeled with a function $\mathcal{Y} \to \{0,1\}$. Each input $(x, y) \in \mathcal{X} \times \mathcal{Y}$ induces an unlabeled graph G_{xy} by evaluating all the label functions on their associated parts of the input (Alice edges on x and Bob edges on y), and keeping the edges that evaluate to 1 and deleting the edges that evaluate to 0. The branching program computes $f\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ iff for all inputs (x, y), the number of s-t paths in G_{xy} is odd iff $f(x, y) = 1$. We define $\oplus\mathsf{BP}(f)$ to be the minimum order (number of nodes) of any two-party parity branching program computing f.

We can now prove Theorem 1, restated here for convenience.

Theorem 1. $\mathsf{ZAM}(f) \leq O(\oplus\mathsf{BP}(f))$ *for all* f.

Proof. Let (G, s, t) be a minimum-order $\oplus\mathsf{BP}$ for f. We may assume (with at most a factor 2 increase in the order of G) that each node of G is owned by either Alice or Bob, in the sense that the outgoing edges are either all owned by Alice or all owned by Bob. Fix an input (x, y). We modify G_{xy} to obtain a graph G^+_{xy} by adding a directed edge from t to s and adding a self-loop to each node except s and t. Let $M = M(x, y)$ be the adjacency matrix of G^+_{xy}. The s-t paths of G_{xy} are in one-to-one correspondence with the cycle covers of G^+_{xy} (as in [Val79]). But now the mod-2 determinant of M is exactly computing the parity of the number of cycle covers of G^+_{xy}, which equals the parity of the number of s-t paths in G_{xy}, which equals $f(x, y)$. Since each node of G is owned by Alice or Bob, M is a two-party matrix and thus we can use Lemma 7 to evaluate its mod-2 determinant. The cost of the protocol is the order of G^+_{xy}, which is at most $2 \cdot \oplus\mathsf{BP}(f)$. □

4.3 Upper bound for equality

The above determinant approach does not really yield interesting examples of ZAM protocols with $o(n)$ communication. Here we note that such a protocol exists at least for the equality function.

Theorem 8. $\mathsf{ZAM}(\mathrm{EQ}) \le O(\log n)$.

Proof. Let $C\colon \{0,1\}^n \to \{0,1\}^N$ be an error-correcting code with $N = \Theta(n)$ and constant relative minimum distance $\delta > 0$. Consider the following protocol for EQ_n on input (x, y): Alice chooses $i \in [N]$ uniformly at random, and then Alice and Bob run some constant cost ZAM protocol for EQ_1 (e.g., adapting the XOR protocol from Figure 1) to check that $C(x)_i = C(y)_i$. This way Merlin's message consists of i together with a label from the EQ_1 protocol. The perfect completeness, unambiguity, and zero-information properties are inherited from the EQ_1 protocol. The soundness is $1 - \delta/2$ and this can be brought down by repeating the protocol constantly many times. □

Remark 2. We can achieve an upper bound of $\log n + O(1)$ in Theorem 8 for any positive constant soundness by using a code where each coordinate is k bits long (so the alphabet has size 2^k) and running the EQ_1 protocol ℓ times for each of the k bits in the i-th coordinate. Choosing k and ℓ to be large enough constants depending on the desired soundness gives a communication bound of $\lceil \log N \rceil + O(k\ell) \le \log n + O(1)$.

5 ZAM Lower Bounds

In this section we prove Theorem 2, restated here for convenience.

Theorem 2. $\mathsf{ZAM}(f) \ge \Omega(\mathsf{coNP}(f))$ *for all* f.

Henceforth we fix a ZAM protocol Π computing a function $f\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ with randomness sets $\mathcal{R}, \mathcal{Q}$; the completeness need not be perfect. The following is a central definition.

Definition 2. The *(projected) double cover* of Π is the collection of all rectangles of the form

$$R_{\tau,\tau'} = \big\{(x, y) \,:\, ((x, r), (y, q)) \in \tau \cap \tau' \text{ for some } r \in \mathcal{R}, q \in \mathcal{Q}\big\}$$

where $\tau \neq \tau'$ are proof rectangles of Π. In words, $R_{\tau,\tau'}$ consists of all inputs for which τ and τ' overlap inside the block of that input. Note that $R_{\tau,\tau'}$ is indeed a rectangle.

A key observation is that since Π is unambiguous, the double cover only covers 0-inputs of f. If we knew that the double cover covered *all* the 0-inputs of f, then the double cover would be a conondeterministic protocol and would hence have cost at least $\mathsf{coNP}(f)$. In this case, we would be done since the number of proof rectangles of Π is greater than the square root of the number of rectangles in the double cover, so the communication cost of Π would be at least $\mathsf{coNP}(f)/2$. Unfortunately, this assumption does not always hold: Figure 3 shows an example of a ZAM protocol (for XOR) where the block of some 0-input has no overlapping rectangles.

The outline of our proof is as follows.

(1) We identify a sufficient condition for a 0-input to be covered by the double cover, and thus we show that the double cover makes useful progress toward covering $f^{-1}(0)$.
(2) We patch up the double cover to obtain a conondeterministic protocol for f, by introducing rectangles to cover the remaining 0-inputs.
(3) We analyze the patched-up protocol's communication cost (which is at least $\mathsf{coNP}(f)$) to show that it is at most a constant factor larger than the cost of Π.

Step (1): Finding intersections. We define the *connectivity graph* G_f of f as follows: the vertex set is $f^{-1}(1)$ and two 1-inputs are adjacent iff they share the same x-value or the same y-value (i.e., they lie on the same row or the same column of the communication matrix of f). If C is a connected component of G_f, we say that a 0-input (x, y) is *surrounded* by C if there exist x' and y' such that $(x, y'), (x', y) \in C$ (i.e., C hits both the row and the column of (x, y)). The following lemma is our sufficient condition for step (1).

Lemma 9. *Every 0-input surrounded by a connected component is covered by the double cover of* Π.

We work toward the proof of Lemma 9. Let $\pi_{xy}(\tau)$ be the probability that Π accepts the proof τ on input (x, y). Because τ is a rectangle, we can write $\pi_{xy}(\tau) = a_x(\tau) \cdot b_y(\tau)$ where a_x and b_y are some functions known to Alice and Bob, respectively. We define the *shape* of the proof τ inside (x, y) as the pair $(a_x(\tau), b_y(\tau))$.

The zero-information property says that $\pi_{xy}(\tau)$ is the same for all 1-inputs (x, y) (and without loss of generality we assume this common probability is positive; otherwise τ could be eliminated from Π). In fact, more is true:

Claim 10. *If* $(x, y), (x', y') \in f^{-1}(1)$ *belong to the same connected component of* G_f*, then any proof* τ *has the same shape inside* (x, y) *as inside* (x', y').

Proof. Suppose first that (x, y) and (x', y') lie on the same row so that $x = x'$. Then $a_x(\tau) = a_{x'}(\tau)$, and this quantity is positive since $\pi_{xy}(\tau)$ is positive. But now from $a_x(\tau) \cdot b_y(\tau) = \pi_{xy}(\tau) = \pi_{x'y'}(\tau) = a_{x'}(\tau) \cdot b_{y'}(\tau)$ it follows that $b_y(\tau) = b_{y'}(\tau)$ and we are done in this case. The argument is analogous if (x, y) and (x', y') lie on the same column. More generally, the claim follows by induction along a path from (x, y) to (x', y'). □

The statement of Claim 10 does not necessarily hold when (x, y) and (x', y') are not in the same connected component of G_f. Indeed, in the example of Figure 3 the two 1-inputs are in different connected components, and inside their blocks the proofs have different shapes.

Proof of Lemma 9. Let $(x, y) \in f^{-1}(0)$ be surrounded by $(x, y'), (x', y) \in C$. By Claim 10 the shape of any proof τ is the same in (x, y') as in (x', y). But this implies that the shape of τ in (x, y) also

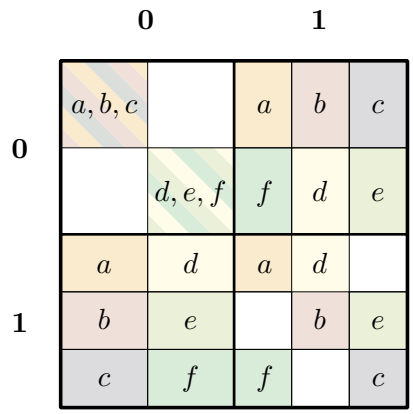


Figure 3: XOR example: The block of the 0-input $(1, 1)$ has no overlapping rectangles.

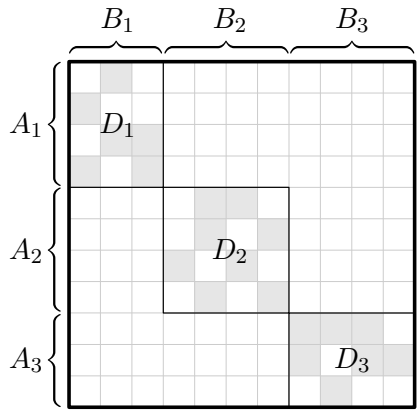


Figure 4: Decomposing f along connected components. Here 1-inputs are shaded in gray.

matches its shapes in (x, y') and in (x', y). In particular $\pi_{xy}(\tau) = \pi_{xy'}(\tau)$ so that

$$\sum_{\tau} \pi_{xy}(\tau) = \sum_{\tau} \pi_{xy'}(\tau).$$

Because Π is unambiguous on 1-inputs, the sum on the right calculates the acceptance probability of Π on (x, y'), which is at least the completeness parameter. If all the proofs inside (x, y) were pairwise disjoint, then the sum on the left would calculate the acceptance probability of Π on (x, y), which is at most the soundness parameter. To avoid a contradiction, we must have that some pair of proofs intersect inside the (x, y) block. □

Step (2): Patching up the double cover. Let $C_1, \ldots, C_m$ be the connected components of G_f. We decompose the communication matrix of f as follows; see Figure 4. Define A_i to be the projection of C_i onto the first coordinate (Alice's set of inputs $\mathcal{X}$). Note that the A_i are pairwise disjoint. Assuming for simplicity that f does not contain all-0 rows, then $A_1 \cup \cdots \cup A_m$ is a partition of $\mathcal{X}$. We can similarly define sets B_i that partition Bob's set of inputs $\mathcal{Y}$. Let $D_i = A_i \times B_i$ and note that each 1-input is now contained in a unique $D_i \supseteq C_i$.

