Separating k-Player from t-Player One-Way Communication, with Applications to Data Streams

David P. Woodruff
Carnegie Mellon University, Pittsburgh, PA, USA
dwoodruf@andrew.cmu.edu

Guang Yang
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
Conflux, Beijing, China
guang.research@gmail.com

Abstract

In a k-party communication problem, the k players with inputs $x_1, x_2, \ldots, x_k$, respectively, want to evaluate a function $f(x_1, x_2, \ldots, x_k)$ using as little communication as possible. We consider the message-passing model, in which the inputs are partitioned in an arbitrary, possibly worst-case manner, among a smaller number t of players ($t < k$). The t-player communication cost of computing f can only be smaller than the k-player communication cost, since the t players can trivially simulate the k-player protocol. But how much smaller can it be? We study deterministic and randomized protocols in the one-way model, and provide separations for product input distributions, which are optimal for low error probability protocols. We also provide much stronger separations when the input distribution is non-product.

A key application of our results is in proving lower bounds for data stream algorithms. In particular, we give an optimal $\Omega(\varepsilon^{-2} \log(N) \log\log(mM))$ bits of space lower bound for the fundamental problem of $(1 \pm \varepsilon)$-approximating the number $\|x\|_0$ of non-zero entries of an n-dimensional vector x after m updates each of magnitude M, and with success probability $\geq 2/3$, in a strict turnstile stream. Our result matches the best known upper bound when $\varepsilon \geq 1/\mathsf{polylog}(mM)$. It also improves on the prior $\Omega(\varepsilon^{-2} \log(mM))$ lower bound and separates the complexity of approximating L_0 from approximating the p-norm L_p for p bounded away from 0, since the latter has an $O(\varepsilon^{-2} \log(mM))$ bit upper bound.

2012 ACM Subject Classification Theory of computation → Streaming models; Theory of computation → Complexity classes; Theory of computation → Lower bounds and information complexity

Keywords and phrases Communication complexity, multi-player communication, one-way communication, streaming complexity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.97

Category Track A: Algorithms, Complexity and Games

Related Version The full version hosted on arXiv `https://arxiv.org/abs/1905.07135`.

Funding This work was supported in part by the National Natural Science Foundation of China Grants No. 61433014, 61602440, 61761136014, 61872334, 61502449, and the 973 Program of China Grant No. 2016YFB1000201.

Acknowledgements We would like to thank Yuval Ishai and Eyal Kushilevitz for initiating the problem of separating worst-case partition communication complexity from streaming complexity, which was our starting point. We also thank the ICALP referees for very helpful comments which helped us revise our initial submission. D. Woodruff would also like to thank the Chinese Academy of Sciences, as well as the Simons Institute for the Theory of Computing.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 97; pp. 97:1–97:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Consider a k-party communication problem, in which the players have inputs $x_1, x_2, \ldots, x_k$ respectively, and want to compute a function $f(x_1, x_2, \ldots, x_k)$ of their inputs using as little communication as possible. We consider the message-passing model, in which the inputs are partitioned in an arbitrary, possibly worst-case manner among a smaller number t of players. That is, we partition $\{1, 2, \ldots, k\}$ into t subsets $S_1, S_2, \ldots, S_t$ such that $\cup_{i=1}^{t} S_i = \{1, 2, \ldots, k\}$ and $S_i \cap S_j = \emptyset$ for every $1 \leq i < j \leq t$, and let the i-th player P_i hold the sequence of inputs $y_i := \left(x_{i_1}, x_{i_2}, \ldots, x_{i_{|S_i|}}\right)$. We are still interested in computing the original function f. The total communication required must be smaller than in the original k-player setting, since the t players can simulate the protocol involving the original k players. A natural question is: *how much smaller can the communication be?*

There are many communication models that are possible, but our main motivation for looking at this question comes from applications to data streams, see below, and so we are primarily interested in the *one-way number-in-hand* model. In this model, each of the t players can only see its own input. The first player composes a message m_1 based on its input y_1 and sends m_1 to the second player. The second player takes m_1 and its input y_2 to compute a message m_2 for the third player, and so on. The t-th (also the last) player, upon receiving the message m_{t-1} from the $(t-1)$-st player, computes the output of the protocol based on m_{t-1} and its own input y_t. We sometimes abuse notation and refer to the output as m_t. The total communication cost is the maximum of $\sum_{i=1}^{t} |m_i|$, where $|m_i|$ denotes the length of the i-th message and the maximum is taken over all possible inputs $y_1, \ldots, y_t$ (which is a partition of $\{x_1, \ldots, x_k\}$) and all random coin tosses of the players. For streaming applications we are especially interested in $\max_{i \in \{1,\ldots,t\}} |m_i|$.

To explain the connection to data streams, almost all known lower bound arguments on the memory required of a data stream algorithm are proven via communication complexity, or at least can be reformulated using communication complexity. The basic idea is to partition the elements of an input stream contiguously, consisting of say k elements, into a possibly smaller number t of players. Then one argues that if there is a data stream algorithm solving the problem, then the communication problem can be solved by passing the memory contents as messages from player to player. Note that this naturally gives rise to the one-way number-in-hand model. Since the total communication cost is $t \cdot S$, where S is the size of the memory of the streaming algorithm, if the randomized t-player communication complexity of the function f is CC_t, we must have $S \geq CC_t/t$. Many lower bounds in data streams are proven already with two players. However, it is known that for some functions more players are needed to obtain stronger lower bounds, such as for estimating the frequency moments in insertion only streams (see, e.g., [3, 17] and references therein).

One cannot help but ask *how powerful is communication complexity for proving data stream lower bounds*? Another natural question is: *for a given function f, which number t of players should one partition the stream into*? Yet another question is regarding the input distribution – should it be a product distribution for which the inputs to the players are chosen independently, or should the inputs be drawn from a non-product distribution to obtain the best space lower bounds? Since we are interested in the limits of using t players for establishing lower bounds for data stream algorithms, we allow the original k inputs (which correspond to the k elements in a stream) to be partitioned in the worst possible way for a t-player communication protocol, as this will give the strongest possible lower bound.

