# The Communication Complexity of Set Intersection Under Product Distributions

Rotem Oshman
Tel-Aviv University, Israel

Tal Roth
Tel-Aviv University, Israel

## Abstract

We consider a multiparty setting where $k$ parties have private inputs $X_1, \ldots, X_k \subseteq [n]$ and wish to compute the intersection $\bigcap_{\ell=1}^{k} X_\ell$ of their sets, using as little communication as possible. This task generalizes the well-known problem of set disjointness, where the parties are required only to determine whether the intersection is empty or not. In the worst-case, it is known that the communication complexity of finding the intersection is the same as that of solving set disjointness, regardless of the size of the intersection: the cost of both problems is $\Omega\left(n \log k + k\right)$ bits in the shared blackboard model, and $\Omega\left(nk\right)$ bits in the coordinator model.

In this work we consider a realistic setting where the parties' inputs are independent of one another, that is, the input is drawn from a product distribution. We show that this makes finding the intersection significantly easier than in the worst-case: only $\tilde{\Theta}((n^{1-1/k}\left(\mathrm{H}(S)+1\right)^{1/k}) + k)$ bits of communication are required, where $\mathrm{H}(S)$ is the Shannon entropy of the intersection $S$. We also show that the parties do not need to know the exact underlying input distribution; if we are given in advance $O(n^{1/k})$ samples from the underlying distribution $\mu$, we can learn enough about $\mu$ to allow us to compute the intersection of an input drawn from $\mu$ using expected communication $\tilde{\Theta}((n^{1-1/k}\,\mathbb{E}[|S|]^{1/k}) + k)$, where $|S|$ is the size of the intersection.

**2012 ACM Subject Classification** Theory of computation → Communication complexity

**Keywords and phrases** Communication complexity, intersection, set disjointness

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2023.95

**Category** Track A: Algorithms, Complexity and Games

**Funding** Research funded by the Israel Science Foundation, Grant No. 2801/20, and also supported by Len Blavatnik and the Blavatnik Family foundation.

## 1 Introduction

Communication complexity is concerned with understanding the communication cost of computing on data that is partitioned between $k \geq 2$ parties, with each party holding a private input $\boldsymbol{X}^i$.[1] The parties would like to jointly compute some function $f(\boldsymbol{X}^1, \ldots, \boldsymbol{X}^k)$ of their data, using as little communication as possible. Two models of communication are typically studied: in the *shared blackboard* model, the parties communicate by writing messages on a "board" that all the other parties can read (essentially, they communicate by broadcast); in the *private-channel* model, the parties communicate over private channels. For both models, there is a wealth of protocols and lower bounds characterizing the cost of computing different functions, and obtaining applications in areas ranging from distributed graph algorithms (see [30, 10, 6, 8, 9] and many more), to streaming algorithms (e.g., [1, 3, 15])

[1] This is called *number-in-hand* because each party holds its own private input; in the *number-on-forehead* model, each party can see the inputs of all the other parties, but not its own input. The number-on-forehead model has compelling applications in circuit complexity, but it is not a realistic model of a distributed system.

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).
Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 95; pp. 95:1–95:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and beyond. We focus on the *distributional setting*, where the inputs $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^k$ are drawn from a distribution $\mu$, and our goal is to compute $f(\boldsymbol{X}^1, \ldots, \boldsymbol{X}^k)$ with a low error probability over $\mu$.

In this paper we study the cost of computing an intersection: each party holds a set $\boldsymbol{X}^i \subseteq [n]$, and our goal is to output the intersection, $\boldsymbol{S} = \bigcap_{i=1}^{n} \boldsymbol{X}^i$. This fundamental problem has many applications, including computing joins for distributed databases [10, 16]; computing the Jaccard similarity of data sets, the number of distinct elements, and rarity [10]; and algebraic function computation on reconciled data [19]. Recently this problem also found applications in online advertising [14], notably at Google [18].

Computing the intersection of sets $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^k$ is as at least as hard as solving the *set disjointness* problem, where we are required to determine whether $\cap_{i=1}^{k} \boldsymbol{X}^i = \emptyset$. Set disjointness is known to require $\Omega(n)$ bits of communication for two-parties [26, 24], $\Omega(n \log k)$ bits of communication for $k$ parties on the shared blackboard [8], and $\Omega(kn)$ bits of communication for $k$ parties with private channels [6]. However, all of these hardness results hold only when the parties' inputs are highly-correlated; if the input is drawn from a *product distribution* (i.e., the parties' inputs are independent of one another), then two-party set disjointness requires only $\Theta(\sqrt{n})$ bits [2, 5], and this was recently extended to the multi-party setting, where it was shown that the communication cost is $\tilde{\Theta}(n^{1-1/k})$ in both the shared blackboard and the coordinator models [12]. We note that all the lower bounds mentioned above hold even for input distributions where the intersection is of small constant size – that is, either the input sets do not intersect at all, or they have a non-empty but constant-sized intersection, and our goal is to distinguish between these two cases.

Our main question in this paper is whether under product distributions we can efficiently compute the full intersection of the parties' inputs, rather than merely determining whether it is empty or not. We show that the answer is *yes*, at least when the intersection is not too large: if the expected intersection size is $\mathbb{E}[\boldsymbol{S}] = s$, then we can compute the full intersection using $\tilde{\Theta}(n^{1-1/k} s^{1/k})$ bits of communication in expectation, in both the shared blackboard and the coordinator models. This can be viewed as a natural extension of the tight bound for set disjointness in the product case, which is $\tilde{\Theta}(n^{1-1/k})$, even when $s = \Theta(1)$ [12]. Our protocol and lower bound bear some similarities to [12]. We generalize our result in two ways, motivated by practical applications.

**"Learning" the input distribution.** In this scenario, instead of being told the input distribution $\mu$, we are given iid *samples* from $\mu$, and must "learn" whatever we can about $\mu$ before running the protocol on the actual input (which is also drawn from $\mu$). Can we learn enough about $\mu$ to exploit its product structure, without requiring a prohibitive number of samples? In Section 4.1 we show that the answer is *yes*: $\tilde{O}(n^{1/k})$ sample suffice to learn enough about $\mu$ to solve any future instance with optimal communication cost.

▶ **Theorem 1.** *Let $\delta > 0$, and assume we have access to $O\left(n^{1/k} \log\left(nk/\delta\right)\right)$ iid samples from an unknown product distribution $\mu$. Then we can construct a zero-error two-round protocol $\Pi$ for computing the intersection, such that with probability at least $1-\delta$ over the samples, the protocol $\Pi$ that we constructed has expected communication cost $O\left(kn^{1-1/k}\, \mathbb{E}\left[|\boldsymbol{S}|+1\right]^{1/k} \log n\right)$ on inputs drawn from $\mu$.*

In particular, when $k \geq \log n$, we require only a single sample from $\mu$. This is perhaps surprising, since it is known that in order to fully learn a distribution over $kn$ bits – that is, in order to output a distribution $\mu'$ that is $\epsilon$-close to $\mu$ in statistical distance – the number of samples required is $\Omega\left(2^{nk}/(nk)\right)$ [28]. The key is that instead of learning the entire distribution, we show that it suffices to estimate the marginal expectation of each input bit, which is a much easier task.

It remains open whether $\Omega(n^{1/k})$ samples are truly necessary to obtain the optimal communication complexity, and more generally, what is the tradeoff between the number of samples we have and the communication complexity we can obtain. However, we show in Section 5 that if we do not have *any* prior information about the distribution $\mu$ (i.e., no samples), then the fact that $\mu$ is a product distribution is not helpful at all: for any function $f$, computing $f$ under an unknown product distribution is as hard as computing $f$ under non-product distributions.

**Large but predictable intersections.** Although we assume that the parties have independent inputs, we do not assume that the elements inside a given party's input are independent of one another: for example, if each party's input is a list of items purchased by some set of customers, then the elements may be highly correlated, as one item purchased by a customer is likely to tell us a lot about other items the same customer is likely to purchase. Correlations between elements can lead to a situation where the intersection is "large but fairly predictable", in the sense that while the intersection $S$ has large expected size, its Shannon entropy $\mathrm{H}(S)$ is much smaller. As an extreme example, if we have two parties with inputs $X, Y$ that are each either $[n]$ or $\emptyset$ with probability $1/2$, then the expected size of the intersection is $n/2$, but its Shannon entropy is only 1.

In Section 4.3 we show that it is not the size of the intersection but its entropy that matters: when the distribution $\mu$ is known, we can replace the size $|\boldsymbol{S}|$ of the intersection with its entropy, $\mathrm{H}(\boldsymbol{S})$, and obtain the following.

▶ **Theorem 2.** *Let $\mu$ be a product distribution known to all the parties. Then in the coordinator model, there is an $O(\log n)$-round zero-error deterministic protocol for finding the intersection, with expected communication cost at most $O\left(k^2 n^{1-1/k} \left(\mathrm{H}(\boldsymbol{S})+1\right)^{1/k} \log n + k\right)$, where the expectation is with respect to the input distribution $\mu$.*

We remark that for non-product distributions this is not possible: in the hard distribution of Razborov [24] for two-party set disjointness, the intersection has entropy $O(\log n)$, as it is always either empty or contains a single element which is uniformly random over $[n]$. Nevertheless, even determining whether the intersection is empty or not requires $\Omega(n)$ bits of communication, and this of course implies that *finding* the intersection also requires $\Omega(n)$ communication.

**Lower bounds.** To complement our protocols above, in Section 5 we prove a matching lower bound, up to polylogarithmic factors:

▶ **Theorem 3.** *For every $n, k \in \mathbb{N}$ with $2 \le k \le \log n$, and for every $s \in [1, n/2]$, there exists a product distribution $\mu$ over $\{0,1\}^{n \times k}$ such that*

- $\mathbb{E}_{\mu}\left[\,|\boldsymbol{S}|\,\right] = s$,
- $s \le \mathrm{H}(\boldsymbol{S}) \le (s+1)\log n$, *and*
- *Any deterministic protocol for computing the intersection with error probability at most $1/10$ over $\mu$ has expected communication complexity $\Omega(n^{1-1/k} s^{1/k}/k^2)$.*

Although the lower bound is stated only for $k \le \log n$ parties, we can "stretch" the lower bound from $k = \log n$ to larger $k$ by generating the inputs of the first $\log n$ parties using the hard distribution from the theorem, and giving the remaining parties the set $[n]$. As a result, for $k > \log n$, we obtain a lower bound of $\tilde{\Omega}(n)$ regardless of the intersection size $s$, and this is tight up to polylogarithmic factors.