Each input in $D_i \setminus C_i$ is a 0-input surrounded by C_i and is hence covered by the double cover by Lemma 9. The inputs not in any D_i are all 0-inputs and they form a "non-equality-like" structure, which can be covered by applying the standard conondeterministic protocol for equality to this structure. Here is the formal description of our conondeterministic protocol Γ for f.

Protocol Γ. On input (x, y) let $i, j \in [m]$ be such that $x \in A_i$ and $y \in B_j$, and guess one of the following two cases:

1. Guess a pair of distinct proof rectangles τ, τ' and check that $(x, y) \in R_{\tau,\tau'}$.
2. Guess an index $\ell \in [\log m]$ and a bit b, and check that when i and j are written in binary, the ℓ-th bit of i is b and the ℓ-th bit of j is $1 - b$.

To formally verify the correctness of Γ, we need to observe that it covers *all* 0-inputs and covers *only* 0-inputs. For an arbitrary 0-input (x, y), if $i = j$ then (x, y) is surrounded by C_i and is hence

covered by case 1 by Lemma 9, and if $i \neq j$ then (x, y) is covered by case 2. The double cover rectangles of case 1 cover only 0-inputs by unambiguity, and the "non-equality" rectangles of case 2 cover only 0-inputs since each 1-input is in some $C_i \subseteq D_i$ and thus has $i = j$.

Step (3): Analyzing the cost. The communication cost of case 1 is at most twice the cost of Π, and the communication cost of case 2 is $O(\log \log m)$. That is, in symbols,

$$|\Gamma| \leq 2 \cdot |\Pi| + O(\log \log m). \tag{2}$$

To simplify this estimate, we note that if G_f has m connected components, then f has a fooling set of size m: simply pick one 1-input from each connected component. This implies that $\mathsf{P}(f) \geq \log m$ (where $\mathsf{P}(f)$ denotes the deterministic communication complexity of f). We prove later in Section 7 (as Corollary 19) that $\mathsf{UAM}(f) \geq \Omega(\log \mathsf{P}(f))$. Using this we deduce that

$$|\Pi| \geq \mathsf{ZAM}(f) \geq \mathsf{UAM}(f) \geq \Omega(\log \mathsf{P}(f)) \geq \Omega(\log \log m).$$

Thus (2) can in fact be written as $|\Gamma| \leq O(|\Pi|)$, so we get $|\Pi| \geq \Omega(|\Gamma|) \geq \Omega(\mathsf{coNP}(f))$ as desired. This proves Theorem 2.

6 UAM Lower Bound for Set-Intersection

In this section we prove Theorem 3, restated here for convenience.

Theorem 3. $\mathsf{UAM}(\textsc{Inter}) = \Theta(n)$.

In Section 6.1 we prove the lower bound $\mathsf{UAM}(\textsc{Inter}) \geq \Omega(n)$ assuming two lemmas, which we prove in Section 6.2 and Section 6.3. Throughout this section, we adhere to the convention that capital letters denote random variables, and bold letters denote tuples. We write $\|X - Y\|$ for the total variation distance between the distributions of random variables X and Y. Also, when we write $\leq o(1)$ we formally mean a quantity that is upper bounded by some sufficiently small positive constant, which may be different for different instances of $o(1)$.

6.1 Deriving the lower bound

We make use of tools from information complexity, and for this we need to define a distribution over inputs X, Y with respect to which information cost is measured. As in the standard approach [BYJKS04, JKS03], we need to define an additional random variable jointly distributed with the input, which is used to "condition" the input distribution. However, in contrast to the standard approach, we employ a *two-stage* conditioning scheme involving two jointly distributed variables V and W. With some foresight, the key benefit of this is what we call the *disjoint windows property*, formalized in Proposition 13 below.

Distribution for $\mathrm{AND}^{-1}(0)$. Fix a small constant $\delta > 0$ (to be determined later). We define four jointly distributed random variables (V, W, X, Y); see Figure 5. First let

$$V = \begin{cases} \mathsf{edge\text{-}A} & \text{with probability } \delta/2, \\ \mathsf{edge\text{-}B} & \text{with probability } \delta/2, \\ \mathsf{singleton} & \text{with probability } 1 - \delta. \end{cases}$$

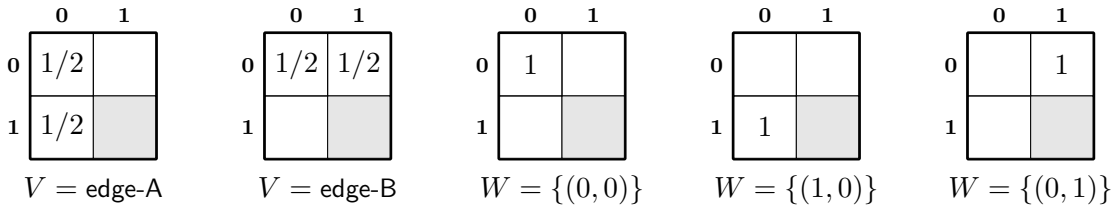


Figure 5: Distribution of $(X, Y) \in \mathrm{AND}^{-1}(0)$ conditioned on an outcome of (V, W).

Conditioned on an outcome of V, the variable $W \subseteq \mathrm{AND}^{-1}(0)$ is defined by

$$\begin{array}{ll} \text{if } V = \mathsf{edge\text{-}A} & \text{then } W = \{(0,0),(1,0)\}, \\ \text{if } V = \mathsf{edge\text{-}B} & \text{then } W = \{(0,0),(0,1)\}, \\ \text{if } V = \mathsf{singleton} & \text{then } W \in_u \big\{\{(0,0)\},\{(1,0)\},\{(0,1)\}\big\}. \end{array}$$

Here edge-A, edge-B, and singleton are arbitrary syntactic labels, and $\in_u$ means "is uniform in". Note that the outcome of V is fully determined by the outcome of W. Also note that the marginal distribution of W places probability $(1-\delta)/3$ on each of the three outcomes $\{(0,0)\}$, $\{(1,0)\}$, and $\{(0,1)\}$, and probability $\delta/2$ on each of the two outcomes $\{(0,0),(1,0)\}$ and $\{(0,0),(0,1)\}$. Finally, let

$$(X, Y) \in_u W.$$

Distribution for $\textsc{Inter}^{-1}(1)$. We define jointly distributed random variables $(I, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y})$. Let $I \in_u [n]$, and conditioned on $I = i$, let $(X_i, Y_i) = (1,1)$ and let the variables $(V_j, W_j, X_j, Y_j) : j \in [n] \setminus \{i\}$ be mutually independent and each distributed identically to (V, W, X, Y) as above. Note that $\boldsymbol{V}$ and $\boldsymbol{W}$ are $(n-1)$-tuples that we view as indexed by $[n] \setminus \{I\}$. We let $\boldsymbol{X}_{-I} = (X_1, \ldots, X_{I-1}, X_{I+1}, \ldots, X_n)$ and similarly for $\boldsymbol{Y}_{-I}$.

Consider a UAM protocol for $\textsc{Inter}$ with completeness some constant sufficiently close to 1 (i.e., $1 - o(1)$) and soundness some constant sufficiently less than 1. Let $\Pi = \Pi(\boldsymbol{X}, \boldsymbol{Y})$ be the random variable denoting the unique proof accepted by the protocol on the random input $(\boldsymbol{X}, \boldsymbol{Y})$ or $\perp$ if no such proof exists. We consider the *partial information cost* $\mathrm{PIC}(\Pi) = \mathbb{I}(\Pi : \boldsymbol{X}_{-I}, \boldsymbol{Y}_{-I} \mid I, \boldsymbol{W})$ as introduced in [JKS03]. If $\mathrm{PIC}(\Pi) \geq \Omega(n)$ then we are done since the worst-case proof length is at least $\mathbb{H}(\Pi \mid I, \boldsymbol{W}) \geq \mathrm{PIC}(\Pi)$, so assume $\mathrm{PIC}(\Pi) \leq o(n)$. A direct sum property proven in [JKS03] states that $\mathrm{PIC}(\Pi) \geq (n-1) \cdot \mathbb{E}_{i \neq j} \mathbb{I}(\Pi : X_j, Y_j \mid I = i, \boldsymbol{W})$ where the expectation is uniformly over the $n(n-1)$ possibilities for i and $j \neq i$. Thus there is a *non-leaky* pair of coordinates, which we henceforth assume is $\{i, j\} = \{1, 2\}$ (without specifying which is i and which is j), satisfying $\mathbb{E}_{\{i,j\}=\{1,2\}} \mathbb{I}(\Pi : X_j, Y_j \mid I = i, \boldsymbol{W}) \leq o(1)$ where the expectation is uniformly over the two possibilities $(i, j) = (1, 2)$ and $(i, j) = (2, 1)$. For notational convenience, let us change the tuples $\boldsymbol{V}$ and $\boldsymbol{W}$ so that they are indexed by $[n] \setminus \{1, 2\}$ and thus they no longer include the random variables V_j and W_j. With this notation, we have

$$\mathbb{E}_{\{i,j\}=\{1,2\}} \mathbb{I}(\Pi : X_j, Y_j \mid I = i, W_j, \boldsymbol{W}) \leq o(1). \tag{3}$$