1.1 Our Results

In this paper we study these communication questions and their connections to data streams.

We first make the simple observation that for non-product input distributions, the communication complexity can be arbitrarily smaller if we partition the k inputs into $t < k$ players. Indeed, consider the k-player set disjointness problem in which the i-th player, $1 \le i \le k$, has a set $S_i \subseteq [n]$, where for notational simplicity we define $[n] := \{1, 2, \ldots, n\}$ for $n \in \mathbb{N}$. The input distribution satisfies the promise that either (1) $S_i \cap S_j = \emptyset$ for every $1 \le i < j \le k$, or (2) there is a unique item $a \in [n]$ such that $a \in S_i$ for all $i \in [k]$, and for any other $a' \neq a$, there is at most one $i \in [k]$ for which $a' \in S_i$. It is well-known that the randomized communication complexity of this problem is $\Omega(n/k)$ [3, 8, 10], and that the bound holds even for multiple rounds of communication and players share a common blackboard. However, if we look at $t < k$ players and an arbitrary, even if the worst-case mapping of the input sets $S_1, \ldots, S_k$ to the t players, then by the pigeonhole principle there exists a player who gets two input sets S_i, S_j with $i \neq j$. Now this player can locally determine the output of the function by checking if $S_i \cap S_j = \emptyset$. Thus with $t < k$ players the problem is solvable using $O(1)$ bits per player. This simple argument shows that for non-product distributions, there can be an arbitrarily large gap between the k-player and the t-player worst-case-partitioned randomized communication complexities. Note that this example applies to a symmetric problem, meaning that the k-player set disjointness problem is invariant under any one-to-one assignment of $x_1, \ldots, x_k$ to the k players.

Perhaps surprisingly, and this is one of the main messages of our work: for symmetric functions and product input distributions, we show that for any $t < k$, for deterministic one-way communication complexity or randomized one-way communication complexity with error probability $1/\mathsf{poly}(k)$, there *is no gap* in maximum message length between the k-player and t-player communication complexities. That is, the gap is at most a multiplicative $O(1)$ factor in message length and $O(k)$ in total communication. Further, this gap is tight, as there are problems for which the input distribution is a product distribution, and the t-player communication with $1/\mathsf{poly}(k)$ error probability is $O(\log k)$ for constant $t = O(1)$, while the k-player communication with $1/\mathsf{poly}(k)$ error probability is $\Omega(k \log k)$. Thus, the answer for product input distributions is significantly different than what we saw for non-product distributions, even for symmetric functions.

We also show that for constant error protocols and under product input distributions, the gap is at most a multiplicative $O(\log k)$ factor in message length and $O(k \log k)$ in total communication. Further, we show there exists a symmetric function and input distribution which is product on any $k-1$ out of k inputs, for which this gap is best possible. We leave open the question of the existence of a symmetric function and product input distribution (on all k inputs rather than $k-1$ out of k) which realizes this gap for constant error protocols.

One takeaway message from our results is that when showing space lower bounds for data stream algorithms computing symmetric functions on product distributions, by looking at 2-player communication complexity (which is by far the most common communication setup), there is only an $O(1)$ factor loss for error probability $1/\mathsf{poly}(k)$ protocols, and an $O(\log k)$ factor loss for constant error protocols.

Data Stream Lower Bounds: As a key application of our lower bound techniques, we provide a space lower bound for $(1 \pm \varepsilon)$-approximating the *Hamming norm* in the strict turnstile model. This problem, which is also known as the L_0 *norm estimation* and denoted by T_ε, requires estimating $\|\mathbf{x}\|_0 := |\{i \mid x_i \neq 0\}|$ of a vector $\mathbf{x} = (x_1, \ldots, x_N)$ and outputting an estimate $\widetilde{F}$ for which $(1-\varepsilon)\|\mathbf{x}\|_0 \le \widetilde{F} \le (1+\varepsilon)\|\mathbf{x}\|_0$ with constant probability. The vector $\mathbf{x}$ is initialized to all zeros and undergoes a sequence of m updates each of the form $(i, v) \in [N] \times [\pm M]$, where $[\pm M] := \{0, \pm 1, \ldots, \pm M\}$ and each update (i, v) causes $x_i \leftarrow x_i + v$. In the strict turnstile model $x_i \geq 0$ holds for all i and at all points in the stream. We obtain an $\Omega\left(\varepsilon^{-2}\log(N)\log\log(mM)\right)$ bits of space lower bound for $(1 \pm \varepsilon)$-approximating the Hamming norm. This lower bound matches the best known upper bound $O\left(\varepsilon^{-2}\log(N)\left(\log(1/\varepsilon) + \log\log(mM)\right)\right)$ [12] for any $\varepsilon \geq 1/\mathsf{polylog}(mM)$. Note that $\varepsilon \geq 1/\mathsf{polylog}(mM)$ is required in order to obtain polylogarithmic space, and so is the most common setting of parameters. Perhaps surprisingly, there is an upper bound of $O\left(\varepsilon^{-2}\log(mM)\right)$ bits of space for $(1 \pm \varepsilon)$-approximating L_p for $p > 0$ [13] (improving an earlier $O\left(\log^2 N\right)$ bound of [9]; see also a time-efficient version in [11]), and thus we provide a strict separation in the complexities for $p = 0$ and $p > 0$. The Hamming norm has many applications, as it corresponds to estimating the number of distinct values, and can be used to estimate set union and intersection sizes (see [7] where it was introduced).