Our lower bound actually applies to a weak output model, where every element of the intersection can be output by a different party: at the end of the protocol, each party $\ell$ outputs a list of decisions of the form "$i \in \boldsymbol{S}$" or "$i \notin \boldsymbol{S}$". We require that for every coordinate $i \in [n]$, one party must output a decision for $i$, but the identity of the party that output a decision for $i$ need not be known in advance (that is, it may be a function of the transcript). The party $j$ that outputs a decision for $i$ may rely on its own input $\boldsymbol{X}^j$ when making the decision. This output model is quite weak compared to the standard output model that we assume in our protocols, where the output to the computation must be computable from the transcript of the protocol. Making the lower bound work in this weak model is technically challenging: our lower bound uses information-theoretic arguments, which typically rely on the fact that an *external observer* must learn a lot of information about the inputs, but this is not necessarily true in the weak output model.

We also remark that all of the results discussed up to this point (both upper and lower bounds) assume that the protocol must output the *entire* intersection correctly with high probability: if we output a set that differs from the true intersection in even a single element, this is considered an error. One can also consider a weaker notion, which is more appealing for lower bounds, where for every $i \in [n]$, we only need to determine whether $i \in \boldsymbol{S}$ with good *marginal* error probability, independent of the other elements. This weaker notion is only meaningful when many coordinates have constant probability bounded away from 0 and 1 of being in the intersection, otherwise we can simply guess independently for each coordinate whichever outcome is more likely for that coordinate; e.g., if $\Pr[i \in \boldsymbol{S}] = 1/\sqrt{n}$ for every $i$, we can guess that the intersection is empty, and still be correct on every element with marginal probability $1 - 1/\sqrt{n}$. However, if every element has probability between 1/3 and 1/2 of being in the intersection, then we can also prove a tight lower bound even for the case where the protocol only needs to succeed with good marginal probability on each element (see the full version of this paper for a proof of this theorem).

## 2 Related Work

Set disjointness has been studied extensively, in many versions and models; we refer to the surveys [11, 27] for more background. The problem of computing the intersection, or of finding an element in the intersection, has also been studied, for two parties [7, 25, 10, 13, 29, 4, 17] and for more than two parties [10, 23]; to our knowledge, all previously mentioned prior work is for either worst-case hardness (that is, a non-distributional setting, where the inputs are chosen adversarially), or for non-product distributions, and is thus not directly relevant to the current paper. In addition, [20, 21, 22] studied a different scenario where two parties wish to compute the bitwise-AND of their input vectors (as well as other functions), assuming the coordinates of the vectors are iid, in the regime where the input length goes to infinity and the error is vanishing. In contrast, here we consider multi-party intersection with a fixed input length and constant error, and we do not assume that the coordinates are iid.

The hardness of set disjointness under product distributions was first studied in [2], which proved a lower bound of $\Omega(\sqrt{n})$ and an upper bound of $O(\sqrt{n}\log n)$ on the communication complexity of the problem. Later, [5] eliminated the log-factor and improved the upper bound to $O(\sqrt{n})$, and showed that in general, when the parties' inputs have mutual information $I$ with one another, the communication cost of set disjointness is $\tilde{\Theta}(\sqrt{n(I+1)})$ (the product case is the case where $I = 0$). It turns out that the techniques of [2, 5] do not scale to more than 2 parties, but in [12], using different techniques, it was shown that $\tilde{\Theta}(n^{1-1/k})$ bits are necessary and sufficient in the $k$-player setting.

The protocols of [2, 5, 12] for set disjointness share the following feature: at any point in the protocol, if we identify that given what we have learned so far the probability that the inputs are disjoint is bounded by some small threshold $\epsilon$, then we halt and output "not disjoint". If the probability of disjointness is greater than $\epsilon$, we rely on this fact to make progress: in [2, 5], we use it to efficiently sample a large set that is disjoint from one player's input, and those elements are then discarded from consideration; in [12], we exploit the fact that no single element is likely to be in the intersection to show that each element $i \in [n]$ is probably missing from the input of some specific player $p(i)$. We then ask each player $j$ to say only the elements $X_j \cap \{i : p(i) = j\}$, as this set is likely to be small, but at the same time it helps us learn of many elements that are definitely not in the intersection. If we want to find the intersection in full, the basic approach of [12] continues to work if we know that we have a *small* intersection, but it breaks down when the intersection is large.

Our work generalizes the basic approach of [12] to handle larger intersection sizes. This yields a protocol for finding the intersection that depends on the *entropy* of the intersection instead of its expected size (as already noted in the introduction, the former may be much smaller than the latter). We also show that the basic protocol of [12] can be made *robust*, in the sense that the players do not need to know the exact underlying input distribution.

When the number of players is $k \geq \log n$, [12] gives a different protocol that actually finds the entire intersection, and has communication cost $\tilde{O}(n)$. We show that this protocol can be made robust as well, and in fact a *single* sample from the underlying product distribution is enough, with high probability, for the players to be able to successfully execute the protocol.

As for lower bounds, [12] gave a lower bound on finding the intersection under a product distribution, for the case where the expected intersection size is 1 (which coincides with the set disjointness problem), when the transcript reveals the intersection to an *external observer*. In this work, we generalize this lower bound in two ways. First, our lower bound must handle larger intersections, up to linear in $n$. This large range of intersection sizes implies, naturally, that the lower bound proof must handle both very small and very large probabilities, which requires delicate handling. Secondly, our bound is proven in the weaker model where for each coordinate, one of the players must decide whether this coordinate is in the intersection or not, and the identity of this player may not even be known in advance.

## 3 Preliminaries

**Notation.** We use boldface letters to denote random variables. Given a vector $v$ indexed by $\{1, \ldots, m\}$ and a subset of coordinates $J = \{j_1, \ldots, j_\ell\}$, we denote by $v_J = v_{j_1,\ldots,j_\ell}$ coordinates $J$ of $v$. If $\boldsymbol{A}$ is a random variable and $\mathcal{E}$ is an event, then $\boldsymbol{A}|_{\mathcal{E}}$ denotes the distribution of $\boldsymbol{A}$ conditioned on $\mathcal{E}$.

The input to the $k$ players is denoted $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^k \in \{0,1\}^n$. It is convenient to sometimes view the inputs to the players as sets, and sometimes as the characteristic vectors of their sets. We use $\boldsymbol{X}_i^\ell$ to denote the $i$-th coordinate of player $\ell$'s input, when viewed as a characteristic vector. The intersection of the players' inputs is denoted $\boldsymbol{S} = \bigcap_{\ell \in [k]} \boldsymbol{X}^\ell$, and for each $i \in [n]$, $\boldsymbol{S}_i$ is an indicator for the event "$i \in \boldsymbol{S}$". We refer to $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n$ as the *bits of the intersection*.

We sometimes abuse notation by conflating Bernoulli distributions with their expected value: for example, if the input distribution is $\mu$, we use $\mu_i^\ell$ to denote both the marginal distribution of $\boldsymbol{X}_i^\ell$, and the expected value of $\boldsymbol{X}_i^\ell$.

**The shared blackboard model.** In this classical model of multiparty communication, we have $k$ players, with private inputs $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^k$. The players communicate by writing on a *shared blackboard* that all players can see. At any point in the protocol, the identity of the next player to write on the board is determined by the current contents of the board. We refer to the contents of the board as the *transcript* of the protocol, and denote it by the random variable $\mathbf{\Pi}$.

**The coordinator model.** In the coordinator model of multiparty computation, in addition to the $k$ players, we also have a *coordinator*, who has no input. The players communicate with the coordinator over private channels, but players cannot communicate directly with one another. The order of communication is governed by the coordinator, and the *transcript* of the protocol consists of all messages sent and received by the coordinator.

**Set intersection.** In the $k$-player set intersection problem, our goal is to compute:

$$\mathrm{INT}_{n,k}(X^1, \ldots, X^k) = \cap_{\ell=1}^{k} X^{\ell}.$$

Since the intersection can be very large, it is crucial that we do not charge the players for "writing" the output at the end of the protocol. Instead, we assume one of the following two output models:

- In our upper bounds, the output is some predetermined function of the transcript; in other words, an external observer can learn the intersection just by observing the transcript of the protocol, without knowing any of the inputs.
- In our lower bounds, the output is jointly produced by the players, with each player $j$ choosing at the end of the protocol a set of indices $I_j$, and outputting the bits $\{\boldsymbol{S}_i : i \in I_j\}$. The index set $I_j$ may depend on the transcript of the protocol, but not on player $j$'s input. However, the *values* that player $j$ outputs for the bits $\{\boldsymbol{S}_i : i \in I_j\}$ may depend on player $j$'s input. Thus, an external observer that sees only the transcript of the protocol is able to learn which player will output which bits, but not the values of the output bits. We require that every bit $\boldsymbol{S}_i$ must be output by some player; if more than one player outputs the bit $\boldsymbol{S}_i$, and the players disagree, this is considered an error.

**Information theory and entropy.** The *Shannon entropy* of a random variable $\boldsymbol{A} \sim \mu$ is given by

$$\mathrm{H}_{\mu}(\boldsymbol{A}) = \sum_{a \in supp(\mu)} \mu(a) \log \frac{1}{\mu(a)},$$

where $supp(\mu)$ denotes the support of the distribution $\mu$. We omit the subscript $\mu$ when the distribution is clear from the context.

Given jointly-distributed variables $(\boldsymbol{A}, \boldsymbol{B}) \sim \mu$, with marginal distributions $\mu_{\boldsymbol{A}}$ and $\mu_{\boldsymbol{B}}$ respectively, the *conditional entropy* of $\boldsymbol{A}$ given $\boldsymbol{B}$ is

$$\mathrm{H}_{\mu}(\boldsymbol{A}|\boldsymbol{B}) = \mathop{\mathbb{E}}_{b \sim \mu_{\boldsymbol{B}}} \left[ \mathrm{H}_{\mu|_{\boldsymbol{B}=b}}(\boldsymbol{A}) \right].$$

We sometimes abuse notation by writing $\mathrm{H}_{\mu}(\boldsymbol{A}|\mathcal{E})$ to denote $\mathrm{H}_{\mu|_{\mathcal{E}}}(\boldsymbol{A})$ (here, $\mathcal{E}$ is an event, not a random variable).