Definition 3 (Non-leaky conditioning)**.** We say that an outcome $\boldsymbol{w}$ is *non-leaky* if

$$\mathbb{E}_{\{i,j\}=\{1,2\}} \mathbb{I}(\Pi : X_j, Y_j \mid I = i, W_j, \boldsymbol{W} = \boldsymbol{w}) \leq o(1).$$

Definition 4 (Windows)**.** We define the *window* of an outcome $\boldsymbol{w} = (w_3, \dots, w_n)$ to be the set of inputs $\{0,1\}^2 \times \{0,1\}^2 \times w_3 \times \cdots \times w_n$, which is a rectangle since each w_k is a rectangle. (As a technicality, we are writing inputs as $x_1y_1, x_2y_2, \dots, x_ny_n$ here, rather than $x_1x_2 \dots x_n, y_1y_2 \dots y_n$.)

Definition 5 (Double cover)**.** Recall from Section 5 that the *double cover* of the protocol Π is the collection of all pairwise intersections of Merlin's rectangles, "projected" to the inputs (i.e., an input is in the projected pairwise intersection of two rectangles iff the pairwise intersection contains a point in the block of that input).

Lemma 11. *For some constant $C > 2$ there exists a set $\mathcal{W}$ of $\Omega(C^n)$ many non-leaky $\boldsymbol{w}$'s with pairwise disjoint windows.*

Lemma 12. *For every non-leaky $\boldsymbol{w}$ there exists an input in $\boldsymbol{w}$'s window that is contained in a rectangle of the double cover.*

We prove Lemma 11 in Section 6.2, and we prove Lemma 12 in Section 6.3. First we see how to use these two lemmas to finish the proof of Theorem 3. For each $\boldsymbol{w} \in \mathcal{W}$, fix an associated input given by Lemma 12. These associated inputs are all distinct (by disjointness of the windows), so there are $\Omega(C^n)$ many of them, and they are all covered by the double cover. By unambiguity, each rectangle of the double cover is contained in $\textsc{Inter}^{-1}(0)$ and thus has size at most 2^n. Therefore the double cover must have $\Omega(C^n)/2^n \geq \exp(\Omega(n))$ many rectangles in order to cover all the associated inputs for $\boldsymbol{w} \in \mathcal{W}$. Thus the protocol has at least $\sqrt{\exp(\Omega(n))} \geq \exp(\Omega(n))$ many proof rectangles, which means the worst-case proof length is $\Omega(n)$. This finishes the proof of Theorem 3.

6.2 Finding many non-leaky disjoint windows

We now prove Lemma 11. Applying Markov's inequality to (3) tells us there are many non-leaky $\boldsymbol{w}$'s but does not help us find $\boldsymbol{w}$'s with disjoint windows. For the latter, we observe the following key property of our two-stage conditioning scheme.

Proposition 13 (Disjoint windows property)**.** *Let $\boldsymbol{v}$ be any outcome of $\boldsymbol{V}$. The windows corresponding to the possible outcomes of $\boldsymbol{W} \mid \boldsymbol{V} = \boldsymbol{v}$ are pairwise disjoint.*

Proof. Two different outcomes $\boldsymbol{w}$ consistent with $\boldsymbol{v}$ must differ in a coordinate $k \in [n] \setminus \{1,2\}$ on which $v_k = \mathsf{singleton}$, e.g., one having $w_k = \{(1,0)\}$ and the other having $w_k = \{(0,1)\}$—but then all inputs in the former outcome's window would have $(x_k, y_k) = (1,0)$ and all inputs in the latter outcome's window would have $(x_k, y_k) = (0,1)$, so the windows are disjoint. □

We claim that there exists a vector $\boldsymbol{v}$ (indexed by $[n] \setminus \{1,2\}$) with the following two properties.

- **(P1)** There are at at least $3^{(1-2\delta)(n-2)}$ many $\boldsymbol{w}$'s consistent with $\boldsymbol{v}$.
- **(P2)** At least half of the $\boldsymbol{w}$'s consistent with $\boldsymbol{v}$ are non-leaky.

Combining (P1) and (P2) tells us there are at least $\frac{1}{2} \cdot 3^{(1-2\delta)(n-2)}$ many non-leaky $\boldsymbol{w}$'s consistent with $\boldsymbol{v}$, and Proposition 13 tells us their windows are pairwise disjoint. This proves the lemma with $C = 3^{1-2\delta}$, which is greater than 2 provided δ is small enough, say $\delta = 1/8$. We show the existence of $\boldsymbol{v}$ by arguing that each of (P1) and (P2) individually holds with high probability over $\boldsymbol{v}$.

(P1) is equivalent to having $v_k = \mathsf{singleton}$ for at least $(1-2\delta)(n-2)$ many $k \in [n] \setminus \{1,2\}$. The expected number of such k's is $(1-\delta)(n-2)$, so (P1) holds with high probability by a concentration

bound (e.g., the variance of the number of such k's is $O(n)$ and so Chebyshev's inequality is good enough).

For (P2), note that applying Markov's inequality to (3) shows that with high probability over $\boldsymbol{v}$, $\mathbb{E}_{\{i,j\}=\{1,2\}} \mathbb{I}(\Pi : X_j, Y_j \mid I = i, W_j, \boldsymbol{V} = \boldsymbol{v}, \boldsymbol{W}) \leq o(1)$ holds. In this event, since the distribution of $\boldsymbol{W} \mid \boldsymbol{V} = \boldsymbol{v}$ is uniform over all $\boldsymbol{w}$ consistent with $\boldsymbol{v}$, another application of Markov's inequality shows that for at least half of the $\boldsymbol{w}$'s consistent with $\boldsymbol{v}$, $\mathbb{E}_{\{i,j\}=\{1,2\}} \mathbb{I}(\Pi : X_j, Y_j \mid I = i, W_j, \boldsymbol{V} = \boldsymbol{v}, \boldsymbol{W} = \boldsymbol{w}) \leq o(1)$ holds. In the latter case, $\boldsymbol{w}$ is non-leaky since the event "$\boldsymbol{V} = \boldsymbol{v}, \boldsymbol{W} = \boldsymbol{w}$" is the same as the event $\boldsymbol{W} = \boldsymbol{w}$. This finishes the proof of Lemma 11.

6.3 Small information leakage yields conondeterminism

We now prove Lemma 12. Fix a non-leaky $\boldsymbol{w}$. For conceptual clarity, we associate to $\boldsymbol{w}$ a corresponding subprotocol $\Pi_{\boldsymbol{w}}$ that is the restriction of Π to $\boldsymbol{w}$'s window and is a UAM protocol for $\textsc{Inter}_2$; see Figure 6a. For notational convenience, let us now change the tuples $\boldsymbol{X}$ and $\boldsymbol{Y}$ so that they are indexed by $[n] \setminus \{1, 2\}$ and thus they no longer include the random variables X_1, X_2 and Y_1, Y_2.

Protocol $\boldsymbol{\Pi_w}$. On input (x_1x_2, y_1y_2):

1. Using private coins, Alice samples $\boldsymbol{x}$ and Bob samples $\boldsymbol{y}$ from $(\boldsymbol{X}, \boldsymbol{Y}) \mid \boldsymbol{W} = \boldsymbol{w}$.
2. Alice and Bob simulate Π on input $(x_1x_2\boldsymbol{x}, y_1y_2\boldsymbol{y})$.

Note that $\boldsymbol{w}$'s window is naturally partitioned into 16 "panes" according to the first two coordinates of the input, and these panes correspond to the "blocks" for the 16 possible inputs to $\Pi_{\boldsymbol{w}}$. Since all w_k's are rectangles, the (x_1x_2, y_1y_2) pane of $\boldsymbol{w}$'s window is a rectangle and $(x_1x_2\boldsymbol{X}, y_1y_2\boldsymbol{Y}) \mid \boldsymbol{W} = \boldsymbol{w}$ is uniformly distributed in this pane, and hence the sampling on line 1 can indeed be done with private coins and no communication. Since all w_k's are contained in $\mathrm{AND}^{-1}(0)$, we have $\textsc{Inter}_n(x_1x_2\boldsymbol{x}, y_1y_2\boldsymbol{y}) = \textsc{Inter}_2(x_1x_2, y_1y_2)$ for all inputs in the window, and hence $\Pi_{\boldsymbol{w}}$ is indeed a UAM protocol for $\textsc{Inter}_2$. Since $\boldsymbol{w}$ is non-leaky and the transcript $\Pi_{\boldsymbol{w}}(x_1x_2, y_1y_2)$ is distributed the same as $\Pi(x_1x_2\boldsymbol{X}, y_1y_2\boldsymbol{Y}) \mid \boldsymbol{W} = \boldsymbol{w}$, we have

$$\mathbb{E}_{\{i,j\}=\{1,2\}} \mathbb{I}(\Pi_{\boldsymbol{w}} : X_j, Y_j \mid I = i, W_j) \leq o(1). \tag{4}$$