Technical Overview: We first illustrate the idea behind showing there is no gap between k-player and 2-player deterministic one-round communication complexity. The first player P_1 of the k-player protocol pretends to be Alice, the first player of the 2-player protocol, to create the message m_1 as Alice would do and sends it to the second player P_2 of the k-player protocol. Having received this message m_1, P_2 enumerates over all possible inputs of P_1 until finding one which would cause P_1 to send m_1. Since the protocol is deterministic and it evaluates a function defined on a product domain, meaning that it is a total function on a domain of the form $S_1 \times S_2 \times \cdots \times S_k$, the function value must be the same as long as P_1's input results in the same message m_1 to be sent. So P_2 can arbitrarily pick one of those inputs as his guess for P_1. Now P_2 has a guess x for P_1's input together with his own input y, and P_2 can simulate Alice in the 2-player protocol. This is feasible because the 2-player protocol works under any partitioning of the inputs. Then P_2 sends to the third player P_3 the message that Alice would send to Bob in the 2-player protocol, given that Alice had input (x, y). In case when every player P_i cannot figure out how many input items have been processed from his own input and the received message m_{i-1}, which is important for his simulation of the 2-player protocol, an additional logarithmic-many-bits index carrying this piece of information should be passed together with the simulated messages. In this way, the entire k-player protocol can be simulated and the per player communication equals to the communication of the 2-player protocol between Alice and Bob, sometimes plus the additional logarithmic many bits for the index. Moreover, both protocols are deterministic.

For the randomized case with a product input distribution, we first consider 2-player protocols with error probability $1/\mathsf{poly}(k)$. We would like to run the same simulation as for deterministic protocols, except now it is unclear how the second player P_2 can reconstruct a valid input x for the first player P_1 from the first message m_1. A natural thing would be for P_2 to choose the input x_1 to P_1 for which the probability of sending m_1, given that P_1's input is x_1, is greatest. This is not correct though, since the overall probability of P_1 holding x_1 and sending m_1 may be less than the $1/\mathsf{poly}(k)$ error bound and the protocol could afford to be always wrong on such a combination of x_1 and m_1. Thus we need some balancing between two probabilities: i) the first player P_1 sends m_1 on input x_1; and ii) the protocol output is correct given that P_1 has input x_1 and sends m_1.

The above naturally suggests that we should impose an input product distribution μ. Then it must be that for a good fraction of x, weighted according to μ, the k-player protocol is correct when the first player has input x_1 and sends message m_1. Thus we can sample x from the conditional distribution on μ given that message m_1 is sent. Here, for correctness, it is crucial that μ is a product distribution; this ensures for most settings of remaining player's inputs (weighted according to μ), for most choices of x_1 (weighted according to μ)

giving rise to m_1, the function evaluated on the inputs is the same, and x_1 can be sampled independently of remaining inputs. Once we have sampled x_1, and given that the second player has private input x_2 in the k-player protocol, we can then have the second player pretend to be Alice of a randomized 2-player protocol with input (x_1, x_2), similar to the deterministic case. Ultimately, we will show that under distribution μ we obtain a protocol with total communication at most $O(k)$ times that of the 2-player protocol with error probability $1/\mathsf{poly}(k)$ (and an $O(1)$ multiplicative blowup in maximum message length, times that of the 2-player protocol), where the factor k comes from the number of invocations of the 2-player protocol.

We illustrate the optimality of the randomized reduction above by looking at the Sum-Equal problem studied by Viola [16]: in this problem each of k players holds an input x_i mod p, where $p = \Theta\left(k^{1/4}\right)$ is a prime, and they wish to determine whether $\sum_i x_i = 0$ or $1 \mod p$. Viola shows this problem has randomized communication complexity $\Theta(k \log k)$, for both randomized protocols with constant error probability as well as deterministic protocols (and thus also randomized protocols with $1/\text{poly}(k)$ error probability). Moreover, for randomized protocols with $1/\text{poly}(k)$ error probability, Viola's $\Omega(k \log k)$ lower bound holds even for a product distribution on the inputs (where if $\sum_i x_i \mod p \notin \{0,1\}$ the output can be arbitrary). We observe that under any partition of the inputs into 2-players Alice and Bob, the problem can be solved with $O(\log k)$ bits with probability $1 - 1/\text{poly}(k)$ just by running an equality test on the sum modulo p of Alice and the negated sum modulo p of Bob. Thus, this illustrates that the factor $O(k)$ gap for protocols for product input distributions with $1/\text{poly}(k)$ error probability is *optimal*.

On the other hand, for constant error protocols and a product input distribution, there is a 2-player $O(1)$ bit upper bound in the public coin model which comes from running an equality test with constant error probability (since we measure error with respect to an input distribution, equality has an $O(1)$ upper bound with constant error). We note that the k-player protocol has communication $\Omega(k \log k)$ for constant error protocols, which gives the $\Omega(k \log k)$ factor gap we claimed. The only downside is that the $\Omega(k \log k)$ lower bound holds for an input distribution which is product on $k-1$ out of k players, rather than all k players. We leave it as an open question to give an optimal separation for product input distributions for constant error probability.

Given the importance of Viola's problem in showing separations, we next show a *direct sum theorem* for his problem, showing its communication complexity increases to $\Omega(kr \log k)$ for solving a constant fraction of r independent copies. To show the direct sum theorem for Viola's problem, one issue is that, unlike for two players where the technique of *information complexity* often provides direct sum theorems, for k-players the analogues are much weaker. A natural route would be to take Viola's *corruption bound*, argue it implies a high information bound, and then apply standard direct sum theorems for information. This approach does not give an information cost lower bound on private coin protocols, though one can fix it for two players using [5], which improves upon a bound in [6]. However, for k players similarly strong bounds are unknown. Another natural approach is to use the fact that if a problem has a corruption bound, then one immediately has a direct sum for it [4]. Again though, this is only for two players or the *number on forehead* model, and not for our setting.