We rely on the following basic properties of the entropy:

1. Entropy is non-negative: $\mathrm{H}(\boldsymbol{A}) \geq 0$.
2. Conditioning does not increase entropy: $\mathrm{H}(\boldsymbol{A}|\boldsymbol{B}) \leq \mathrm{H}(\boldsymbol{A})$.
3. The chain rule for entropy: $\mathrm{H}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m) = \sum_{i=1}^{m} \mathrm{H}(\boldsymbol{A}_i | \boldsymbol{A}_{i-1}, \ldots, \boldsymbol{A}_1)$.
4. Subadditivity: $\mathrm{H}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m) \leq \sum_{i=1}^{m} \mathrm{H}(\boldsymbol{A}_i)$, with equality iff $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$ are independent.

For $p \in [0,1]$, we use $\mathrm{H}(p)$ as short-hand notation for the Shannon entropy of a Bernoulli random variable with probability $p$ of being 1. In our lower bound, we use the following fact:

▶ **Fact 4.** *Let* $p \in [0,1]$*, Then* $\min\{p, 1-p\} \leq \mathrm{H}(p)$*.*

To measure the amount of information a protocol reveals about its inputs, we use *mutual information.* The *mutual information* between random variables $\boldsymbol{A}$ and $\boldsymbol{B}$ is given by $\mathsf{I}(\boldsymbol{A}\,;\,\boldsymbol{B}) := H(\boldsymbol{A}) - H(\boldsymbol{A} \mid \boldsymbol{B})$. For random variables $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, the *conditional mutual information* between $\boldsymbol{A}$ and $\boldsymbol{B}$ given $\boldsymbol{C}$ is $\mathsf{I}(\boldsymbol{A}\,;\,\boldsymbol{B} \mid \boldsymbol{C}) := H(\boldsymbol{A} \mid \boldsymbol{C}) - H(\boldsymbol{A} \mid \boldsymbol{B}, \boldsymbol{C})$.

The following Lemma will be useful in our lower bound:

▶ **Lemma 5.** *Let* $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}$ *be random variables, such that* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are independent. Then* $\mathsf{I}(\boldsymbol{A}\,;\,\boldsymbol{\Pi} \mid \boldsymbol{B}) = \mathsf{I}(\boldsymbol{A}\,;\,\boldsymbol{\Pi}) + \mathsf{I}(\boldsymbol{A}\,;\,\boldsymbol{B} \mid \boldsymbol{\Pi})$*.*

For an event $\mathcal{E}$, we sometimes abuse notation and denote $\mathsf{I}(\boldsymbol{A}\,;\,\boldsymbol{B} \mid \mathcal{E}) := \mathsf{I}(\boldsymbol{A}|_{\mathcal{E}}\,;\,\boldsymbol{B}|_{\mathcal{E}})$.

To measure the difference between two distributions, we use KL divergence:

▶ **Definition 6** (KL divergence)**.** *For two distributions* $\mu$,$\mu'$ *supported over a set* $\chi$*, the* KL divergence *of* $\mu$ *from* $\mu'$ *is:*

$$D\left(\mu \,||\, \mu'\right) := \sum_{x \in \chi} \mu(x) \log \frac{\mu(x)}{\mu'(x)}.$$

We sometimes use $D\left(p \,||\, p'\right)$ as short-hand notation for the divergence between the Bernoulli distributions with probability $p$ and $p'$ (resp.) of being 1.

KL divergence has the following monotonicity property, which will be useful in our upper bound:

▶ **Lemma 7.** *Let* $0 < p < q \leq a < 1/100$ *be constants, then* $D\left(p+a \,||\, p\right) \geq D\left(q+a \,||\, q\right)$*.*

The proof of Lemma 7 will appear in the full version of this paper.

Our lower bound also uses *Pinsker's inequality*, which asserts that for any $p, p' \in (0,1)$ we have $|p - p'| \leq \sqrt{D\left(p \,||\, p'\right) \ln 2/2}$.

The mutual information between two variables $\boldsymbol{A}, \boldsymbol{B}$ is equal to the *expected divergence* of $\boldsymbol{A}$'s posterior distribution given $\boldsymbol{B}$, from $\boldsymbol{A}$'s prior distribution (or vice-versa):

▶ **Fact 8.** *For any random variables* $\boldsymbol{A}, \boldsymbol{B}$ *we have* $\mathsf{I}(\boldsymbol{A}\,;\,\boldsymbol{B}) = \mathbb{E}_{b \sim \boldsymbol{B}}\left[D\left(\boldsymbol{A}|_{\boldsymbol{B}=b} \,||\, \boldsymbol{A}\right)\right]$*.*

The following technical lemmas will be useful in our lower bound. The first bounds the "difference" between two Bernoulli random variables, in terms of their KL divergence:

▶ **Lemma 9.** *Let* $p, q$ *be constants in* $(0,1)$*, and let* $\alpha \in (0, 1/2)$*, such that* $D\left(q \,||\, p\right) < p\alpha^2/(4\ln 2)$*. Then we have* $q/p \in ((1-\alpha)p, (1+\alpha)p)$*.*

In Section 4.2 we use the tight version of the additive Chernoff bound, which is stated in terms of KL divergence: for a sum $\boldsymbol{Y} = \sum_{i=1}^{m} \boldsymbol{Y}_i$ of iid Bernoulli random variables with $\mathbb{E}\left[\boldsymbol{Y}_i\right] = p$, we have

$$\Pr\left[\sum_{i=1}^{n} \boldsymbol{Y}_i/n \geq p + \epsilon\right] \leq e^{-D(p+\epsilon \,||\, p)}, \qquad \text{and} \qquad \Pr\left[\sum_{i=1}^{n} \boldsymbol{Y}_i/n \leq p - \epsilon\right] \leq e^{-D(p-\epsilon \,||\, p)}.$$

# 4 Upper Bounds

In this section we give three upper bounds. The first two address the case where the number of parties is $k \le \log n$: we first give a protocol with expected communication $\tilde{O}(n^{1-1/k}\,\mathbb{E}\left[\boldsymbol{S}\right]^{1/k})$, which can also handle input distributions that are known only approximately, and then build on it to construct a protocol with expected communication $\tilde{O}(n^{1-1/k}\mathrm{H}\left(\boldsymbol{S}\right)^{1/k})$, replacing the expected size of the intersection with its entropy. These two protocols can be used in either the shared blackboard model or the coordinator model, since one model can simulate the other with multiplicative cost at most $O(k) = O(\log n)$.

The third protocol is for the case where $k > \log n$, in the coordinator model, which is the harder of the two models when $k$ is large. This final protocol relies on advance access to only a single sample from the input distribution, and computes the intersection with expected communication $\tilde{O}(n + k)$.

**Approximate knowledge of a distribution.** As explained above, some of our protocols assume that the players do not exactly know the underlying input distribution, and instead are provided advanced access to samples from the distribution. We use these samples to approximate the marginal distribution of every bit $\boldsymbol{X}_i^j$ in the input. It is crucial to allow both multiplicative and additive approximation error, as allowing only one type of error would make it costlier to obtain the approximation (in terms of the number of samples required; see Section 4.2 below).

▶ **Definition 10.** *Let $\epsilon \ge 0, \alpha \in [0,1)$ and let $b \in [0,1]$. We say that a value $a \in [0,1]$ is an* $(\alpha,\epsilon)$-approximation *of $b$ if $(1-\alpha)a - \epsilon \le b \le (1+\alpha)a + \epsilon$ and also $(1-\alpha)(1-a) - \epsilon \le 1 - b \le (1+\alpha)(1-a) + \epsilon$.*

We extend this definition to a distribution $\mu$ over $\{0,1\}^{n\times k}$ by saying that a collection of values $\left(a_i^\ell\right)_{i\in[n],\ell\in[k]} \subseteq [0,1]^{n\times k}$ is an $(\alpha,\epsilon)$*-approximation for the marginals of* $\mu$ if $a_i^\ell$ is an $(\alpha,\epsilon)$-approximation of the marginal $\mu_i^\ell$ for every $i \in [n]$ and $\ell \in [k]$.

## 4.1 Basic Protocol for Computing Intersections ($k \le \log n$)

In this section we give our protocol for computing the intersection of the players' inputs assuming approximate knowledge of the marginals of the input distribution. We assume that the number of players is $k \le \log n$.

▶ **Theorem 11.** *Suppose all players know values $\left(a_i^\ell\right)_{i\in[n],\ell\in[k]}$ that* $(\alpha,\epsilon)$-approximate *the marginals of a product distribution $\mu$. Then there is a zero-error two-round protocol in the coordinator model for finding the intersection, with expected communication cost*

$$O\left(\left(\frac{1+\alpha}{1-\alpha}\right)\left(kn^{1-1/k}\,\mathbb{E}\left[|\boldsymbol{S}|\right]^{1/k} + 2\epsilon kn\right)\log n + k\right),$$

*where the expectation is over the input distribution $\mu$.*

From here on, we will refer to this protocol as $\Pi_{\text{base}}$.

**High-level overview.** We begin with a high-level overview of $\Pi_{\text{base}}$, assuming for simplicity that the players know the true marginals $\left(\mu_i^\ell\right)_{i\in[n],\ell\in[k]}$ of the input distribution.

For each coordinate $i \in [n]$, the protocol checks whether $i$ is in the intersection using one of the following two strategies:

- "The 1-strategy": this strategy is appropriate for coordinates $i$ where some player $\ell$ has a very small probability that $\boldsymbol{X}_i^\ell = 1$. In this case we can make good progress at little expected cost by asking player $\ell$ to speak up only if it has $\boldsymbol{X}_i^\ell = 1$: with good probability, player $\ell$ says nothing, and we learn that coordinate $i$ is not in the intersection.

  Concretely, we find the player $\ell$ that has the smallest probability that $\boldsymbol{X}_i^\ell = 1$, that is, the smallest value of $\mu_i^\ell$ (breaking ties arbitrarily), and ask this player to send index $i$ iff $\boldsymbol{X}_i^\ell = 1$. If player $\ell$ did not send index $i$, we learn that $\boldsymbol{X}_i^\ell = 0$, and therefore coordinate $i$ is not in the intersection. However, if player $i$ did send index $i$, then $\boldsymbol{X}_i^\ell = 1$, and coordinate $i$ might be in the intersection; to check, we simply ask all players $\ell' \neq \ell$ to send $\boldsymbol{X}_i^{\ell'}$, and then we check if they all sent 1.