Claim 14. *If (4) holds then*

$$\|\Pi_{\boldsymbol{w}}(z) - \Pi_{\boldsymbol{w}}(z')\| \leq o(1) \text{ for all } \{z, z'\} = \{(10,10),(11,10)\},\ \{(10,10),(10,11)\},\ \{(01,01),(11,01)\},\ \{(01,01),(01,11)\}. \tag{5}$$

Claim 15. *If (5) holds then*

$$\|\Pi_{\boldsymbol{w}}(z) - \Pi_{\boldsymbol{w}}(z')\| \leq o(1) \text{ for all } z, z' \in \textsc{Inter}_2^{-1}(1). \tag{6}$$

Claim 16. *If (6) holds then $\Pi_{\boldsymbol{w}}$ has a pair of proof rectangles that intersect.*

Lemma 12 follows immediately by stringing together (4), Claim 14, Claim 15, and Claim 16, and observing that $\Pi_{\boldsymbol{w}}$ having a pair of proof rectangles that intersect is equivalent to Π having a pair of proof rectangles that intersect within $\boldsymbol{w}$'s window. Claim 14 is a fairly standard calculation, and when combined with Claim 15 this shows that if a protocol has low partial information cost, then it

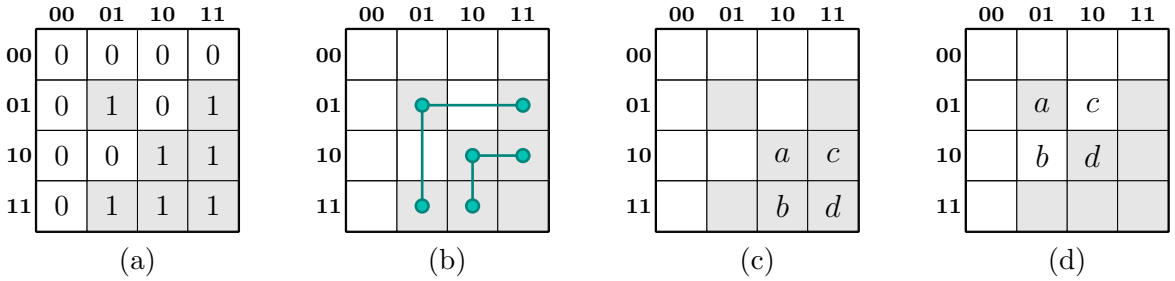


Figure 6: (a) Truth table of $\textsc{Inter}_2$. (b) Four possibilities for $\{z, z'\}$ in Claim 14. (c,d) Inputs a, b, c, d considered in the proofs of Claims 15 and 16.

is close in statistical distance to being zero-information. In Section 5 we saw that zero-information protocols must create intersecting proof rectangles inside the blocks for 0-inputs. In a similar spirit, Claim 16 shows that the close-to-zero-information subprotocol $\Pi_{\boldsymbol{w}}$ must do the same.

We need the following general and widely used lemma; see [BYJKS04, Lemma 6.2] and [Lin91].

Lemma 17 (Information vs. statistical distance)**.** *Let $Z \in_u \{1,2\}$ be jointly distributed with a random variable Ψ. Then $\mathbb{I}(\Psi : Z) \geq \|\Psi_1 - \Psi_2\|^2/2$ where $\Psi_z = (\Psi \mid Z = z)$ for $z \in \{1,2\}$.* □

Proof of Claim 14. The four possibilities for $\{z, z'\}$ correspond to the four edges illustrated in Figure 6b, and they arise from the four different ways of choosing (i,j) and $v_j \in \{\mathsf{edge\text{-}A}, \mathsf{edge\text{-}B}\}$. For example, $\{z, z'\} = \{(10,10),(11,10)\}$ corresponds to $(i,j) = (1,2)$ and $v_2 = \mathsf{edge\text{-}A}$, and by symmetry we may restrict our attention to this possibility. By the definition of conditional mutual information, we have

$$\begin{aligned}\mathbb{I}(\Pi_{\boldsymbol{w}} : X_2, Y_2 \mid I = 1, W_2) = {} & (\delta/2) \cdot \mathbb{I}(\Pi_{\boldsymbol{w}} : X_2, Y_2 \mid I = 1, V_2 = \mathsf{edge\text{-}A}) + \\ & (\delta/2) \cdot \mathbb{I}(\Pi_{\boldsymbol{w}} : X_2, Y_2 \mid I = 1, V_2 = \mathsf{edge\text{-}B}) + \\ & (1-\delta) \cdot \mathbb{I}(\Pi_{\boldsymbol{w}} : X_2, Y_2 \mid I = 1, V_2 = \mathsf{singleton}, W_2).\end{aligned} \tag{7}$$

The second and third summands are nonnegative (in fact, the third is 0 since each outcome of $W_2 \mid V_2 = \mathsf{singleton}$ leaves (X_2, Y_2) constant). In the first summand (X_2, Y_2) is uniformly distributed on two distinct values, so we can apply Lemma 17 to get $\mathbb{I}(\Pi_{\boldsymbol{w}} : X_2, Y_2 \mid I = 1, V_2 = \mathsf{edge\text{-}A}) \geq \|\Pi_{\boldsymbol{w}}(10,10) - \Pi_{\boldsymbol{w}}(11,10)\|^2/2$. Now (7) becomes

$$\mathbb{I}(\Pi_{\boldsymbol{w}} : X_2, Y_2 \mid I = 1, W_2) \geq (\delta/4) \cdot \|\Pi_{\boldsymbol{w}}(10,10) - \Pi_{\boldsymbol{w}}(11,10)\|^2.$$

Finally, from (4) we get that

$$\|\Pi_{\boldsymbol{w}}(10,10) - \Pi_{\boldsymbol{w}}(11,10)\| \leq \sqrt{(8/\delta) \cdot \mathbb{E}_{\{i,j\}=\{1,2\}} \mathbb{I}(\Pi_{\boldsymbol{w}} : X_j, Y_j \mid I = i, W_j)} \leq o(1)$$

since δ is a positive constant. □

Proof of Claim 15. We prove the claim for $\{z, z'\} = \{(10,10),(11,11)\}$. Then by symmetry, the claim also holds for $\{z, z'\} = \{(01,01),(11,11)\}$, and the full claim follows by the triangle inequality.

Consider the inputs $a = (10,10)$, $b = (11,10)$, $c = (10,11)$, $d = (11,11)$; see Figure 6c. For any proof τ let $\pi_a(\tau) = \mathbb{P}[\Pi_{\boldsymbol{w}}(a) \text{ accepts } \tau]$, and let $\pi_a(\perp) = \mathbb{P}[\Pi_{\boldsymbol{w}}(a) \text{ accepts no proof}]$, and similarly

for b, c, d. Let $\gamma_{ab}(\tau) = |\pi_a(\tau) - \pi_b(\tau)|$ and $\gamma_{ab}(\bot) = |\pi_a(\bot) - \pi_b(\bot)|$, and similarly for other pairs of inputs. By the rectangular structure of proofs we have $\pi_a(\tau) \cdot \pi_d(\tau) = \pi_b(\tau) \cdot \pi_c(\tau)$, and by a case analysis we have $\pi_b(\tau) \cdot \pi_c(\tau) \geq \pi_a(\tau)^2 - \pi_a(\tau)\gamma_{ab}(\tau) - \pi_a(\tau)\gamma_{ac}(\tau)$. Thus if $\pi_a(\tau) > 0$ then $\pi_d(\tau) = \frac{\pi_b(\tau)\cdot\pi_c(\tau)}{\pi_a(\tau)} \geq \pi_a(\tau) - \gamma_{ab}(\tau) - \gamma_{ac}(\tau)$. Hence if $\pi_a(\tau) > \pi_d(\tau)$ then $\gamma_{ad}(\tau) \leq \gamma_{ab}(\tau) + \gamma_{ac}(\tau)$. Therefore

$$\begin{aligned}
\|\Pi_{\boldsymbol{w}}(a) - \Pi_{\boldsymbol{w}}(d)\| &\leq \gamma_{ad}(\bot) + \textstyle\sum_{\tau\,:\,\pi_a(\tau)>\pi_d(\tau)} \gamma_{ad}(\tau) \\
&\leq o(1) + \textstyle\sum_{\tau\,:\,\pi_a(\tau)>\pi_d(\tau)} \big(\gamma_{ab}(\tau) + \gamma_{ac}(\tau)\big) \\
&\leq o(1) + 2 \cdot \|\Pi_{\boldsymbol{w}}(a) - \Pi_{\boldsymbol{w}}(b)\| + 2 \cdot \|\Pi_{\boldsymbol{w}}(a) - \Pi_{\boldsymbol{w}}(c)\| \\
&\leq o(1) + o(1) + o(1) \\
&\leq o(1)
\end{aligned}$$

where $\gamma_{ad}(\bot) \leq o(1)$ follows by completeness, and the fourth line follows by (5). □