Instead, our proof is inspired by Viola's rectangle argument for a single copy of the Sum-Equal problem, where each rectangle, restricted to the first $k-1$ players, is a product distribution on which the protocol generates a message to the k-th player. We use a rectangle argument on multiple copies where the output is now a binary vector instead of a single bit. The main obstacle is that we must consider the Hamming distance between the protocol output and the correct answer in a vector space, which is much more involved than studying the error probability for a single instance. The intuition of our proof is that for every large rectangle, there must be linearly many copies that appear (almost) uniformly random in the last player's view. The above argument is fairly intricate, and involves several levels of conversion: i) a large rectangle implies large conditional entropy in many players' inputs; ii) the large entropy of all copies implies we have min-entropy at least 1 on many copies; iii) a random variable of min-entropy at least 1 can always be decomposed into a convex combination of uniform distributions over two elements; iv) the summation of sufficiently many independent random variables that are each drawn from a uniform-over-two-element distribution turns out to be nearly uniform, and hence many Sum-Equal copies look uniform to the last player.

Thus, the last player can hardly outperform a random guess. Note that it is insufficient to prove uniformity for many copies individually (which is not too hard using the same idea as in Viola's proof), since such a situation could be simulated with a much smaller rectangle with very small error. We instead perform our rectangle argument inductively to show most copies appear almost uniform, even if conditioned on previous copies. For space considerations this induction is mostly deferred to the full version.

This direct sum technique has further applications. One application is to proving a lower bound for approximating the Hamming norm in a strict turnstile stream. Using a result of [2], to show lower bounds for streaming algorithms in the strict turnstile model, it suffices to show lower bounds in the simultaneous communication model, where each player simultaneously sends a message to a referee who outputs the answer. While our direct sum theorem holds in this more restrictive model, we also need to consider a composition of the gap-Hamming problem on top of the Sum-Equal instances as well as an augmented index version of the composed problem. In the augmented problem we additionally give a referee an index i and the answers to all copies j, with $j > i$. Similar augmentation has been studied for L_p-norms [13]. This allows us to reduce our communication problem to Hamming norm approximation, and ultimately prove our data stream lower bound.

2 Preliminaries

A function $f : \Sigma^k \to \Gamma$ is called a *k-party symmetric function* if for every $(x_1, x_2, \ldots, x_k) \in \Sigma^k$ and for every permutation σ over $\{1, 2, \ldots, k\}$, there is $f(x_1, \ldots, x_k) = f\left(x_{\sigma(1)}, \ldots, x_{\sigma(k)}\right)$. A k-dimensional vector space S is called a *product space* if it can be represented as $S = S_1 \times S_2 \times \cdots \times S_k$. A distribution μ is called a *product distribution* if it is obtained by taking the product of k independent distributions, i.e., $\mu = \mu_1 \times \mu_2 \times \cdots \times \mu_k$.

In the t-player communication complexity model, there are t computationally unbounded players, e.g., $P_1, \ldots, P_t$, required to compute a function $f : X_1 \times \cdots \times X_t \to Y$, where f is usually a t-party symmetric function. Each player P_i is given a private input $x_i \in X_i$ and follows a fixed protocol to exchange messages. For every input $(x_1, \ldots, x_t)$, the message transcript is denoted by $\Pi_t(x_1, \ldots, x_t)$ when all players follow the protocol Π_t (when Π_t is randomized, $\Pi_t(x_1, \ldots, x_t)$ is a random variable taking probabilities over players' random coins). A deterministic protocol Π_t computes f if there is a function Π_{out} such that $\Pi_{out}\left(\Pi_t^{(t)}(x_1, \ldots, x_t), x_t\right) \equiv f$, where $\Pi_t^{(t)}(x_1, \ldots, x_t)$ denotes P_t's view under the execution of Π_t on input $(x_1, \ldots, x_t)$ and for simplicity we let $\Pi_{out}(x_1, \ldots, x_t) := \Pi_{out}\left(\Pi_t^{(t)}(x_1, \ldots, x_t), x_t\right)$. A δ-error randomized protocol Π_t for f requires the existence of Π_{out} such that for all inputs $(x_1, \ldots, x_t)$, $\Pr\left[\Pi_{out}(x_1, \ldots, x_t) = f(x_1, \ldots, x_t)\right] \geq 1 - \delta$. The *communication cost* of Π_t is the maximum size of $\Pi_t(x_1, \ldots, x_t)$ over all $x_1, \ldots, x_t$ and all random coins. The *t-player deterministic communication complexity* (resp. *t-player δ-error randomized communication complexity*), denoted by $\mathbf{DCC}_t(f)$ (resp. $\mathbf{RCC}_{t,\delta}(f)$), is the cost of the best t-player deterministic (resp. δ-error randomized) protocol Π_t for f.

Given a k-party function $f : X_1 \times \cdots \times X_k \to Y$ and $t < k$, we define $\mathbf{DCC}_t(f)$ and $\mathbf{RCC}_{t,\delta}(f)$ under a *worst-case partition* of inputs. That is, let $f_t(z_1, \ldots, z_t) = f(x_1, \ldots, x_k)$ be defined for every partition $i_0 = 0 \le i_1 \le \cdots \le i_t = k$ and $z_j := (x_{i_{j-1}+1}, \ldots, x_{i_j})$, and the t-player communication complexity of f is defined with respect to the worst choice of f_t, i.e., $\mathbf{DCC}_t(f) := \max_{f_t} \mathbf{DCC}_t(f_t)$ and $\mathbf{RCC}_{t,\delta}(f) := \max_{f_t} \mathbf{RCC}_{t,\delta}(f_t)$.

Given a t-party function f and its input distribution μ, we let $\mathbf{DCC}^{\mu}_{t,\delta}(f)$ denote the communication cost of the best t-player deterministic protocol Π_t computing f such that $\Pr_{x\sim\mu}[\Pi_{out}(x) \neq f(x)] \le \delta$. Similarly we define $\mathbf{RCC}^{\mu}_{t,\delta}(f)$ for randomized protocols.