  The expected communication cost of this strategy is at most $k \log n \min_{\ell\in[k]} \mu_i^\ell$: with probability $1 - \mu_i^\ell$ we have $\boldsymbol{X}_i^\ell = 0$, and in this case no bits are sent.[2] With probability $\min_{\ell\in[k]} \mu_i^\ell$, the player that has the minimum $\mu_i^\ell$ sends index $i$, and the other players $\ell'$ follow suit by announcing $\boldsymbol{X}_i^{\ell'}$, for a total cost of at most $k \log n$ bits.
- "The 0-strategy": this strategy is appropriate for coordinates $i$ where all players $\ell \in [k]$ are fairly likely to have $\boldsymbol{X}_i^\ell = 1$ (i.e., $\mu_i^\ell$ is fairly large). In this case, we ask each player $\ell$ to announce index $i$ iff $\boldsymbol{X}_i^\ell = 0$, and we then know that $i$ is in the intersection iff no player sent index $i$.

  The expected communication cost of this strategy is $\log n \cdot \sum_{\ell\in[k]} \left(1 - \mu_i^\ell\right)$.

To choose which strategy to pursue for a given coordinate $i$, we simply compare the expected cost of the two strategies, and choose the strategy with the smaller expected cost; however, since we do not have access to the true marginals $\left(\mu_i^\ell\right)_{i\in[n],\ell\in[k]}$, we use the estimates $\left(a_i^\ell\right)_{i\in[n],\ell\in[k]}$ in their place. Thus, we estimate the cost of the 1-strategy to be $k \min_{\ell\in[k]} a_i^\ell$, and the cost of the 0-strategy to be $\sum_{\ell\in[k]}(1 - a_i^\ell)$ (ignoring the $\log n$ factor), and we choose to follow the 1-strategy for coordinate $i$ iff $k \min_{\ell\in[k]} a_i^\ell < \sum_{\ell\in[k]}(1 - a_i^\ell)$.

We remark that this protocol generalizes a protocol for set disjointness that appeared in [12], but in [12] only the 1-strategy was required, because we could assume that no single coordinate had high probability of being in the intersection – otherwise we could simply guess that the intersection is not empty.

**Detailed description of the protocol.** The players first partition the coordinates into two sets, $I_1$ (for the 1-strategy) and $I_0$ (for the 0-strategy), defined as follows:

$$I_1 := \left\{ i \in [n] \;\middle|\; k \min_{\ell\in[k]} a_i^\ell \leq \sum_{\ell\in[k]} (1 - a_i^\ell) \right\}, \qquad \text{and} \qquad I_0 := [n] \setminus I_1.$$

Note that this is done with no communication, as all players know $(a_i^\ell)_{i\in[n],\ell\in[k]}$.

Next, for any $i \in I_1$, let $owner(i)$ be the player $\ell$ believed to be most likely to have $i \notin X_i^\ell$ (if there are several such players, we choose the first one):

$$owner(i) := \min \left\{ \ell \in [k] \;\middle|\; a_i^\ell = \min_{m\in[k]} a_i^m \right\}.$$

[2] Technically, players are not allowed to convey information by staying silent. In our implementation below, this is handled by having the players announce all their indices as a set, rather than going over the coordinates one-by-one as we do in our informal overview here. The sets are encoded using a variable-length encoding, and "no bits are sent for coordinate $i$" technically means that index $i$ does not appear in any player's set.

We partition $I_1$ into subsets $I_1^1, \ldots, I_1^k$ by owner, with $I_1^\ell := \{i \in I_1 \mid \mathit{owner}(i) = \ell\}$ for each $i \in [k]$. The protocol proceeds as follows.

1. Each player $\ell \in [k]$ announces $\overline{X^\ell} \cap I_0$ and $X^\ell \cap I_1^\ell$.
2. The coordinator can now deduce the intersection in the $I_0$ coordinates, as it holds that $\cup_{\ell \in [k]} \left(\overline{X^\ell} \cap I_0\right) = I_0 \setminus \cap_{\ell \in [k]} X^\ell$. The coordinator also sets $T := \cup_{\ell \in [k]} \left(X^\ell \cap I_1^\ell\right)$, and announces $T$ to all players.
3. Each player $\ell \in [k]$ sends $X^\ell \cap T$ to the coordinator. The coordinator declares that the intersection in $I_1$ is $\left(\cap_{\ell \in [k]} X^\ell\right) \cap T$.

**Expected communication cost.** We prove a tighter bound than the one claimed in Theorem 11, as the tighter bound will be useful to us in Section 4.3:

▷ Claim 12. When executed with an $(\alpha, \epsilon)$-approximation of the marginals of $\mu$, the expected communication cost of $\Pi_{\text{base}}$ is

$$O\left(\left(\frac{1+\alpha}{1-\alpha}\right)\left(k\sum_{i=1}^{n} \min\left\{\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}, \left(1-\mathbb{E}\left[\boldsymbol{S}_i\right]\right)^{1/k}\right\} + 2\epsilon k n\right)\log n + k\right). \tag{1}$$

To obtain Theorem 11 from the claim, we apply Hölder's inequality to the inner sum:

$$\sum_{i=1}^{n} \min\left\{\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}, \left(1-\mathbb{E}\left[\boldsymbol{S}_i\right]\right)^{1/k}\right\} \le \sum_{i=1}^{n} \mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k} \le n^{1-1/k}\, \mathbb{E}\left[|\boldsymbol{S}|\right]^{1/k}.$$

Plugging this into (1) yields the theorem.

The proof of Claim 12 is given in the full version of this paper, but we give a sketch here. We begin by considering an idealized version of the protocol, where the coordinates are partitioned into subsets $J_0, J_1$ based on their true marginals (which are not known the players):

$$J_1 := \left\{ i \in [n] \;\middle|\; k \min_{\ell \in [k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right] \le \sum_{\ell \in [k]} \left(1 - \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]\right) \right\} \qquad \text{and} \qquad J_0 := [n] \setminus J_1.$$

For each $i \in J_1$, the idealized protocol follows the 1-strategy, paying $k \min_{\ell \in [k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]$ in expected communication; for each $i \in J_0$, the idealized protocol follows the 0-strategy, paying $\sum_{\ell \in [k]} \left(1 - \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]\right)$ in expected communication.

Due to the way in which $J_0, J_1$ are defined, we are able to show that the idealized protocol pays an expected cost per coordinate of at most $k \min\left\{\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}, \left(1-\mathbb{E}\left[\boldsymbol{S}_i\right]\right)^{1/k}\right\}$:

▶ **Lemma 13.** *Let $i \in [n]$. If $i \in J_0$, then*

$$\sum_{\ell \in [k]} \mathbb{E}\left[1 - \boldsymbol{X}_i^\ell\right] \le k \min\left\{E\left[\boldsymbol{S}_i\right]^{1/k}, \left(1-\mathbb{E}\left[\boldsymbol{S}_i\right]\right)^{1/k}\right\},$$

*and if $i \in J_1$, then*

$$k \min_{\ell \in [k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right] \le k \min\left\{E\left[\boldsymbol{S}_i\right]^{1/k}, \left(1-\mathbb{E}\left[\boldsymbol{S}_i\right]\right)^{1/k}\right\}.$$

**Proof.** Fix a coordinate $i \in [n]$. Observe that since $\mu$ is a product distribution,

$$\min_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right] \le \left(\prod_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]\right)^{1/k} = \left(\mathbb{E}\left[\prod_{\ell\in[k]} \boldsymbol{X}_i^\ell\right]\right)^{1/k} = \mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}.$$

Hence if $\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k} \le (1-\mathbb{E}\left[\boldsymbol{S}_i\right])^{1/k}$ and $i \in J_1$, then we have:

$$\min_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right] \le \mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k} = \min\left\{E\left[\boldsymbol{S}_i\right]^{1/k}, (1-\mathbb{E}\left[\boldsymbol{S}_i\right])^{1/k}\right\}.$$

Similarly, if $\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k} \le (1-\mathbb{E}\left[\boldsymbol{S}_i\right])^{1/k}$ and $i \in J_0$, then we have:

$$\sum_{\ell\in[k]} \mathbb{E}\left[1-\boldsymbol{X}_i^\ell\right] < k \min_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right] \le k\,\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k} = k \min\left\{\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}, (1-\mathbb{E}\left[\boldsymbol{S}_i\right])^{1/k}\right\}.$$

Now, if $(1-\mathbb{E}\left[\boldsymbol{S}_i\right])^{1/k} < \mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}$, i.e., $\mathbb{E}\left[\boldsymbol{S}_i\right] > 1/2$, then observe that this implies that $i \in J_0$, as we have that:

$$1/2 < \mathbb{E}\left[\boldsymbol{S}_i\right] = \mathbb{E}\left[\prod_{\ell\in[k]} \boldsymbol{X}_i^\ell\right] = \prod_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right] \le \min_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right],$$

where the last inequality is since all the expectations are upper bounded by 1. This in turn implies that:

$$\sum_{\ell\in[k]} \mathbb{E}\left[1-\boldsymbol{X}_i^\ell\right] \le \frac{k}{2} < k \min_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right],$$

and hence $i \in J_0$. Now we have that:

$$\begin{aligned}
&\sum_{\ell\in[k]} \mathbb{E}\left[1-\boldsymbol{X}_i^\ell\right] \\
&= k\left(1-(1/k)\sum_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]\right) \\
&\le k\left(1-\left(\prod_{\ell\in[k]} \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]\right)^{1/k}\right) && \text{(AM-GM inequality.)} \\
&= k\left(1-\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}\right).
\end{aligned}$$

Finally, observe that

$$1-\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k} \le 1-\mathbb{E}\left[\boldsymbol{S}_i\right] \le (1-\mathbb{E}\left[\boldsymbol{S}_i\right])^{1/k} = \min\left\{\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}, (1-\mathbb{E}\left[\boldsymbol{S}_i\right])^{1/k}\right\}. \quad ◀$$