Proof of Claim 16. Consider the inputs $a = (01, 01)$, $b = (10, 01)$, $c = (01, 10)$, $d = (10, 10)$; see Figure 6d (these are *not* the same as in the proof of Claim 15). As before, for any proof τ let $\pi_a(\tau) = \mathbb{P}[\Pi_{\boldsymbol{w}}(a) \text{ accepts } \tau]$ and $\gamma_{ad}(\tau) = |\pi_a(\tau) - \pi_d(\tau)|$, and similarly for other inputs. Assuming that no proof rectangles intersect, it follows that

$$\begin{aligned}
\max\big(\mathbb{P}[\Pi_{\boldsymbol{w}}(b) \text{ accepts}], \mathbb{P}[\Pi_{\boldsymbol{w}}(c) \text{ accepts}]\big) &\geq \tfrac{1}{2}\big(\mathbb{P}[\Pi_{\boldsymbol{w}}(b) \text{ accepts}] + \mathbb{P}[\Pi_{\boldsymbol{w}}(c) \text{ accepts}]\big) \\
&= \tfrac{1}{2} \textstyle\sum_\tau \big(\pi_b(\tau) + \pi_c(\tau)\big) \\
&\geq \tfrac{1}{2} \textstyle\sum_\tau 2\sqrt{\pi_b(\tau) \cdot \pi_c(\tau)} \\
&= \tfrac{1}{2} \textstyle\sum_\tau 2\sqrt{\pi_a(\tau) \cdot \pi_d(\tau)} \\
&\geq \tfrac{1}{2} \textstyle\sum_\tau 2 \cdot \min\big(\pi_a(\tau), \pi_d(\tau)\big) \\
&= \tfrac{1}{2} \textstyle\sum_\tau \big(\pi_a(\tau) + \pi_d(\tau) - \gamma_{ad}(\tau)\big) \\
&\geq \tfrac{1}{2}\big(\mathbb{P}[\Pi_{\boldsymbol{w}}(a) \text{ accepts}] + \mathbb{P}[\Pi_{\boldsymbol{w}}(d) \text{ accepts}]\big) \\
&\quad -\|\Pi_{\boldsymbol{w}}(a) - \Pi_{\boldsymbol{w}}(d)\| \\
&\geq \tfrac{1}{2}\big(1 - o(1) + 1 - o(1)\big) - o(1) \\
&\geq 1 - o(1)
\end{aligned}$$

where the third line follows by the AM–GM inequality, and the second-to-last line follows by completeness and by (6). This contradicts soundness. □

7 Discrepancy Lower Bound for UAM

In this section we prove Theorem 4, restated here for convenience.

Theorem 4. $\mathsf{UAM}(f) \geq \Omega(\mathsf{PP}(f))$ *for all* f.

Our proof actually yields (for free) a more general theorem concerning *bounded-ambiguity* protocols: a k-UAM protocol is defined similarly to a UAM protocol except that now we allow up to k rectangles to overlap on any given point in the block of a 1-input (so UAM = 1-UAM). We note that [KNSW92] and [Kla10] have studied bounded ambiguity in the context of NP and BP·UP, respectively.

Theorem 18. *k-$\mathsf{UAM}(f) \geq \Omega(\mathsf{PP}(f)/k)$ for all f and all k.*

In particular, Theorem 4 has the following corollary, which was used in the proof of Theorem 2 and which can be used to derive $\Omega(\log n)$ lower bounds for all the functions listed in Table 1. Recall that $\mathsf{P}(f)$ stands for the deterministic communication complexity of f.

Corollary 19. *$\mathsf{UAM}(f) \geq \Omega(\log \mathsf{P}(f))$ for all f.*

Strictly speaking, Corollary 19 is not an immediate consequence of Theorem 4 because PP is usually defined as a public-coin model and thus $\mathsf{PP}(\text{EQ}) = \Theta(1)$ even though $\log \mathsf{P}(\text{EQ}) = \Theta(\log n)$. However, our proof of Theorem 18 does in fact construct a private-coin PP protocol and hence Corollary 19 follows from a standard exponential-loss derandomization lemma [KN97, Lemma 3.8]. For the sake of exposition, we give a self-contained proof of Corollary 19 first, and only then do we prove the slightly more complicated Theorem 18.

7.1 A weak lower bound

Proof of Corollary 19. Let Π be a $\mathsf{UAM}_{3/4,1/4}$ protocol for f of communication cost $|\Pi|$. We convert this protocol to a deterministic protocol of communication cost $2^{O(|\Pi|)}$.

We first need some notation. Let $\pi_{xy}(\tau)$ be the probability that Π accepts the proof τ on input (x, y). Because τ is a rectangle, we can write $\pi_{xy}(\tau) = a_x(\tau) \cdot b_y(\tau)$ where a_x and b_y are some functions known to Alice and Bob, respectively. Let $\mathcal{A}_x$ be the set of all pairs of proofs $\{\tau, \tau'\}$ with $\tau \neq \tau'$ such that Alice's execution of Π on input x would accept both proofs τ and τ' simultaneously under some outcome of her private coins. Alice can construct $\mathcal{A}_x$ in a brute force manner. Define $\mathcal{B}_y$ similarly.

Simulation. The deterministic protocol consists of just a single message from Alice to Bob, after which Bob can compute the answer without communication. That is, the protocol is *one-way*.

Deterministic one-way protocol. On input (x, y):

1. Alice sends Bob a message containing:
   (a) An encoding of the set $\mathcal{A}_x$.
   (b) For each proof τ, an encoding $\tilde{a}_x(\tau)$ of the value $a_x(\tau)$ up to some ℓ bits of precision.
2. Bob computes the set $\mathcal{A}_x \cap \mathcal{B}_y$.
   (a) If $\mathcal{A}_x \cap \mathcal{B}_y \neq \emptyset$ then Bob rejects.
   (b) If $\mathcal{A}_x \cap \mathcal{B}_y = \emptyset$ then Bob accepts iff $\sum_\tau \tilde{a}_x(\tau) \cdot b_y(\tau) > 1/2$.

Analysis. If $\mathcal{A}_x \cap \mathcal{B}_y \neq \emptyset$, then some pair of proof rectangles must intersect inside the block of (x, y). In this case $f(x, y) = 0$ as Π is unambiguous. If $\mathcal{A}_x \cap \mathcal{B}_y = \emptyset$ then all proof rectangles are pairwise disjoint inside (x, y), in which case the acceptance probability is $\sum_\tau a_x(\tau) \cdot b_y(\tau)$. A simple calculation [KN97, Lemma 3.8] shows that some $\ell \leq O(|\Pi|)$ bits of precision suffice to ensure that the value computed by Bob on line 2.(b) is within $< 1/4$ of the acceptance probability, so by completeness and soundness the value is $> 1/2$ iff $f(x, y) = 1$. In all cases, Bob outputs the correct answer. The communication cost is $2^{2|\Pi|} + \ell \cdot 2^{|\Pi|} \leq 2^{O(|\Pi|)}$ bits. □

7.2 A strong lower bound

Proof of Theorem 18. Let Π be a k-$\mathsf{UAM}_{3/4,1/4}$ protocol for f of cost $|\Pi|$. We convert Π into a PP protocol Γ of cost $O(k|\Pi|)$. Fix some input (x,y) and let E_τ denote the event that Π accepts a proof τ on input (x,y). With this notation, $\mathbb{P}[\Pi(x,y) \text{ accepts}] = \mathbb{P}[\cup_\tau E_\tau]$. By inclusion–exclusion, $\mathbb{P}[\cup_\tau E_\tau] = \sum_I (-1)^{|I|+1}\mathbb{P}[E_I]$ where I ranges over all nonempty sets of proofs and $E_I = \cap_{\tau\in I} E_\tau$. It is a basic fact that if we truncate the inclusion–exclusion formula at the k-th level—i.e., we only sum over sets I of cardinality $\le k$—then if k is odd we get an overestimate for $\mathbb{P}[\cup_\tau E_\tau]$, and if k is even we get an underestimate. We may assume that k is even (replace k by $k+1$ if necessary) so that

$$\mathbb{P}[\cup_\tau E_\tau] \ge \sum_{I\,:\,|I|\in[k]} (-1)^{|I|+1}\mathbb{P}[E_I]. \tag{8}$$

If (x,y) is a 1-input, we have $\mathbb{P}[E_I] = 0$ for all I of cardinality $> k$ because of k-unambiguity. In this case, the right side of (8) calculates exactly the acceptance probability $\mathbb{P}[\cup_\tau E_\tau]$, which is at least $3/4$ by completeness. If (x,y) is a 0-input then the right side is at most $1/4$ by soundness.

Simulation. We design a private-coin PP protocol Γ whose acceptance probability is a scaled-and-shifted version of the right side of (8).

Protocol Γ. On input (x,y):

1. Alice samples a uniformly random subset I of proofs with $|I| \in [k]$ and sends I to Bob.
2. Run $\Gamma_I^{\pm}$ and accept accordingly.

Subprotocol $\Gamma_I^{\pm}$:

1. Run Γ_I and let $b \in \{0,1\}$ be its output. If $|I|$ is odd, output b; otherwise output $1-b$.

Subprotocol Γ_I:

1. Accept with probability $1/2$. Otherwise continue below.
2. Sample values (r,q) for the private coins of Π.
3. Accept iff Π on input (x,y) and randomness (r,q) would accept all the proofs in I.