In the restricted *one-way communication model* [15, 1, 14], the i-th player sends exactly one message to the $(i+1)$-st player for $i \in [t-1]$ following Π_t, and then P_t announces the output of Π_t as specified by Π_{out}. Note that in this setting there are only $k-1$ messages sent by $P_1, \ldots, P_{k-1}$, and we do not count the final output announced by P_t in the communication in order to best correspond to streaming algorithms. This is also known as a *sententious* protocol in previous work, e.g., [16]. We denote the *t-player one-way communication complexities of f* by $\overrightarrow{\mathbf{DCC}}_t(f)$ and $\overrightarrow{\mathbf{RCC}}_{t,\delta}(f)$, respectively.

In the *common reference string model* (aka *CRS model*), there is a sequence of public random coins, which is by default a uniformly random binary string, accessible to all players. The obvious advantage of communication in the CRS model is that players have access to the same random string and thus save the cost of synchronizing their private coins.

A streaming algorithm is an algorithm that scans the input $(x_1, \ldots, x_m) \in \Sigma^m$ as m stream input items in sequence, updates its internal memory of size $s = o(m \log|\Sigma|)$ (i.e., a streaming automaton with 2^s states, where the space cost of updating the internal memory is not accounted for), and finally outputs a function $f(x_1, \ldots, x_m)$ evaluated on all input items. If the best deterministic (resp. δ-error randomized) streaming algorithm computes f with s bits of memory and t passes over the data stream, then we say the *deterministic* (resp. *δ-error*) *streaming complexity* of f is st, denoted by $\mathbf{DSC}(f) = st$ (resp. $\mathbf{RSC}_\delta(f) = st$). In a popular and standard setting, a streaming algorithm scans the input stream in a *single pass* and only processes every input item once. The necessary amount of memory required by such single-pass algorithms is called the *single-pass deterministic/δ-error streaming complexity* and denoted by $\overrightarrow{\mathbf{DSC}}(f)$ and $\overrightarrow{\mathbf{RSC}}_\delta(f)$ respectively.

Note that every streaming algorithm can be naturally interpreted as a communication protocol where each party holds some (possibly an empty set of) input items on the stream and the messages capture the memory updates. The connection between streaming complexity and communication complexity trivially follows in the following lemma.

▶ **Lemma 1.** *For every function f and error tolerance δ, for every $k \in \mathbb{N}$, it holds that:*

$$\mathbf{DSC}(f) \ge \frac{1}{k} \cdot \mathbf{DCC}_k(f), \quad \mathbf{RSC}_\delta(f) \ge \frac{1}{k} \cdot \mathbf{RCC}_{k,\delta}(f)$$

Furthermore, similar relations hold for $\overrightarrow{\mathbf{DSC}}$, $\overrightarrow{\mathbf{RSC}}_\delta$ and $\overrightarrow{\mathbf{DCC}}_k$, $\overrightarrow{\mathbf{RCC}}_{k,\delta}$.

3 Communication Complexity for Functions on Non-Product Spaces

▶ **Theorem 2.** *For every $t \ge 2$, there is a t-party symmetric function f defined on $D \subseteq \{0,1\}^n = \left(\{0,1\}^{n/t}\right)^t$ such that for $\delta < 1/4$, $\overrightarrow{\mathbf{DCC}}_{t-1}(f) \le t-1$ but $\mathbf{RCC}_{t,\delta}(f) = \Omega(n/t)$. If $t = O(1)$, then $\overrightarrow{\mathbf{DCC}}_{t-1}(f) = O(1)$ and $\mathbf{RSC}_\delta(f) \ge \frac{1}{t} \cdot \mathbf{RCC}_{t,\delta}(f) = \Omega(n)$.*

Proof. Consider the t-party set disjointness problem $\textsc{Disj}_{n/t,t}$ defined as follows: there are t players $P_1, \dots, P_t$ such that every player P_i holds a private indicator vector $\mathbf{x}_i \in \{0,1\}^{n/t}$ which represents a subset of $[n/t]$, i.e., $\textsc{Disj}_{n/t,t}(\mathbf{x}_1, \dots, \mathbf{x}_t) = \vee_{j=1}^{n/t} (\wedge_{i=1}^{t} x_{i,j})$, where $x_{i,j}$ denotes the j-th coordinate of $\mathbf{x}_i$. We consider the domain D such that the vectors $\mathbf{x}_1, \dots, \mathbf{x}_t \in \{0,1\}^{n/t}$ are either (1) pairwise disjoint, or (2) sharing a unique element $j \in [n/t]$. Let f be the function that computes $\textsc{Disj}_{n/t,t}$ on domain D.

On the one hand, it is easy to verify that $\overrightarrow{\mathbf{DCC}}_{t-1}(f) \le t-1$. Indeed, at least one of the $t-1$ players obtains two distinct indicator vectors and hence can itself decide the output of f. The communication is 1 bit per player to pass the result, and hence the total communication is bounded by $t-1$ since there are $t-1$ players.

On the other hand, the $\Omega(n/t)$ lower bound for $\mathbf{RCC}_{t,\delta}(f)$ follows from the known lower bound for multi-player set disjointness (see [3], which was improved to optimal in [8, 10]). The lower bound for $\mathbf{RSC}_{\delta}(f)$ immediately follows by Lemma 1. ◀

4 Deterministic Communication and Streaming Complexity

We first show that 2-player one-way communication complexity is equivalent to the streaming complexity of single-pass streaming algorithms in the deterministic setting.

▶ **Theorem 3.** *For every symmetric function f, $\overrightarrow{\mathbf{DCC}}_2(f) \le \overrightarrow{\mathbf{DSC}}(f) \le \overrightarrow{\mathbf{DCC}}_2(f) + \log n$.*

Proof. Obviously, $\overrightarrow{\mathbf{DSC}}(f) \ge \overrightarrow{\mathbf{DCC}}_2(f)$ since a 2-player communication protocol simulates a streaming algorithm. It remains to prove $\overrightarrow{\mathbf{DSC}}(f) \le \overrightarrow{\mathbf{DCC}}_2(f) + \log n$.