Since the idealized protocol pays $k \min\left\{\mathbb{E}\left[\boldsymbol{S}_i\right]^{1/k}, (1-\mathbb{E}\left[\boldsymbol{S}_i\right])^{1/k}\right\}$ per coordinate $i$, its total cost matches the bound from Claim 12 with $\alpha = \epsilon = 0$. To prove the claim for the actual protocol, we relate it to the idealized protocol, and show that its communication cost is similar. The key here is to show that even if we "misclassify" a coordinate $i$ by placing it in $I_1$ when the idealized protocol has it in $J_0$ or vice-versa, the penalty is not too large: the partition into $I_1$ vs. $I_0$ is based on the estimates $\left(a_i^\ell\right)_{i\in[n],\ell\in[k]}$, which are close to the true marginals $\left(\mu_i^\ell\right)_{i\in[n],\ell\in[k]}$ on which the partition into $J_1$ vs. $J_0$ is based. If $i \in I_1$ but $i \in J_0$ or vice-versa, then we must be close to the threshold where one strategy becomes preferable to the other, and therefore it does not matter too much which strategy we choose to pursue. For example, consider the case where $i \in I_1$ but $i \in J_0$. Then our actual protocol communicates $k\,\mathbb{E}\left[\boldsymbol{X}_i^{owner(i)}\right]$ bits in expectation for coordinate $i$, while an idealized protocol (where the players use $J_0, J_1$) communicates $\sum_{\ell\in[k]} \mathbb{E}\left[1 - \boldsymbol{X}_i^\ell\right]$ bits in expectation. Now, since $i \in I_1$, we have $k \min_{\ell\in[k]} a_i^\ell \le \sum_{\ell\in[k]}(1 - a_i^\ell)$. Hence:

$$\begin{aligned} k\,\mathbb{E}\left[\boldsymbol{X}_i^{owner(i)}\right] &\le k\left((1+\alpha)a_i^{owner(i)} + \epsilon\right) = (1+\alpha)k\min_{\ell\in[k]} a_i^\ell + \epsilon k \\ &\le (1+\alpha)\sum_{\ell\in[k]}(1 - a_i^\ell) + \epsilon k \\ &\le (1+\alpha)\sum_{\ell\in[k]} \frac{(1 - \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]) + \epsilon}{1-\alpha} + \epsilon k \\ &\le \frac{1+\alpha}{1-\alpha}\left(\sum_{\ell\in[k]} \mathbb{E}\left[1 - \boldsymbol{X}_i^\ell\right] + 2\epsilon k\right). \end{aligned}$$

Finally, showing a similar bound on the other possible types of misclassification, summing over all coordinates and applying Lemma 13 completes the proof for Claim 12.

## 4.2 Approximating the Marginals

In this section we show how to compute an $(1/4, n^{1/k})$-approximation of a value $b \in [0,1]$, given access to $\tilde{\Theta}(n^{1/k})$ iid samples from a Bernoulli distribution with probability $b$ of returning 1. We then apply this procedure to obtain the estimates $\left(a_i^\ell\right)_{i\in[n],\ell\in[k]}$ that are used in the protocol of the previous section.

In general, to obtain an $(\alpha, \epsilon)$-approximation of a value $b \in [0,1]$, it suffices to take $\tilde{\Theta}(1/(\alpha^2\epsilon))$ samples from Bernoulli($b$). In fact, we can provide a stronger guarantee: with high probability, our estimate $\boldsymbol{a}$ is either

- Purely additive: a $(0, \epsilon)$-approximation of $b$, that is, $\boldsymbol{a} - \epsilon \le b \le \boldsymbol{a} + \epsilon$; or,
- Purely multiplicative: an $(\alpha, 0)$-approximation of $b$, that is, $(1-\alpha)\boldsymbol{a} \le b \le (1+\alpha)\boldsymbol{a}$ and $(1-\alpha)(1-\boldsymbol{a}) \le 1 - b \le (1+\alpha)(1-\boldsymbol{a})$.

However, we do not know in advance (or even in hindsight) whether the estimate that we get will be purely additive or purely multiplicative. We note that if we were to insist on always having a purely additive estimate or on always having a purely multiplicative estimate, then we would require significantly more samples. For example, to obtain an purely additive $\pm\epsilon$ approximation of a value $b$ close to 1/2, we would require $\Omega(1/\epsilon^2)$ samples rather than $\Omega(1/\epsilon)$, which is important in our case, since $\epsilon$ is very small (roughly $n^{-1/k}$); to obtain a purely multiplicative $(1 \pm \alpha)$-approximation of a value $b$ close to 0 we require $\Omega(1/b)$ samples (i.e., an unbounded number of samples when $b$ is unknown). Thus, it is important that our protocol can handle the type of estimate that we produce here, which can have both types of approximation error.

Obtaining the estimate is very simple, but the analysis is somewhat delicate:

▶ **Lemma 14.** *For any $\alpha, \epsilon, \delta \in (0,1)$ and $b \in (0,1)$, given access to $O(\frac{1}{\alpha^2\epsilon}\log(1/\delta))$ iid samples of Bernoulli(b), with probability $1-\delta$ we can compute a value $a$ that is either an $(\alpha, 0)$-approximation or a $(0, \epsilon)$-approximation to $b$ (or both).*

**Proof.** Let $m = (100/(\alpha^2\epsilon))\log(4/\delta)$. Given samples $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_m \sim \text{Bernoulli}(b)$, the estimate we output is $\boldsymbol{a} = \sum_{i=1}^{m} \boldsymbol{B}_i/m$. We claim that this estimate $(\alpha, \epsilon)$-approximates $b$ with probability $1 - \delta$. We divide into cases based on the values of $\epsilon$ and $b$.

**Case 1: $\epsilon > \frac{1}{100}$.** In this case we prove that $\boldsymbol{a}$ is a $(0, \epsilon)$-approximation to $b$ with probability $1 - \delta$. By the additive Chernoff bound,

$$\Pr(\boldsymbol{a} > b + \epsilon) \leq e^{-m \cdot D(b+\epsilon \,||\, b)} \leq e^{-m\cdot\epsilon^2} \leq e^{-m\cdot\epsilon/100},$$

where the second step uses Pinsker's inequality, and the last step uses the fact that $\epsilon > 1/100$. Similarly, $\Pr(\boldsymbol{a} < b - \epsilon) \leq e^{-m\cdot D(b-\epsilon \,||\, b)} \leq e^{-m\cdot\epsilon^2} \leq e^{-m\cdot\epsilon/100}$. Since $m \geq (100/\epsilon)\log(4/\delta)$ we have $e^{-m\cdot\epsilon/100} \leq \delta/4$, so the probability that either $\boldsymbol{a} > b + \epsilon$ or $\boldsymbol{a} < b - \epsilon$ is less than $\delta$. In other words, with probability at least $1 - \delta$, we have $\boldsymbol{a} - \epsilon \leq b \leq \boldsymbol{a} + \epsilon$, as required.

**Case 2: $\epsilon < b < 1-\epsilon$.** In this case we prove that $\boldsymbol{a}$ is an $(\alpha, 0)$-approximation to $b$ with probability $1 - \delta$. By the multiplicative Chernoff bound, $\Pr(\boldsymbol{a} \notin (1 \pm \alpha/2)b) \leq 2e^{-(\alpha/2)^2 bm/3} \leq 2e^{-\alpha^2\epsilon m/12}$. Similarly, $\Pr(1 - \boldsymbol{a} \notin (1 \pm (\alpha/2))(1-b)) \leq 2e^{-(\alpha/2)^2(1-b)m/3} \leq 2e^{-\alpha^2\epsilon m/12}$. Since $m = (100/(\alpha^2\epsilon))\log(4/\delta)$, we have $e^{-\alpha^2\epsilon m/3} \leq \delta/4$, and thus the probability that either $\boldsymbol{a} \notin (1\pm\alpha/2)b$ or $1-\boldsymbol{a} \notin (1\pm\alpha/2)(1-b)$ is at most $\delta$. Note that if $(1-\alpha/2)b \leq \boldsymbol{a} \leq (1+\alpha/2)b$, then we also have $b \leq \boldsymbol{a}/(1-\alpha/2) \leq (1+\alpha)\boldsymbol{a}$ and $b \geq \boldsymbol{a}/(1+\alpha/2) \geq (1-\alpha)\boldsymbol{a}$, as required, and similarly for $1 - b$ and $1 - \boldsymbol{a}$.

**Case 3: $b \leq \epsilon \leq 1/100$ or $1 - b \leq \epsilon \leq 1/100$.** In this case we prove that $\boldsymbol{a}$ is a $(0, \epsilon)$-approximation to $b$ with probability $1 - \delta$. Let us assume that $b \leq \epsilon \leq 1/100$, as the other case is symmetric. First, observe that $\Pr(\boldsymbol{a} < b - \epsilon) = 0$, as $b - \epsilon < 0$. For the other side, by the additive Chernoff bound, $\Pr[\boldsymbol{a} > b + \epsilon] \leq e^{-m\cdot D(b+\epsilon \,||\, b)}$. Using Lemma 7 and the fact that $b \leq \epsilon$, we can bound the divergence from below by $D(b + \epsilon \,||\, b) \geq D(2\epsilon \,||\, \epsilon)$; and using a technical lemma from [9] and the fact that $\epsilon \leq 1/100$, we have $D(2\epsilon \,||\, \epsilon) \geq 2\epsilon/10$. All together, we see that $\Pr[\boldsymbol{a} > b + \epsilon] \leq e^{-2\epsilon m/10} \leq \delta$. ◀

Plugging in $\epsilon = n^{-1/k}$ and $\alpha = 1/4$, we see that $O(n^{1/k})$ samples suffice to estimate a single marginal $\mu_i^\ell$ with sufficient accuracy, and $O(n^{1/k}\log(nk))$ samples suffice to approximate the entire distribution.

## 4.3 Entropy-Based Protocol ($k \leq \log n$)

In this section we refer to the protocol of Section 4.1 as *the base protocol*, $\Pi_{\text{base}}$. We show how to build on the base protocol to obtain a better protocol in the case where the intersection has small entropy. For convenience, we describe the new protocol in the shared blackboard model (the protocol can be adapted to the coordinator model with a multiplicative overhead of $O(k) = O(\log n)$ by having the coordinator forward every message to all players).

**High-level overview.** In this overview we assume for simplicity that $\Pr[\boldsymbol{S}_i = 1] \leq 1/2$ for each $i$, which means that $\mathrm{H}(\boldsymbol{S}_i) \geq \mathbb{E}[\boldsymbol{S}_i]$. This suffices to convey the main ideas; in the actual protocol, we work with $\min(\mathbb{E}[\boldsymbol{S}_i], 1 - \mathbb{E}[\boldsymbol{S}_i])$ instead of $\mathbb{E}[\boldsymbol{S}_i]$, and rely on the fact that $\mathrm{H}(p) \geq \min(p, 1-p)$ for every $p \in [0, 1]$.