Analysis. Clearly $\mathbb{P}[\Gamma_I(x,y) \text{ accepts}] = 1/2 + \mathbb{P}[E_I]/2$. Hence $\mathbb{P}[\Gamma_I^{\pm}(x,y) \text{ accepts}] = 1/2 + (-1)^{|I|+1}\mathbb{P}[E_I]/2$. Let m be the number of proofs used by Π so that $\log m = \Theta(|\Pi|)$. Then in the first step of Γ each outcome of I occurs with probability ϵ where $1/\epsilon = \sum_{i\in[k]} \binom{m}{i} \le m^k$. The acceptance probability of Γ can now be calculated as

$$\begin{aligned}
\mathbb{P}[\Gamma(x,y) \text{ accepts}] &= \mathbb{E}_I\, \mathbb{P}[\Gamma_I^{\pm}(x,y) \text{ accepts}] \\
&= 1/2 + \mathbb{E}_I\, (-1)^{|I|+1}\mathbb{P}[E_I]/2 \\
&= 1/2 + (\epsilon/2)\cdot \textstyle\sum_{I\,:\,|I|\in[k]} (-1)^{|I|+1}\mathbb{P}[E_I].
\end{aligned}$$

By the discussion following (8), we have that if (x,y) is a 1-input then the acceptance probability of Γ is at least $1/2 + (\epsilon/2)\cdot(3/4)$, and if (x,y) is a 0-input then the acceptance probability of Γ is at most $1/2 + (\epsilon/2)\cdot(1/4)$. Hence we have an acceptance gap of $\Theta(\epsilon)$ centered around $1/2 + (\epsilon/2)\cdot(1/2)$ (and the center can be trivially shifted to $1/2$). Since Γ communicates $O(k|\Pi|)$ bits, its total cost is $O(k|\Pi|) + \log(1/\Theta(\epsilon)) \le O(k|\Pi|) + O(k\log m) \le O(k|\Pi|)$. □

8 Query Separation of UAM from SBP

We define our query complexity measures in Section 8.1. Then we prove Theorem 5 in Section 8.2.

8.1 Definitions

A randomized decision tree $\mathcal{T}$ of height q is a probability distribution over deterministic decision trees T of height q. We assume for convenience that a deterministic decision tree of height q has a full set of 2^q leaves.

An SBP decision tree for a boolean function f is a randomized decision tree where the acceptance probability is at least α on 1-inputs and at most $\alpha/2$ on 0-inputs, for some $\alpha > 0$. Here α is called the *acceptance threshold*; α can be arbitrarily small and should be thought of as a function of the input size n for a family of decision trees. We define $\mathsf{SBP}^{\mathsf{dt}}(f)$ to be the minimum height of any SBP decision tree for f. (Note that we do not charge for α being small as in the definition of SBP communication complexity.)

We may assume without loss of generality that an SBP decision tree $\mathcal{T}$ of height q is *non-adaptive* in the sense that each $T \in \mathrm{supp}(\mathcal{T})$ is a function of some fixed set of q (as opposed to 2^q) input variables. Indeed, take each $T \in \mathrm{supp}(\mathcal{T})$ and replace it with a uniform distribution over $\{T_p\}_p$ where p is a root-to-leaf path in T and T_p accepts iff p is accepting and the input is consistent with the queries along p. Note that if T accepts a particular input, then a random tree in $\{T_p\}_p$ accepts with probability exactly 2^{-q}. In summary, we have a non-adaptive SBP decision tree whose height remains q and whose acceptance threshold is $\alpha 2^{-q}$, which is positive as required by the definition.

We define a UAM decision tree in a natural way, as a collection of deterministic decision trees $T_{r,g}$ parametrized by r and g coming from some finite sets. The acceptance probability is the probability over random r that there exists a guess g such that $T_{r,g}$ accepts the input. For unambiguity we require that for each 1-input and each r, there is at most one g such that $T_{r,g}$ accepts the input. We define $\mathsf{UAM}^{\mathsf{dt}}(f)$ to be the minimum height of any UAM decision tree for f with perfect completeness and soundness $1/2$.

8.2 Separation

Let m be even. Define a partial function $\textsc{Gut}$ on $m \times m$ boolean matrices M by

$$\textsc{Gut}(M) \coloneqq \begin{cases} 1 & \text{if each row of } M \text{ has a single 1,} \\ 0 & \text{if } m/2 \text{ rows of } M \text{ have two 1's and the other rows are all 0's,} \\ * & \text{otherwise.} \end{cases}$$

Here $*$ means "undefined", and the total input size is $n = m^2$. We are now ready to prove Theorem 5, restated here for convenience.

Theorem 5. $\mathsf{UAM}^{\mathsf{dt}}(\textsc{Gut}) \leq O(1)$ *and* $\mathsf{SBP}^{\mathsf{dt}}(\textsc{Gut}) \geq \Omega(n^{1/4})$.

Proof. The easy fact that $\mathsf{UAM}^{\mathsf{dt}}(\textsc{Gut}) \leq O(1)$ is witnessed by a decision tree that picks a random row and guesses the location of a 1 in that row. We claim that $\mathsf{SBP}^{\mathsf{dt}}(\textsc{Gut}) \geq \Omega(m^{1/2})$.

Let $Y \in \textsc{Gut}^{-1}(1)$ be a uniformly random 1-input and let $N \in \textsc{Gut}^{-1}(0)$ be a uniformly random 0-input. The following key technical lemma (which we prove below) states that the random variables Y and N are *locally indistinguishable* as viewed through small rectangles $Q \subseteq [m] \times [m]$.

More precisely, letting Y_Q denote the restriction of Y to the entries Q, we argue that any local view Y_Q appearing in a 1-input is also found in the 0-input N with comparable probability. (Note that the converse does not hold: there are outcomes of N_Q that never appear as Y_Q.)

Lemma 20. *For all $q \times q$ rectangles Q and outcomes $y \in \{0,1\}^{q \times q}$, $\mathbb{P}[N_Q = y] \geq 0.9 \cdot \mathbb{P}[Y_Q = y]$, provided $q \leq o(m^{1/2})$.*

Now assume for contradiction that $\mathcal{T}$ is a non-adaptive SBP decision tree of height $q \leq o(m^{1/2})$ for f. We think of each $T \in \operatorname{supp}(\mathcal{T})$ as a boolean function $\{0,1\}^{Q(T)} \to \{0,1\}$ where $Q(T) \subseteq [m] \times [m]$ is some $q \times q$ rectangle (we may feed T the whole $q \times q$ rectangle spanned by its q queries). That is, in notation, $T(M) = T(M_{Q(T)})$ for any input matrix M. Using Lemma 20 we can calculate

$$\begin{aligned}
\max_{M \in f^{-1}(0)} \mathbb{P}[\mathcal{T} \text{ accepts } M] &\geq \mathbb{P}[\mathcal{T} \text{ accepts } N] \\
&= \mathbb{E}_{T \sim \mathcal{T}} \mathbb{P}[T(N) = 1] \\
&= \mathbb{E}_{T \sim \mathcal{T}} \mathbb{P}[T(N_{Q(T)}) = 1] \\
&= \mathbb{E}_{T \sim \mathcal{T}} \textstyle\sum_{y : T(y)=1} \mathbb{P}[N_{Q(T)} = y] \\
&\geq \mathbb{E}_{T \sim \mathcal{T}} \textstyle\sum_{y : T(y)=1} 0.9 \cdot \mathbb{P}[Y_{Q(T)} = y] \\
&= 0.9 \cdot \mathbb{E}_{T \sim \mathcal{T}} \mathbb{P}[T(Y_{Q(T)}) = 1] \\
&= 0.9 \cdot \mathbb{E}_{T \sim \mathcal{T}} \mathbb{P}[T(Y) = 1] \\
&= 0.9 \cdot \mathbb{P}[\mathcal{T} \text{ accepts } Y] \\
&\geq 0.9 \cdot \min_{M \in f^{-1}(1)} \mathbb{P}[\mathcal{T} \text{ accepts } M],
\end{aligned}$$

which contradicts the acceptance threshold property of $\mathcal{T}$. This finishes the proof of Theorem 5 assuming Lemma 20. □

Proof of Lemma 20. Write $Q = R \times C$ where $|R| = |C| = q$. We say that a matrix is *i-heavy* if exactly i of its rows are nonzero. Hence Y is m-heavy and N is $m/2$-heavy. Let M_R denote the restriction of any matrix M to the rows R. The probability that N_R is i-heavy is given by the hypergeometric probability mass function $\operatorname{hyp}(i) := \binom{q}{i}\binom{m-q}{m/2-i}/\binom{m}{m/2}$.

For technical convenience we start by approximating this hypergeometric distribution with a binomial one. Specifically, define a random $m \times m$ matrix N' via the following procedure: independently for each row of N', with probability $1/2$ let the row be all 0's, and with probability $1/2$ let the row contain a random pair of 1's. Now the probability that N'_R is i-heavy is given by $\operatorname{bin}(i) := \binom{q}{i}2^{-q}$. For $q \leq o(m^{1/2})$, a standard estimate [JKK05, §6.2] states that $\operatorname{hyp}(i) = (1 \pm o(1))\operatorname{bin}(i)$ for all $0 \leq i \leq q$. This implies, for all y,

$$\begin{aligned}
\mathbb{P}[N_Q = y] &= \textstyle\sum_i \operatorname{hyp}(i) \cdot \mathbb{P}[N_Q = y \mid N_R \text{ is } i\text{-heavy}] \\
&= \textstyle\sum_i \operatorname{hyp}(i) \cdot \mathbb{P}[N'_Q = y \mid N'_R \text{ is } i\text{-heavy}] \\
&= \textstyle\sum_i (1 \pm o(1)) \operatorname{bin}(i) \cdot \mathbb{P}[N'_Q = y \mid N'_R \text{ is } i\text{-heavy}] \\
&= (1 \pm o(1)) \cdot \mathbb{P}[N'_Q = y].
\end{aligned} \tag{9}$$