Suppose the input stream is $(x_1, \dots, x_n) \in \Sigma^n$, and for every partition into $(x_1, \dots, x_i)$ and $(x_{i+1}, \dots, x_n)$ there is a deterministic 2-player one-way protocol Π_2^i computing f. We design the deterministic single-pass streaming algorithm A for f by simulating 2-player one-way communication protocols under different partitions. The memory usage of A is therefore bounded by the maximum communication cost of the simulated 2-player protocols plus an index in $[n]$ recording the number of processed items. Notice that when processing the item x_{i+1}, A has already processed $x_1, \dots, x_i$ and has (m_i, i) in memory. A can thus reconstruct a compatible guess of $x_1'', \dots, x_i''$ that would induce exactly the message m_i as in Π_2^i, and then sets the memory to be $(m_{i+1}, i+1)$ where m_{i+1} is the message sent in Π_2^{i+1} when P_1 has $(x_1'', \dots, x_i'', x_{i+1})$ and P_2 has $(x_{i+2}, \dots, x_n)$. A repeats this process for every $i = 1, \dots, n-1$ and at the end it outputs $f(x_1, \dots, x_n)$.

Therefore, we complete the proof with $\overrightarrow{\mathbf{DCC}}_2(f) \le \overrightarrow{\mathbf{DSC}}(f) \le \overrightarrow{\mathbf{DCC}}_2(f) + \log n$. ◀

▶ **Corollary 4.** *For every k-party symmetric function f,*

$$(k-1) \cdot \overrightarrow{\mathbf{DCC}}_2(f) \le \overrightarrow{\mathbf{DCC}}_k(f) \le (k-1) \cdot \left(\overrightarrow{\mathbf{DCC}}_2(f) + \log k\right)$$

Proof. Combining Lemma 1 and Theorem 3, it follows that

$$\overrightarrow{\mathbf{DCC}}_k(f) \le (k-1) \cdot \overrightarrow{\mathbf{DSC}}(f) \le (k-1) \cdot \left(\overrightarrow{\mathbf{DCC}}_2(f) + \log k\right)$$

The other direction $\overrightarrow{\mathbf{DCC}}_k(f) \ge (k-1) \cdot \overrightarrow{\mathbf{DCC}}_2(f)$ holds by giving $z_j = \emptyset$ to every player $j \in \{2, \dots, k-1\}$ in the k-player case, when the problem degenerates to 2-player communication but the same message has to be passed $k-1$ times. ◀

Such a linear separation naturally extends to the communication complexity of t-player versus k-player protocols, as long as $2 \le t < k$. Thus, the deterministic communication complexity grows *linearly* in the number of parties.

We remark that if every player must get a non-trivial input, i.e., at least one input element to the function, the linear growth remains for some but not all problems. For example, the communication complexity of the parity of k bits is linear in the number of players. However, to decide whether k elements in $[k]$ are distinct, the 2-player protocol requires communication $\log \binom{k}{k/2} \approx k - \log \sqrt{k}$, whereas the k-player worst-case communication grows sublinearly, i.e. for k players the communication is no more than $\sum_{i=1}^{k-1} \log \binom{k}{i} \ll (k-1) \cdot \log \binom{k}{k/2}$.

5 Communication Complexity for Functions on a Product Space

5.1 Separations for Randomized Communication Complexity

In this section, we consider the communication cost of randomized multi-player protocols defined on product input distributions and present a $k \log k$ versus $t \log t$ separation between k-player and t-player communication complexity.

First we introduce the Sum-Equal problem (as used in Viola's work [16]).

The *k-player* Sum-Equal *over integers*, denoted by Sum-Equal$_k$, requires deciding whether $\sum_{i=1}^{k} x_i = 0$, where each player P_i is given an integer x_i as well as k. In the CRS model, an additional public random string is also known to all players. The *k-player* Sum-Equal *over* $\mathbb{Z}_m$, denoted by Sum-Equal$_{k,m}$, is defined similarly as Sum-Equal$_k$, except that the input items are drawn from $\mathbb{Z}_m$ and the summation is over $\mathbb{Z}_m$, for a publicly known m.

▶ **Lemma 5** ([16], Theorem 15 and Theorem 29). *For every $k \in \mathbb{N}$, $0 \leq \delta \leq 1/3$, and in the CRS model, the k-player δ-error communication complexity of* Sum-Equal *satisfies:*

(a) *For every $m \in \mathbb{N}$, $\overrightarrow{\mathbf{RCC}}_{k,\delta}(\text{Sum-Equal}_{k,m}) = O\left(k \log(k/\delta)\right)$.*

(b) *For every prime $p \in (k^{1/4}, 2k^{1/4})$, $\mathbf{RCC}_{k,\delta}(\text{Sum-Equal}_{k,p}) = \Omega\left(k \log k\right)$.*

In particular, $\mathbf{RCC}_{k,\delta}(\text{Sum-Equal}_{k,p}) = \Theta\left(k \log k\right)$ in the CRS model if $\delta = \Omega\left(1/\mathsf{poly}(k)\right)$.

We remark that Viola's lower bound for Sum-Equal$_{k,p}$ is proved for a non-product distribution μ_H whose support covers exactly a $2/p$ fraction of the whole (product) input space. Thus if a k-player protocol solves Sum-Equal$_{k,p}$ with error $\delta \leq 1/k$ on a uniform distribution μ over the whole input space, then its error with respect to μ_H is bounded by $\frac{1/k}{2/p} < k^{-3/4}$. By Lemma 5, the $\Omega\left(k\right)$ separation in Corollary 6 naturally follows.