Our protocol is motivated by the observation that the base protocol from Section 4.1 already has the desired communication cost of $\tilde{O}(n^{1-1/k}\mathrm{H}(\boldsymbol{S})^{1/k})$ in the special case where the intersection bits $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n$ are *independent*: by our assumption that $\mathrm{H}(\boldsymbol{S}_i) \geq \mathbb{E}[\boldsymbol{S}_i]$ for each $i$, we can use Claim 12 and Hölder's inequality to see that the base protocol computes the intersection with expected communication cost $\tilde{O}\left(n^{1-1/k}\left(\sum_{i=1}^{n}\mathrm{H}\left(\boldsymbol{S}_i\right)\right)^{1/k}\right)$. In general, $\sum_{i=1}^{n}\mathrm{H}(\boldsymbol{S}_i)$ can be much greater than $\mathrm{H}(\boldsymbol{S})$ (e.g., if $\boldsymbol{S}_1 = \ldots = \boldsymbol{S}_n$). However, if $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n$ are independent, then $\sum_{i=1}^{n}\mathrm{H}\left(\boldsymbol{S}_i\right) = \mathrm{H}(\boldsymbol{S})$, and we are done.

What should we do when $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n$ are not independent? In this case we show that we can *exploit* the correlation between the bits. Given a set of coordinates $I \subseteq [n]$, let us say that the bits of $\boldsymbol{S}_I$ are *nearly-independent* if

$$\sum_{i \in I}\mathrm{H}\left(\boldsymbol{S}_i\right) \leq 2\mathrm{H}\left(\boldsymbol{S}_I\right). \tag{2}$$

Intuitively, (2) requires that the bits of $\boldsymbol{S}_I$ behave “almost as nicely” as independent bits, in that the sum of their marginal entropies is not much greater than their joint entropy (where for truly independent bits these would be equal).

Our protocol finds a maximal subset of coordinates $I \subseteq [n]$ such that the bits $\boldsymbol{S}_I$ are nearly-independent, and uses the base protocol to compute $\boldsymbol{S}_I$. By (2), the communication cost is $\tilde{O}\left(n^{1-1/k}\left(\sum_{i\in I}\mathrm{H}\left(\boldsymbol{S}_i\right)\right)^{1/k}\right) = \tilde{O}\left(n^{1-1/k}\mathrm{H}\left(\boldsymbol{S}_I\right)^{1/k}\right)$. Each remaining coordinate $j \notin I$ is “somewhat dependent” on $\boldsymbol{S}_I$, otherwise we could add $j$ to $I$ and obtain a larger set, contradicting the maximality of $I$. Intuitively, this means that our uncertainty about $\boldsymbol{S}_j$ should decrease after learning $\boldsymbol{S}_I$, and indeed we can prove that $\mathrm{H}\left(\boldsymbol{S}_j \mid \boldsymbol{S}_I\right) \leq (1/2)\mathrm{H}\left(\boldsymbol{S}_j\right)$ (see Lemma 15 in the next section). We now recurse on the remaining coordinates.

After $O(\log n)$ iterations, for each coordinate $j$ that we have not yet solved, the entropy of $\boldsymbol{S}_j$ is reduced to at most $1/2^{\log n} = 1/n$. We can now afford to simply call the base protocol to solve all the remaining coordinates: if $F \subseteq [n]$ is the set of remaining coordinates, then the cost of solving all of them using the base protocol is roughly $\tilde{O}\left(n^{1-1/k}\left(\sum_{i\in F}\mathrm{H}\left(\boldsymbol{S}_i\right)\right)^{1/k}\right) = \tilde{O}\left(n^{1-1/k}\cdot\left(\sum_{i\in F}(1/n)\right)^{1/k}\right) = \tilde{O}\left(n^{1-1/k}\cdot 1\right)$.

**Detailed description of the protocol.** Throughout the protocol, the players maintain a subset $J \subseteq [n]$ of coordinates in which the intersection was already computed, and a distribution $\mu'$, which is the posterior input distribution given what the protocol has learned so far. All entropies computed during the run of the protocol are with respect to the updated distribution, $\mu'$. The protocol is as follows.

1. Initialize $J \leftarrow \emptyset, \mu' \leftarrow \mu$.
2. Repeat $R = \lceil \log n \rceil$ times, or until $J = [n]$:
   2.1. Let $I \subseteq [n] \setminus J$ be a maximal set of nearly-independent coordinates (see (2) above). If there is more than one possible choice for $I$, we choose the lexicographically-smallest one. This step does not require communication.
   2.2. Execute the base protocol $\Pi_{\text{base}}$ on the coordinates of $I$, using the distribution $\mu'$. Let $\tau$ be the transcript of $\Pi_{\text{base}}$, and let $\mu'|_\tau$ be the distribution $\mu'$ conditioned on the event that the transcript of $\Pi_{\text{base}}$ is $\tau$.
   2.3. Update $J \leftarrow J \cup I, \mu' \leftarrow \mu'|_\tau$.
3. Finally, if $J \neq [n]$, call the protocol $\Pi_{\text{base}}$ on the remaining coordinates $[n] \setminus J$, using the distribution $\mu'$.

At the end, we output all intersection elements found during any of the calls to $\Pi_{\text{base}}$.

**Expected communication cost.** In the analysis we rely on the finer bound given in Claim 12 for the communication cost of $\Pi_{\text{base}}$. The bound is stated in terms of the expectations $\mathbb{E}\left[\boldsymbol{S}_i\right]$, but since $\mathrm{H}(p) \geq \min\{p, 1-p\}$ for every $p \in [0,1]$, Claim 12 and Hölder’s inequality imply that the expected cost of $\Pi_{\text{base}}$ when $\alpha = \epsilon = 0$ is

$$O\left(k\sum_{i=1}^{n}\mathrm{H}(\boldsymbol{S}_i)^{1/k}+k\right)=O\left(kn^{1-1/k}\left(\sum_{i=1}^{n}\mathrm{H}(\boldsymbol{S}_i)\right)^{1/k}\right). \tag{3}$$

Our goal now is essentially to replace the term $\sum_{i=1}^{n}\mathrm{H}(\boldsymbol{S}_i)$ in the bound above by $\mathrm{H}(\boldsymbol{S})$.

The full analysis will be given in the full version of this paper. The main idea is that in every iteration $r \le \lceil \log n \rceil$, if $\boldsymbol{I}_r$ is the set of coordinates on which we call $\Pi_{\text{base}}$ in iteration $r$, then by choice of $\boldsymbol{I}_r$ we have $\sum_{i\in \boldsymbol{I}_r}\mathrm{H}_{\mu_r}(\boldsymbol{S}_i) \le 2\mathrm{H}_{\mu_r}(\boldsymbol{S}_{\boldsymbol{I}_r})$. Note that the expectation here is taken with respect to the distribution $\mu_r$, which is the input distribution conditioned on the transcript up to iteration $r$ (exclusive). Together with (3), this means that the cost of calling $\Pi_{\text{base}}$ on $\boldsymbol{I}_r$ is $O(kn^{1-1/k}\mathrm{H}_{\mu_r}(\boldsymbol{S}_{\boldsymbol{I}_r})^{1/k})$.

When we reach the last step of the protocol, the set of remaining coordinates may not be nearly-independent. However, we claim that for every coordinate $i \in [n]$, given the transcript of the entire protocol so far, the conditional entropy of $\boldsymbol{S}_i$ is reduced to at most $1/n$. This is because in every iteration, the protocol either determines $\boldsymbol{S}_i$, reducing its entropy to zero, or solves a set of coordinates on which $\boldsymbol{S}_i$ depends strongly, which also reduces its entropy.

▶ **Lemma 15.** *Let $\boldsymbol{\Pi}_{<r}$ denote the transcript of the protocol up to iteration $r$, exclusive. For every $i \in [n]$ and iteration $r \le R$, $\mathrm{H}_\mu(\boldsymbol{S}_i \mid \boldsymbol{\Pi}_{<r+1}) \le \frac{1}{2}\mathrm{H}_\mu(\boldsymbol{S}_i \mid \boldsymbol{\Pi}_{<r})$.*

**Proof.** We prove that for every iteration $r \le R$ and transcript $\tau_{<r}$,

$$\mathrm{H}_\mu(\boldsymbol{S}_i \mid \boldsymbol{\Pi}_{<r+1}, \boldsymbol{\Pi}_{<r}=\tau_{<r}) \le \frac{1}{2}\mathrm{H}_\mu(\boldsymbol{S}_i \mid \boldsymbol{\Pi}_{<r}=\tau_{<r}).$$

The lemma then follows by taking the expectation over $\tau_{<r}$.

The transcript $\tau_{<r}$ determines the sets $I_1,\dots,I_r$ on which $\Pi_{\text{base}}$ is called in every iteration $1,\dots,r$, as well as $\boldsymbol{S}_{\boldsymbol{I}_1}=S_{I_1},\dots,\boldsymbol{S}_{\boldsymbol{I}_{r-1}}=S_{I_{r-1}}$. The value of $\boldsymbol{S}_{I_r}$ is not determined by $\tau_{<r}$, but it is determined by $\Pi_{<r+1}$, as it is computed in iteration $r$ itself. Therefore,

$$\mathrm{H}(\boldsymbol{S}_i \mid \boldsymbol{\Pi}_{<r+1},\tau_{<r}) = \mathrm{H}(\boldsymbol{S}_i \mid \boldsymbol{\Pi}_{<r+1},\tau_{<r},\boldsymbol{S}_{I_1},\dots,\boldsymbol{S}_{I_r}) \le \mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r},\boldsymbol{S}_{I_1},\dots,\boldsymbol{S}_{I_r}),$$

where the last step uses the fact that conditioning does not increase entropy.

If there is some iteration $t \le r$ such that $i \in I_t$, then clearly $\mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r},\boldsymbol{S}_{I_1},\dots,\boldsymbol{S}_{I_r})=0$, and the lemma follows from the non-negativity of entropy. Otherwise, $i$ is not an element of any set $I_1,\dots,I_r$, and in particular $i \notin I_r$. We claim that $\mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r},\boldsymbol{S}_{I_r}) \le (1/2)\mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r})$, which proves the claim, as $\mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r},\boldsymbol{S}_{I_r}) = \mathrm{H}\left(\boldsymbol{S}_i \,\middle|\, \tau_{<r},\boldsymbol{S}_{I_r},\boldsymbol{S}_{I_1},\dots,\boldsymbol{S}_{I_{r-1}}\right)$ and similarly $\mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r}) = \mathrm{H}\left(\boldsymbol{S}_i \,\middle|\, \tau_{<r},\boldsymbol{S}_{I_1},\dots,\boldsymbol{S}_{I_{r-1}}\right)$ (as $\boldsymbol{S}_{I_1},\dots,\boldsymbol{S}_{I_{r-1}}$ are all determined by $\tau_{<r}$).