Let y be such that $\mathbb{P}[Y_Q = y]$ is nonzero. Then y contains at most one 1 in each row. For $r \in R$ we consider the r-th rows $N'_{r,C}$, $Y_{r,C}$, y_r of N'_Q, Y_Q, y. We claim that

$$\frac{\mathbb{P}[N'_{r,C} = y_r]}{\mathbb{P}[Y_{r,C} = y_r]} \geq 1 - O(q/m). \tag{10}$$

To see this, suppose first that y_r contains a single 1. Then $\mathbb{P}[Y_{r,C} = y_r] = 1/m$ and $\mathbb{P}[N'_{r,C} = y_r] = (1/m) \cdot (m-q)/(m-1)$ so we have (10) in this case. Suppose then that y_r contains only 0's. Then $\mathbb{P}[Y_{r,C} = y_r] = 1 - q/m$ and $\mathbb{P}[N'_{r,C} = y_r] = 1/2 + (1/2) \cdot \binom{m-q}{2}/\binom{m}{2}$. It is again straightforward to verify that (10) holds. Using (10) and the independence of the rows we can now calculate

$$\frac{\mathbb{P}[N'_Q = y]}{\mathbb{P}[Y_Q = y]} = \frac{\prod_{r \in R} \mathbb{P}[N'_{r,C} = y_r]}{\prod_{r \in R} \mathbb{P}[Y_{r,C} = y_r]} \geq (1 - O(q/m))^q \geq 1 - O(q^2/m) \geq 1 - o(1). \tag{11}$$

The lemma follows by putting together (9) and (11). □

9 Open Problems

Some speculative open problems include finding applications of our results or techniques to streaming delegation or to other topics. It is also open to consider multi-party versions of any of the topics considered in this work (though we note that our upper bounds generalize straightforwardly to the number-in-hand model). We now discuss some more concrete open problems related to our complexity measures.

ZAM complexity. One or both of the following must hold, but it is open to prove either: $\mathsf{ZAM}(f) \leq 2^{o(n)}$ for all $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, or $\mathsf{ZAM}(f) \geq \omega(n)$ for some f (even a random f). Proving a superpolynomial ZAM lower bound for an explicit function would also yield a superlogarithmic lower bound against every level of the communication polynomial hierarchy (hence is probably an unrealistically ambitious goal): By definition, every function in the polynomial hierarchy has a constant-depth circuit whose size is exponential in the communication bound, and where each input bit is an arbitrary function of one party's input. Such a circuit can be straightforwardly converted into a two-party (parity) branching program, and then Theorem 1 can be applied to get a ZAM protocol. Proving a superpolynomial ZAM lower bound for a random function may be a more realistic goal, since it would not have nontrivial consequences for the polynomial hierarchy.

A particularly conspicuous open problem is to prove that $\mathsf{ZAM}(\textsc{Disj}) \geq \omega(\log n)$. The function $\textsc{Index}\colon \{0,1\}^n \times [n] \to \{0,1\}$ defined by $\textsc{Index}(x,y) = x_y$ is no harder than Disj, and yet we conjecture that $\mathsf{ZAM}(\textsc{Index}) = \Theta(n)$. Proving the latter would be a step toward showing that ZAM complexity can be superlinear, since it would show that ZAM complexity can be exponential in *one* party's input length (which cannot happen for deterministic communication complexity).

It would be interesting to show that $\mathsf{ZAM}(f) \geq \omega(\mathsf{coNP}(f))$ for some f. This would require new techniques for understanding ZAM protocols. It would also be interesting to prove new lower bounds even for the special case of ZAM where for each 1-input there is a *bijection* between Merlin's proofs and the outcomes of the randomness; note that all of our ZAM upper bounds have this property. One possible approach for obtaining new ZAM lower bounds is to use information complexity tools to lower bound the information cost—not of the distribution of transcripts (obviously) but of some related distribution, such as correlated tuples of transcripts.

UAM complexity. It is open to prove, in the communication world, that $\mathsf{UAM} \not\subseteq \mathsf{SBP}$ (which would show that the corruption bound does not automatically lower bound UAM complexity) or even $\mathsf{UAM} \not\subseteq \mathsf{MA}$, even for a partial function. It would be very interesting to develop techniques for converting SBP query lower bounds into analogous SBP communication lower bounds. It would

also be interesting to compare UAM with other lower bound techniques in the literature (like how Klauck [Kla10] showed that the smooth rectangle bound lower bounds $\mathsf{BP}\cdot\mathsf{UP}$ complexity).

Another direction for exploring the power of UAM is to design interesting protocols. Here is a framework for designing UAM protocols that might be useful: Suppose Alice can construct from her input x a matrix A, and Bob can construct from his input y a matrix B, such that $f(x,y)=1$ iff $\det(A+B)\neq 0$. If no row is nonzero in both A and B, then of course $A+B$ is a two-party matrix and the proof of Theorem 1 yields a ZAM protocol. Otherwise, we can at least get a UAM protocol: Alice generates a random vector $\boldsymbol{v}$, and Merlin sends a claimed preimage $\boldsymbol{u}$ and claimed vectors $A\boldsymbol{u}$ and $B\boldsymbol{u}$. Alice checks that the latter vectors sum to $\boldsymbol{v}$ and that $A\boldsymbol{u}$ is as claimed, and Bob checks that $B\boldsymbol{u}$ is as claimed. If the matrices have size $k\times k$, then we would need $k\cdot\log|\mathbb{F}|\leq o(n)$ for the protocol to be nontrivial. Furthermore, this approach would yield a *conondeterministic* protocol of cost $O(k\cdot\log|\mathbb{F}|)$ (Merlin sends distinct vectors with the same image under $A+B$, and sends the images of these vectors under A and B) and hence would not be useful for functions with $\mathsf{coNP}(f)=\Theta(n)$.

One odd property of UAM is that it does not seem conducive to efficient amplification of the completeness probability. Taking a threshold of several independent executions raises the level of ambiguity (number of rectangles that may simultaneously intersect for a 1-input), and our proof of Theorem 3 shows that ambiguity cannot, in general, be efficiently decreased: Inter is hard for UAM under the promise that at most two coordinates intersect; yet this promise problem has a trivial efficient 2-UAM protocol (using the notation from Section 7) that nondeterministically guesses an intersecting coordinate. It would be interesting to have formal evidence for or against the possibility of completeness amplification for UAM.

AM complexity. The principal open problem here is to prove that $\mathsf{AM}(f)\geq\omega(\log n)$ for some explicit f. The only known AM lower bounds follow from the observation that $\mathsf{AM}(f)\geq\Omega(\log\mathsf{BPP}(f))$ for all f. This implies that $\mathsf{AM}(\textsc{Eq}),\mathsf{AM}(\textsc{Neq}),\mathsf{AM}(\textsc{Gt})\geq\Omega(\log\log n)$, and in fact it is an open problem to prove a $\omega(\log\log n)$ lower bound for any of these three problems.

One possible approach for proving AM lower bounds is to try to reduce to a situation with some amount of unambiguity by using an isolation lemma, and then combine this with our techniques for proving UAM lower bounds. However, we do not see a straightforward way to make this work.

A less ambitious goal might be to prove an explicit linear lower bound for the communication complexity measure that combines the completeness condition of MA and the soundness condition of AM. Such protocols are weaker than *both* MA and AM protocols, so $\Omega(\sqrt{n})$ lower bounds follow from known MA lower bounds. Whereas there seems to be little hope for linear MA lower bounds ([AW09] gives a $O(\sqrt{n}\log n)$ upper bound for many interesting problems), there may be hope for these weaker protocols.

Finally, we highlight the lurking prospect of designing AM protocols that defy our intuitions.

Acknowledgements

We thank Petteri Kaski, Venkatesh Medabalimi, and Robert Robere for discussions.

References

[ACC+14] Anil Ada, Arkadev Chattopadhyay, Stephen Cook, Lila Fontes, Michal Koucký, and Toniann Pitassi. The hardness of being private. *ACM Transactions on Computation Theory*, 6(1), 2014. doi:10.1145/2567671.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006. doi:10.1137/S0097539705446950.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Transactions on Computation Theory*, 1(1), 2009. doi:10.1145/1490270.1490272.

[BEO+13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *Proceedings of the 54th Symposium on Foundations of Computer Science (FOCS)*, pages 668–677. IEEE, 2013. doi:10.1109/FOCS.2013.77.

[BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *Proceedings of the 27th Symposium on Foundations of Computer Science (FOCS)*, pages 337–347. IEEE, 1986. doi:10.1109/SFCS.1986.15.

[BGM06] Elmar Böhler, Christian Glaßer, and Daniel Meister. Error-bounded probabilistic computations between MA and AM. *Journal of Computer and System Sciences*, 72(6):1043–1076, 2006. doi:10.1016/j.jcss.2006.05.001.

[BGPW13] Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. From information to exact communication. In *Proceedings of the 45th Symposium on Theory of Computing (STOC)*, pages 151–160. ACM, 2013. doi:10.1145/2488608.2488628.

[BM88] László Babai and Shlomo Moran. Arthur–Merlin games: A randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988. doi:10.1016/0022-0000(88)90028-1.

[BM13] Mark Braverman and Ankur Moitra. An information complexity approach to extended formulations. In *Proceedings of the 45th Symposium on Theory of Computing (STOC)*, pages 161–170. ACM, 2013. doi:10.1145/2488608.2488629.

[BRS95] Richard Beigel, Nick Reingold, and Daniel Spielman. PP is closed under intersection. *Journal of Computer and System Sciences*, 50(2):191–202, 1995. doi:10.1006/jcss.1995.1017.

[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. doi:10.1016/j.jcss.2003.11.006.