▶ **Corollary 6.** *For prime $p \in (k^{1/4}, 2k^{1/4})$ and $\delta \leq 1/\mathsf{poly}(k)$, there is a product distribution μ such that $\mathbf{RCC}^{\mu}_{k,\delta}(\text{Sum-Equal}_{k,p}) = \Omega\left(k \log k\right)$, $\overrightarrow{\mathbf{RCC}}_{2,\delta}(\text{Sum-Equal}_{k,p}) = O\left(\log k\right)$.*

For a larger error tolerance, say δ is a constant, we have a stronger separation between k-party communication and t-party communication. However, the hard distribution is slightly non-product, that is, it is a product distribution on any $k-1$ out of the k players.

▶ **Corollary 7.** *For every $k \in \mathbb{N}$, there is a k-party symmetric function f such that*

(a) *For any product distribution μ, for every $2 \leq t \leq k$ and $0 \leq \delta \leq 1/3$, $\overrightarrow{\mathbf{RCC}}^{\mu}_{t,\delta}(f) = O\left(t \log(t/\delta)\right)$. In particular, $\overrightarrow{\mathbf{RCC}}^{\mu}_{2,\delta}(f) = O\left(\log(1/\delta)\right)$.*

(b) *There exists a distribution μ_H, which is product on any $k-1$ out of k players, for which $\mathbf{RCC}^{\mu}_{k,\delta}(f) = \Omega\left(k \log k\right)$ as long as $\delta \leq 1/3$.*

For $\delta \geq 1/\mathsf{poly}(t)$, the gap between $\mathbf{RCC}^{\mu}_{k,\delta}(f)$ and $\overrightarrow{\mathbf{RCC}}^{\mu}_{t,\delta}(f)$ is bounded as below:

$$\mathbf{RCC}^{\mu}_{k,\delta}(f) \;/\; \overrightarrow{\mathbf{RCC}}^{\mu}_{t,\delta}(f) = \Omega\left(\frac{k \log k}{t \log t}\right)$$

The outline of the proof of Corollary 7 was given in Section 1. That is, the upper bound in part (a) follows from applying $k = t$ in the first part of Lemma 5, while the lower bound in part (b) follows from the second part of Lemma 5. We defer the proofs to the full version.

5.2 Tightness of the Communication Complexity Separation

The following theorem and corollary show tightness of our separations.

▶ **Theorem 8.** *For every k-party function $f : \Sigma^k \to \Gamma$, product distribution μ over Σ^k, and error tolerance $\delta < 1/3$, if the optimal δ-error 2-player one-way protocol for f does not degenerate to the deterministic case, then the following holds:*

$$\overrightarrow{\mathbf{RCC}}^{\mu}_{k,\delta}(f) \;/\; \overrightarrow{\mathbf{RCC}}_{2,\delta}(f) \leq O\left(k \cdot \left(1 + \frac{\log k}{\log(1/\delta)}\right)\right) = \begin{cases} O\left(k \log k\right) & \text{if } \delta = \Omega\left(1\right) \\ O\left(k\right) & \text{if } \delta = 1/k^{\Omega(1)} \end{cases}$$

Proof sketch. We present the major steps and leave the complete proof to the full version.

First we let Π_0 be the optimal δ-error 2-player one-way protocol Π_0 that computes f with communication $C = \overrightarrow{\mathbf{RCC}}_{2,\delta}(f)$, and construct a new protocol Π_2 by taking $M = O\left(1 + \frac{\log k}{\log(1/\delta)}\right)$ repetitions of Π_0 such that the error probability of Π_2 is reduced to $\delta^2/(16k^2)$. Note that Π_2 is still a 2-player one-way protocol but has communication $O\left(CM\right)$.

Second we prove that for every product input distribution μ over Σ^k, the k-party function f can be evaluated by a randomized k-player one-way protocol Π_k with communication $O\left(k \cdot CM\right)$ and error $\delta/2$ with respect to μ. The idea is that given μ, each player P_i: 1) assumes that the received message m_{i-1} from P_{i-1} will lead to a correct answer with probability $\geq 1 - \frac{\delta}{4k}$; 2) samples a possible input $x'_1, \ldots, x'_{i-1}$ of previous players $P_1, \ldots, P_{i-1}$ on which with probability $\geq 1 - \frac{\delta}{4k}$ the protocol is correct conditioned on m_{i-1} being sent and $(x'_1 \ldots, x'_{i-1}, x_i, \ldots, x_k)$ being the actual input (here we use that μ is a product distribution); 3) and finally sends a message m_i of length $O\left(CM\right)$ as in Π_2 where Alice has input $(x'_1, \ldots, x'_{i-1}, x_i)$. By a union bound the error probability of Π_k is bounded by $\delta/2$ with respect to μ. The fact that μ is a product distribution is used in the second step where the sampling process relies on that previous players' inputs are independently distributed from that of future players.

Thus we finish the proof and conclude that $\overrightarrow{\mathbf{RCC}}^{\mu}_{k,\delta}(f) \leq O\left(kCM\right)$. ◀

Notice that in the proof of Theorem 8, every message in Π_k has the length bounded by $O\left(CM\right)$, which gives an upper bound for the single-pass streaming complexity.

▶ **Corollary 9.** *For every k-party function f and product input distribution μ, and for every $\delta < 1/3$, $\mathbf{RSC}^{\mu}_{\delta}(f) \leq \overrightarrow{\mathbf{RSC}}^{\mu}_{\delta}(f) \leq O\left(1 + \frac{\log k}{\log(1/\delta)}\right) \cdot \overrightarrow{\mathbf{RCC}}_{2,\delta}(f)$.*

6 A Direct Sum for Viola's Problem

We next turn to our direct sum theorem for Viola's problem, which is a crucial building block for our streaming application.