Suppose for the sake of contradiction that

$$\mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r},\boldsymbol{S}_{I_r}) > (1/2)\mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r}). \tag{4}$$

Then we can write

$$\begin{aligned}\sum_{j\in I_r\cup\{i\}}\mathrm{H}(\boldsymbol{S}_j \mid \tau_{<r}) &= \mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r}) + \sum_{j\in I_r}\mathrm{H}(\boldsymbol{S}_j \mid \tau_{<r})\\ &\le 2\mathrm{H}(\boldsymbol{S}_i \mid \tau_{<r},\boldsymbol{S}_{I_r}) + 2\mathrm{H}(\boldsymbol{S}_{I_r} \mid \tau_{<r}) \qquad \text{(by (4) and by choice of } I_r)\\ &= 2\mathrm{H}\left(\boldsymbol{S}_{I_r\cup\{i\}} \,\middle|\, \tau_{<r}\right),\end{aligned}$$

which contradicts the maximality of $I_r$. ◀

▶ **Corollary 16.** *For every $i \in [n]$ we have $\mathrm{H}_\mu(\boldsymbol{S}_i \mid \boldsymbol{\Pi}_{<R+1}) \le 1/n$.*

By the corollary, we can simply use (3) to bound the cost of calling $\Pi_{\text{base}}$ on all the remaining coordinates by $O\left(kn^{1-1/k} \cdot \left(\sum_{i=1}^{n}(1/n)\right)^{1/k}\right) = O\left(kn^{1-1/k}\right)$. The final step in the proof is to carefully sum the costs of all the iterations, using Hölder's inequality and the chain rule for entropy to obtain the final bound of Theorem 2.

## 4.4 Upper Bound for Large $k$

For the case where we have $k \geq \log n$ players, we show that a single sample from the input distribution suffices to later compute the intersection on new inputs with expected communication cost $\tilde{O}(n+k)$. To do so, we modify a protocol from [12].

**High-level overview.** The protocol from [12] handles coordinates differently based on whether they have a non-negligible probability of being in the intersection or not. Let us say that a coordinate $i$ is *negligible* if $\Pr[\boldsymbol{S}_i = 1] < \delta/n$. For negligible coordinates $i$, the protocol simply guesses that $\boldsymbol{S}_i = 0$, without trying to actually compute $\boldsymbol{S}_i$. By union bound, this contributes a total of at most $O(\delta)$ to our error probability. For non-negligible coordinates $i$, it is observed in [12] that since $\mu$ is a product distribution, the expected number of players $\ell$ that have $\boldsymbol{X}_i^\ell = 0$ must be very small; otherwise, $\Pr[\boldsymbol{S}_i = 1] = \prod_{\ell\in[k]} \Pr\left[\boldsymbol{X}_i^\ell = 1\right]$ would be very small, but we assumed that $\Pr[\boldsymbol{S}_i = 1] \geq \delta/n$. This means we can afford to have every player $\ell$ that has $\boldsymbol{X}_i^\ell = 0$ announce this fact to the coordinator, who then determines that $\boldsymbol{S}_i = 1$ iff no player $\ell$ announced that $\boldsymbol{X}_i^\ell = 0$.

In our setting we do not know the input distribution exactly, which can lead to two types of mistakes:

- Classifying a coordinate $i$ as negligible when it is in fact non-negligible: we cannot afford to make even one such mistake, because for such coordinates we always output $\boldsymbol{S}_i = 0$, even though there is non-negligible probability that $\boldsymbol{S}_i = 1$. Thus, when we classify a coordinate as negligible, it must truly be negligible under the unknown input distribution.
- Classifying a coordinate $i$ as non-negligible when it is in fact negligible: this type of mistake does not lead to incorrect outputs, but it can increase our expected communication cost, depending on the expected players that have 0 in coordinate $i$. Unlike the previous case, here we can afford to make some mistakes, but we should avoid classifying a coordinate $i$ as non-negligible if $\sum_{i=1}^{k} \mathbb{E}\left[1 - \boldsymbol{X}_i^\ell\right]$ is large.

We show that when $k \geq \log n$, a single sample from the input distribution suffices to classify coordinates well enough for our purposes. Let $z_i = \sum_{\ell\in[k]}\left(1 - \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]\right)$ be the expected number of zeroes in coordinate $i \in [n]$, under the unknown input distribution. Given one sample $\boldsymbol{A} \sim \mu$, we estimate $z_i$ by $v_i := \sum_{\ell\in[k]}\left(1 - \boldsymbol{A}_i^\ell\right)$. Since $k \geq \log n$, the value $\sum_{\ell\in[k]}\left(1 - \boldsymbol{A}_i^\ell\right)$ is concentrated about its mean, which is $z_i$. This allows us to simultaneously estimate $z_1, \ldots, z_n$ with enough precision that no non-negligible coordinate is classified as negligible, and at the same time, every coordinate $i$ that is classified as non-negligible has small $z_i$.

**Detailed description of the protocol.** As outlined above, we first take a sample $\boldsymbol{A} \sim \mu$, compute the estimates $v_i := \sum_{\ell\in[k]}\left(1 - \boldsymbol{A}_i^\ell\right)$, and then choose the following set of coordinates $N \subseteq [n]$ to classify as non-negligible: $N = \{i \in [n] : v_i \leq \beta \ln(n/\delta)\}$, where $\beta \geq 1$ is a constant whose value will be fixed later.

The protocol then proceeds as follows: given input $\boldsymbol{X} \sim \mu$,

1. For each coordinate $i \notin N$, the coordinator declares that $\boldsymbol{S}_i = 0$, that is, $i \notin \bigcap_{\ell\in[k]} \boldsymbol{X}^\ell$.
2. Simultaneously, each player $\ell$ sends $\boldsymbol{N}^\ell := N \cap \left([n] \setminus \boldsymbol{X}^\ell\right)$ to the coordinator.
3. The coordinator outputs $N \setminus \bigcup_{\ell\in[k]} \boldsymbol{N}^\ell$ as its estimate for the intersection $\boldsymbol{S}$.

**Correctness and expected communication cost.** Let $z_i = \sum_{\ell \in [k]} \left(1 - \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]\right)$ be the expected number of zeroes in coordinate $i \in [n]$, and let $\mathcal{E}$ be the event that for every coordinate $i \in [n]$,

- If $z_i < \ln(n/\delta)$ then $i \in N$ (that is, $v_i \leq \beta \ln(n/\delta)$), and
- If $z_i > \beta^2 \ln(n/\delta)$ then $i \notin N$ (that is, $v_i > \beta \ln(n/\delta)$).

Intuitively, $\mathcal{E}$ is the event that we have classified every coordinate "well enough". Using Chernoff, it is easy to see that when $\beta$ is large enough (say, $\beta \geq 8$), the event $\mathcal{E}$ occurs with probability $\geq 1 - \delta$ over the sample $\boldsymbol{A} \sim \mu$. This implies the correctness of the protocol: whenever $\mathcal{E}$ occurs, every coordinate $i \in [n]$ where $z_i < \beta \ln(n/\delta)$ is identified as non-negligible, $i \in N$. The coordinates in $N$ are correctly solved by the protocol, since every player that has a zero in such a coordinate informs the coordinator. As for coordinates $i \notin N$, these coordinates must have $z_i \geq \beta \ln(n/\delta)$. By Lemma 4 in [12], this implies that $\Pr[\boldsymbol{S}_i = 1] \leq \delta/n$. By union bound, the probability that any such coordinate is in the intersection is at most $\delta$. The error probability of the protocol is therefore bounded by $\Pr\left[\overline{\mathcal{E}}\right] + \Pr[\mathcal{E}] \cdot \delta \leq 2\delta$.

To bound the expected communication, we again condition on the event $\mathcal{E}$, which implies that every coordinate $i \in N$ that we actually solve has $z_i \leq \beta^2 \ln(n/\delta)$. Since $\log n \cdot z_i = \log n \sum_{\ell \in [k]} \left(1 - \mathbb{E}\left[\boldsymbol{X}_i^\ell\right]\right)$ is the expected communication cost of solving $i$, this means we send an expected $O(\log(n) \log(n/\delta))$ bits per coordinate in $N$, for a total of $O(n \log(n) \log(n/\delta))$.

## 5 Lower Bounds

We begin by observing that when we know nothing about the input distribution other than the fact that it is a product distribution, the distributional communication complexity of computing any function $f$ is the same as the worst-case cost. This justifies our need for taking samples from the distribution before constructing the protocols of Sections 4.1, 4.4.

▶ **Observation 17.** *Let $f$ be a function over $\{0,1\}^{n \times k}$, and let $C$ be the worst-case randomized communication complexity*[3] *of $f$ with error probability $1/3$ (on any input). Let $\Pi$ be a (possibly randomized) protocol for computing $f$, such that under* any product distribution *$\mu$ over $\{0,1\}^{n \times k}$ we have $\Pr_{\boldsymbol{X} \sim \mu}[\Pi$ correctly computes $f(\boldsymbol{X})] \geq 2/3$. Then there is a product distribution $\mu$ such that the expected communication cost of $\Pi$ under $\mu$ is $C$.*

**Proof.** For each input $X \in \{0,1\}^{n \times k}$, let $\mu_X$ be the product distribution $\mu_X$ that assigns to each player $\ell$ the input $X^\ell$ (this is a deterministic assignment, but it still qualifies as a product distribution). Because $\mu_X$ is a product distribution, and $\Pi$ can handle any product distribution, when we run $\Pi$ on inputs drawn from $\mu_X$ it has success probability at least $2/3$, which means that it correctly computes $f(X)$ with probability at least $2/3$. Thus, $\Pi$ is in fact a protocol for computing $f$ with worst-case error probability at most $1/3$ on any input. Therefore there must exist some input $X$, and hence some product distribution $\mu_X$, on which $\Pi$ sends $C$ bits in expectation. ◀

**Lower bound on the expected communication cost.** To prove the lower bound of Theorem 3, we follow the information-theoretic lower bound technique of [12]. We note that the common information-theoretic strategy of using a *direct sum* argument, where we lower-bound

[3] The randomized communication complexity of a function $f$ is the minimum over all protocols that compute $f$ with worst-case error $\leq 1/3$ of the expected number of bits sent in the worst case (i.e., on any input) by the protocol. The error probability and the expectation are taken with respect to the protocol's internal randomness.