[CCGT14] Amit Chakrabarti, Graham Cormode, Navin Goyal, and Justin Thaler. Annotations for sparse data streams. In *Proceedings of the 25th Symposium on Discrete Algorithms (SODA)*, pages 687–706. ACM-SIAM, 2014. doi:10.1137/1.9781611973402.52.

[CCM09] Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Annotations in data streams. In *Proceedings of the 36th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 222–234. Springer, 2009. doi:10.1007/978-3-642-02927-1_20.

[CCM+13] Amit Chakrabarti, Graham Cormode, Andrew McGregor, Justin Thaler, and Suresh Venkatasubramanian. On interactivity in Arthur–Merlin communication and stream computation. Technical Report TR13-180, Electronic Colloquium on Computational Complexity (ECCC), 2013. URL: http://eccc.hpi-web.de/report/2013/180/.

[CKS03] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *Proceedings of the 18th Conference on Computational Complexity (CCC)*, pages 107–117. IEEE, 2003. doi:10.1109/CCC.2003.1214414.

[CSWY01] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proceedings of the 42nd Symposium on Foundations of Computer Science (FOCS)*, pages 270–278. IEEE, 2001. doi:10.1109/SFCS.2001.959901.

[Dam90] Carsten Damm. Problems complete for ⊕L. *Information Processing Letters*, 36(5):247–250, 1990. doi:10.1016/0020-0190(90)90150-V.

[DKS12] Anirban Dasgupta, Ravi Kumar, and D. Sivakumar. Sparse and lopsided set disjointness via information theory. In *Proceedings of the 16th International Workshop on Randomization and Computation (RANDOM)*, pages 517–528. Springer, 2012. doi:10.1007/978-3-642-32512-0_44.

[FKN94] Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation. In *Proceedings of the 26th Symposium on Theory of Computing (STOC)*, pages 554–563. ACM, 1994. doi:10.1145/195058.195408.

[GR13a] Tom Gur and Ran Raz. Arthur–Merlin streaming complexity. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 528–539. Springer, 2013. doi:10.1007/978-3-642-39206-1_45.

[GR13b] Tom Gur and Ron Rothblum. Non-interactive proofs of proximity. Technical Report TR13-078, Electronic Colloquium on Computational Complexity (ECCC), 2013. URL: http://eccc.hpi-web.de/report/2013/078/.

[Gro09] André Gronemeier. Asymptotically optimal lower bounds on the NIH-multi-party information complexity of the AND-function and disjointness. In *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 505–516. Schloss Dagstuhl, 2009. doi:10.4230/LIPIcs.STACS.2009.1846.

[GS10] Dmitry Gavinsky and Alexander Sherstov. A separation of NP and coNP in multiparty communication complexity. *Theory of Computing*, 6(1):227–245, 2010. doi:10.4086/toc.2010.v006a010.

[GW96] Anna Gál and Avi Wigderson. Boolean complexity classes vs. their arithmetic analogs. *Random Structures & Algorithms*, 9(1-2):99–111, 1996. doi:10.1002/(SICI)1098-2418(199608/09)9:1/2⟨99::AID-RSA7⟩3.0.CO;2-6.

[GW14] Mika Göös and Thomas Watson. Communication complexity of set-disjointness for all probabilities. In *Proceedings of the 18th International Workshop on Randomization and Computation (RANDOM)*. Schloss Dagstuhl, 2014. To appear.

[IK02] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 244–256. Springer, 2002. doi:10.1007/3-540-45465-9_22.

[IKK09] Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. An axiomatic approach to algebrization. In *Proceedings of the 41st Symposium on Theory of Computing (STOC)*, pages 695–704. ACM, 2009. doi:10.1145/1536414.1536509.

[IW10] Russell Impagliazzo and Ryan Williams. Communication complexity with synchronized clocks. In *Proceedings of the 25th Conference on Computational Complexity (CCC)*, pages 259–269. IEEE, 2010. doi:10.1109/CCC.2010.32.

[Jay09] T.S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of AND. In *Proceedings of the 13th International Workshop on Randomization and Computation (RANDOM)*, pages 562–573. Springer, 2009. doi:10.1007/978-3-642-03685-9_42.

[JK10] Rahul Jain and Hartmut Klauck. The partition bound for classical communication complexity and query complexity. In *Proceedings of the 25th Conference on Computational Complexity (CCC)*, pages 247–258. IEEE, 2010. doi:10.1109/CCC.2010.31.

[JKK05] Norman Johnson, Adrienne Kemp, and Samuel Kotz. *Univariate Discrete Distributions*. Wiley, 3rd edition, 2005.

[JKS03] T.S. Jayram, Ravi Kumar, and D. Sivakumar. Two applications of information complexity. In *Proceedings of the 35th Symposium on Theory of Computing (STOC)*, pages 673–682. ACM, 2003. doi:10.1145/780542.780640.

[Juk06] Stasys Jukna. On graph complexity. *Combinatorics, Probability, & Computing*, 15(6):855–876, 2006. doi:10.1017/S0963548306007620.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012.

[Kla03] Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *Proceedings of the 18th Conference on Computational Complexity (CCC)*, pages 118–134. IEEE, 2003. doi:10.1109/CCC.2003.1214415.

[Kla07] Hartmut Klauck. Lower bounds for quantum communication complexity. *SIAM Journal on Computing*, 37(1):20–46, 2007. doi:10.1137/S0097539702405620.

[Kla10] Hartmut Klauck. A strong direct product theorem for disjointness. In *Proceedings of the 42nd Symposium on Theory of Computing (STOC)*, pages 77–86. ACM, 2010. doi:10.1145/1806689.1806702.

[Kla11] Hartmut Klauck. On Arthur Merlin games in communication complexity. In *Proceedings of the 26th Conference on Computational Complexity (CCC)*, pages 189–199. IEEE, 2011. doi:10.1109/CCC.2011.33.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[KNSW92] Mauricio Karchmer, Ilan Newman, Michael Saks, and Avi Wigderson. Non-deterministic communication complexity with few witnesses. In *Proceedings of the 7th Conference on Structure in Complexity Theory (CCC)*, pages 275–281. IEEE, 1992. doi:10.1109/SCT.1992.215402.

[KP13] Hartmut Klauck and Ved Prakash. Streaming computations with a loquacious prover. In *Proceedings of the 4th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 305–320. ACM, 2013. doi:10.1145/2422436.2422471.

[KP14a] Hartmut Klauck and Supartha Podder. Two results about quantum messages. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS)*. Springer, 2014. To appear.

[KP14b] Hartmut Klauck and Ved Prakash. An improved interactive streaming algorithm for the distinct elements problem. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*. Springer, 2014. To appear.

[Kus92] Eyal Kushilevitz. Privacy and communication complexity. *SIAM Journal on Discrete Mathematics*, 5(2):273–284, 1992. doi:10.1137/0405021.

[Lin91] Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi:10.1109/18.61115.

[Lok01] Satyanarayana Lokam. Spectral methods for matrix rigidity with applications to size-depth trade-offs and communication complexity. *Journal of Computer and System Sciences*, 63(3):449–473, 2001. doi:10.1006/jcss.2001.1786.

[Lok09] Satyanarayana Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1–2):1–155, 2009. doi:10.1561/0400000011.

[LS09] Nathan Linial and Adi Shraibman. Learning complexity vs communication complexity. *Combinatorics, Probability, & Computing*, 18(1–2):227–245, 2009. doi:10.1017/S0963548308009656.

[PRS88] Pavel Pudlák, Vojtech Rödl, and Petr Savický. Graph complexity. *Acta Informatica*, 25(5):515–535, 1988. doi:10.1007/BF00279952.

[PSS14] Periklis Papakonstantinou, Dominik Scheder, and Hao Song. Overlays and limited memory communication mode(l)s. In *Proceedings of the 29th Conference on Computational Complexity (CCC)*. IEEE, 2014. To appear.

[RA00] Klaus Reinhardt and Eric Allender. Making nondeterminism unambiguous. *SIAM Journal on Computing*, 29(4):1118–1131, 2000. doi:10.1137/S0097539798339041.

[Raz89] Alexander Razborov. On rigid matrices. Technical report, Steklov Mathematical Institute, 1989. In Russian.

[Raz92] Alexander Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992. doi:10.1016/0304-3975(92)90260-M.

[RS04] Ran Raz and Amir Shpilka. On the power of quantum proofs. In *Proceedings of the 19th Conference on Computational Complexity (CCC)*, pages 260–274. IEEE, 2004. doi:10.1109/CCC.2004.1313849.

[San89] Miklos Santha. Relativized Arthur–Merlin versus Merlin–Arthur games. *Information and Computation*, 80(1):44–49, 1989. doi:10.1016/0890-5401(89)90022-9.

[Sch89] Uwe Schöning. Probabilistic complexity classes and lowness. *Journal of Computer and System Sciences*, 39(1):84–100, 1989. doi:10.1016/0022-0000(89)90020-2.

[Val77] Leslie Valiant. Graph-theoretic arguments in low-level complexity. In *Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 162–176. Springer, 1977. doi:10.1007/3-540-08353-7_135.

[Val79] Leslie Valiant. Completeness classes in algebra. In *Proceedings of the 11th Symposium on Theory of Computing (STOC)*, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.

[Wun12a] Henning Wunderlich. A note on a problem in communication complexity. Technical report, arXiv, 2012. arXiv:1205.0903.

[Wun12b] Henning Wunderlich. On a theorem of Razborov. *Computational Complexity*, 21(3):431–477, 2012. doi:10.1007/s00037-011-0021-5.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de