the cost of solving each bit of the problem correctly and then sum over the costs, cannot be used in this context: it yields lower bounds on the cost of solving each coordinate with small marginal error, but as we explained in Section 1, computing each bit $\boldsymbol{S}_i$ of the intersection with small marginal error is easy when the marginal probabilities that $\boldsymbol{S}_i = 1$ tend to be small:

▶ **Proposition 18.** *Let $\alpha \in (0,1)$, $\epsilon \in (0,1]$ be constants, $n > (1/\epsilon)^{2/\alpha}$, $k \geq 2$, and let $\mu$ be a product distribution over $\left(\{0,1\}^n\right)^k$, with expected intersection size $s \leq n^{1-\alpha}$. Then there exists a deterministic protocol that reveals the intersection to an external observer, with per-coordinate error at most $\epsilon$ and expected communication cost at most $\tilde{O}\left(n^{(1-\alpha/2)(1-1/k)}(s+1)^{1/k}\right)$.*

**Proof.** First, observe that the average coordinate $i \in [n]$ has expected intersection size $s/n = n^{-\alpha}$. Denote by $I$ the set of coordinates $i \in [n]$ that have expected intersection size $\mathbb{E}[\boldsymbol{S}_i] > n^{-\alpha/2}$. Note that by Markov's inequality $|I| \leq n^{-\alpha/2} \cdot n = n^{1-\alpha/2}$. Now, if $k \leq \log n$, then the players execute the basic protocol of Theorem 11 on the coordinates of $I$. Note that the protocol reveals the intersection in coordinates $I$ to an external observer with zero error. Since the overall expected intersection size for the coordinates is at most $s$, the protocol has expected communication cost $\tilde{O}\left(n^{(1-\alpha/2)(1-1/k)}(s+1)^{1/k}\right)$.

Similarly, if $k > \log n$, the players execute our protocol for $k > \log n$ (described at 4.4) on the coordinates of $I$. Note that the protocol reveals the intersection to an external observer with per-coordinate error at most $\epsilon/n^{1-\alpha/2} < \epsilon$ and expected communication cost at most $\tilde{O}\left(n^{(1-\alpha/2)(1-1/k)} + k\right)$.

For any remaining coordinate $i \notin I$, the external observer simply declares that $i \notin \cap_{\ell\in[k]} X^\ell$, and has a per-coordinate error at most $n^{-\alpha/2} < \epsilon$. ◀

Fix an expected intersection size $s$. Our lower bound uses the distribution where each bit of the input has iid probability $(s/n)^{1/k}$ of being 1 (that is, the distribution is also a product distribution over the elements, not just the players). It is not hard to see that this yields the desired expected intersection size of $s$, and also that the entropy of the intersection is $\tilde{\Theta}(s)$.

Now suppose we are given a protocol $\Pi$ that sends $o\left(n^{1-1/k}s^{1/k}\right)$ bits in expectation. Following [12], we first show that for the average coordinate $i \in [n]$ and transcript $\tau$ of the protocol, for each player $\ell$, the distribution of $\boldsymbol{X}_i^\ell$ conditioned on $\boldsymbol{\Pi} = \tau$ is very close to its prior: intuitively, to rule out an event with prior probability $p$, the protocol must spend $\Omega(p)$ of its information budget; in our case the event is $\boldsymbol{X}_i^\ell = 1$, and $p = (s/n)^{1/k}$. The protocol expends $o\left(n^{1-1/k}s^{1/k}/n\right) = o\left((s/n)^{1/k}\right)$ of its total information budget on the average coordinate, so the event $\boldsymbol{X}_i^\ell = 1$ remains roughly as likely as it was originally.

Consider a specific coordinate $i \in [n]$, and assume that for all the coordinates $j < i$, the bits $\boldsymbol{S}_i$ have been computed correctly; we denote this event by $\mathcal{E}_{<i}$. Since the protocol computes the entire intersection correctly w.h.p., the event $\mathcal{E}_{<i}$ has high probability. Assume w.l.o.g. that player 1 is the one that outputs $\boldsymbol{S}_i$ given the transcript $\tau$: given on the transcript $\tau$, the event $\mathcal{E}$, and the event $\boldsymbol{X}_i^1 = 1$, player 1 must decide whether to output $\boldsymbol{S}_i = 0$ or $\boldsymbol{S}_i = 1$ (if $\boldsymbol{X}_i^1 = 0$ then player 1 knows that $\boldsymbol{S}_i = 0$ and does not need to work to learn the answer). However, we can show that even conditioned on $\tau, \mathcal{E}$, and $\boldsymbol{X}_i^1 = 1$, the distribution of $\boldsymbol{S}_i$ is still very close to its prior, and therefore player 1 has roughly the same uncertainty about whether or not $\boldsymbol{S}_i = 1$ as it had originally, $\mathrm{H}(\boldsymbol{S}_i) = \tilde{\Theta}(s/n)$.

After analyzing the uncertainty about the output in each coordinate $i \in [n]$ conditioned on $\tau, \mathcal{E}$ and the event that the player $\ell$ deciding this coordinate has $\boldsymbol{X}_i^\ell = 1$, we carefully "collect" these uncertainties and add them up, to show that the players jointly have too much uncertainty about the entire intersection and cannot output it correctly with sufficiently high probability. We note that unlike [12], in this process we need to handle conditioning on some fairly high-probability events (e.g., the event that $\boldsymbol{X}_i^\ell = 1$ has probability $(s/n)^{1/k}$, which is constant when $s = \Omega(n)$). This has the potential of distorting the distributions we work with by a lot if not handled properly.

## References

1 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

2 László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 337–347, 1986.

3 Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *43rd Symposium on Foundations of Computer Science (FOCS 2002)*, pages 209–218, 2002.

4 Anup Bhattacharya, Sourav Chakraborty, Arijit Ghosh, Gopinath Mishra, and Manaswi Paraashar. Disjointness through the lens of vapnik-chervonenkis dimension: Sparsity and beyond. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020*, volume 176 of *LIPIcs*, pages 23:1–23:15, 2020.

5 Ralph Bottesch, Dmitry Gavinsky, and Hartmut Klauck. Correlation in hard distributions in communication complexity. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, volume 40 of *LIPIcs*, pages 544–572, 2015.

6 Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, pages 668–677, 2013.

7 Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. From information to exact communication. In *STOC*, pages 151–160. ACM, 2013.

8 Mark Braverman and Rotem Oshman. On information complexity in the broadcast model. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015*, pages 355–364, 2015.

9 Mark Braverman and Rotem Oshman. A rounds vs. communication tradeoff for multi-party set disjointness. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 144–155, 2017.

10 Joshua Brody, Amit Chakrabarti, Ranganath Kondapally, David P. Woodruff, and Grigory Yaroslavtsev. Beyond set disjointness: The communication complexity of finding the intersection. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, PODC '14, pages 106–113, 2014.

11 Arkadev Chattopadhyay and Toniann Pitassi. The story of set disjointness. *ACM SIGACT News*, 41(3):59–85, 2010.

12 Nachum Dershowitz, Rotem Oshman, and Tal Roth. The communication complexity of multiparty set disjointness under product distributions. In *STOC*, pages 1194–1207. ACM, 2021.

13 Dmitry Gavinsky. The communication complexity of the inevitable intersection problem. *Chic. J. Theor. Comput. Sci.*, 2020, 2020.

14 Badih Ghazi, Ben Kreuter, Ravi Kumar, Pasin Manurangsi, Jiayu Peng, Evgeny Skvortsov, Yao Wang, and Craig Wright. Multiparty reach and frequency histogram: Private, secure, and practical. *Proc. Priv. Enhancing Technol.*, 2022(1):373–395, 2022. `doi:10.2478/popets-2022-0019`.

15 André Gronemeier. Asymptotically optimal lower bounds on the nih-multi-party information complexity of the and-function and disjointness. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009*, volume 3 of *LIPIcs*, pages 505–516, 2009.

16 Dirk Van Gucht, Ryan Williams, David P. Woodruff, and Qin Zhang. The communication complexity of distributed set-joins with applications to matrix multiplication. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS*, pages 199–212, 2015. doi:10.1145/2745754.2745779.

17 Dawei Huang, Seth Pettie, Yixiang Zhang, and Zhijun Zhang. The communication complexity of set intersection and multiple equality testing. *SIAM J. Comput.*, 50(2):674–717, 2021.

18 Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung. On deploying secure computing: Private intersection-sum-with-cardinality. In *IEEE European Symposium on Security and Privacy, EuroS&P*, pages 370–389. IEEE, 2020. doi:10.1109/EuroSP48549.2020.00031.

19 Ivo Kubjas and Vitaly Skachek. Two-party function computation on the reconciled data. In *55th Annual Allerton Conference on Communication, Control, and Computing*, pages 390–396. IEEE, 2017. doi:10.1109/ALLERTON.2017.8262764.

20 Nan Ma and Prakash Ishwar. Two-terminal distributed source coding with alternating messages for function computation. In *2008 IEEE International Symposium on Information Theory*, pages 51–55. IEEE, 2008.

21 Nan Ma and Prakash Ishwar. Infinite-message distributed source coding for two-terminal interactive computing. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1510–1517. IEEE, 2009.

22 Nan Ma and Prakash Ishwar. Some results on distributed source coding for interactive function computation. *IEEE Transactions on Information Theory*, 57(9):6180–6195, 2011.

23 Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. *SIAM J. Comput.*, 45(1):174–196, 2016.

24 Alexander A Razborov. On the distributional complexity of disjointness. In *International Colloquium on Automata, Languages, and Programming*, pages 249–253, 1990.

25 Mert Saglam and Gábor Tardos. On the communication complexity of sparse set disjointness and exists-equal problems. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, pages 678–687, 2013.

26 Georg Schnitger and Bala Kalyanasundaram. The probabilistic communication complexity of set intersection. In *Proceedings of the Second Annual Conference on Structure in Complexity Theory 1987*, 1987.

27 Alexander A Sherstov. Communication complexity theory: Thirty-five years of set disjointness. In *International Symposium on Mathematical Foundations of Computer Science*, pages 24–43, 2014.

28 Gregory Valiant and Paul Valiant. A CLT and tight lower bounds for estimating entropy. *Electron. Colloquium Comput. Complex.*, TR10-183, 2010. URL: https://eccc.weizmann.ac.il/report/2010/183.

29 Thomas Watson. Communication complexity with small advantage. *Comput. Complex.*, 29(1):2, 2020.

30 David P. Woodruff and Qin Zhang. When distributed computation is communication expensive. In *Distributed Computing: 27th International Symposium, DISC 2013*, pages 16–30, 2013.