▶ **Theorem 10.** *Let $F : \left(\mathbb{Z}_p^m\right)^k \to \{0,1\}^m$ be the k-party function computing m independent copies of* Sum-Equal$_{k,p}$*, where p is a prime between $k^{1/4}$ and $2k^{1/4}$. For every error tolerance $\delta \in (0, 1/9)$, we say a protocol Π is* correct with probability $1 - \delta$ *if there is a reconstruction function G such that for every fixed $i \in [m]$ and input $x \in \left(\mathbb{Z}_p^m\right)^k$, $G\big(i, \Pi_{out}(x)\big)$ equals the output of the i-th instance of* Sum-Equal$_{k,p}$ *with probability at least $1 - \delta$, over the internal randomness of Π. Then the communication cost of any Π which is correct with probability $1 - \delta$, is $\Omega\left(mk \log k\right)$.*

We give a sketch of the proof of Theorem 10 here, and defer the full proof to the full version.

Proof sketch of Theorem 10. First we fix the randomness used in the protocol Π and convert it into a deterministic protocol Π' that has δ error with respect to a specific input distribution $\mathcal{H}$. Here $\mathcal{H} = (X_1, \ldots, X_{k-1}, X_k + v)$ for independent $X_1, \ldots, X_{k-1}$ uniformly distributing over $\mathbb{Z}_p^m$, $X_k = -\sum_{j=1}^{k-1} X_j$ and v uniformly sampled from $\{0,1\}^m$. Note that $\mathcal{H}_{-k} := (X_1, \ldots, X_{k-1})$ is uniform over $\left(\mathbb{Z}_p^m\right)^{k-1}$.

We next recall the intuition behind rectangle arguments in multi-player number-in-hand communication complexity: every k-player (number-in-hand) deterministic protocol with communication at most c partitions the inputs into $C = 2^c$ sets $R^1, R^2, \ldots, R^C$, where each R^i is a *rectangle* in the form of $R^i = R_1^i \times R_2^t \times \ldots \times R_k^i$ such that every input in R^i induces exactly the same transcript π_i. We will use the rectangle argument to show that Π' uses communication $c \geq \Omega(1) \cdot mk \log k$.

The main step is the following claim (with proof sketched later in this subsection):

▷ Claim 11. If $c < \frac{1-9\delta}{135} \cdot mk \log k$, then for every rectangle R satisfying $\Pr[\mathcal{H}_{-k} \in R_{-k}] \geq 1/(3C) = 1/(3 \times 2^c)$, there must be $L \subseteq [m]$ and $\ell := |L| \geq 9\delta m$ such that conditioned on $X_{-k} \in R_{-k}$, the distribution of $X_k^{(L)}$, which is X_k restricted on L, is ℓ/p-close to the uniform distribution over $\mathbb{Z}_p^\ell$.

Using Claim 11, it is easy to show $\Pr\left[\Pi'(\mathcal{H}) \text{ errs on } \leq 3\delta m \text{ coordinates}\right] \leq 2/3$, which contradicts that Π' has δ error with respect to $\mathcal{H}$ and $\delta < 1/9$. Therefore, the communication cost of Π', and hence of Π, must be $\geq \frac{1-9\delta}{135} \cdot mk \log k = \Omega(mk \log k)$. ◀

Proof sketch of Claim 11. This claim is proved using induction on the size of L. Suppose the claim is true for (w.l.o.g.) the first $\leq \ell - 1$ indices, we prove it for the next one. More specifically, we show that the last player P_k gets nearly no information about the ℓ-th copy when the input distribution follows $\mathcal{H}$ and X_{-k} falls into a sufficiently large rectangle $R_{-k} = R_1 \times \cdots \times R_{k-1}$. That is, for $X_{-k} \sim \left(\mathbb{Z}_p^m\right)^{k-1}$ and $X_k = -\sum_{j=1}^{k-1} X_j$, the marginal distribution $X_k^{(\ell)} \mid X_{-k} \in R_{-k}$ is statistically close to uniform.

The proof outline is as follows: first, let $\mathcal{E}_x$ denote the event that the first $k-1$ players have x on their first $\ell - 1$ coordinates, i.e. $X_{-k}^{[\ell-1]} = x$. Second, we consider frequently appearing x conditioned on $\mathcal{H}_{-k} \in R_{-k}$ such that $\Pr\left[\mathcal{E}_x \mid \mathcal{H}_{-k} \in R_{-k}\right] \geq \frac{1}{2p^{1+(\ell-1)(k-1)}}$ (the missed probability measure is at most $\frac{1}{2p}$ since there are $\leq p^{(\ell-1)(k-1)}$ different choices of x), and let $J_x \subseteq [k-1]$ be the set of players whose input falls into R_{-k} with "significant" probability conditioned on $\mathcal{E}_x$. Specifically, we prove that J_x must have size $|J_x| \geq 0.5k - 1$ for $J_x := \left\{ j \in [k-1] \,\middle|\, \Pr\left[X_j \in R_j \mid \mathcal{E}_x\right] \geq 2^{-1-2c/k} \right\}$. Third, for every player $j \in J_x$, we consider the set $I_{j,x}$ of coordinates such that for every $i \in I_{j,x}$, the conditional min-entropy of $X_j^{(i)}$ is large given that player j's input X_j is consistent with x and falls into R_j. In particular, for $I_{j,x} := \left\{ i \in [m] \mid \mathsf{H}_\infty\left[X_j^{(i)} \mid X_j \in R_j, \mathcal{E}_x\right] \geq 1 \right\}$, there is $|I_{j,x}| > m - \ell - \frac{2(1-9\delta)}{15} m + 1$.

Finally we apply Chebyshev's inequality and a Chernoff bound together with a standard averaging argument to conclude that there is a fixed coordinate, w.l.o.g. we call it ℓ, such that with probability $\geq 1 - e^{-\Omega(k)}$, the conditional min-entropy $\mathsf{H}_\infty\left[X_j^{(\ell)} \mid X_j \in R_j, \mathcal{E}_x\right] \geq 1$ for $\geq k/30$ players $j \in [k-1]$. As a result, the last player P_k's input $X_k^{(\ell)} = -\sum_{j=1}^{k-1} X_j^{(\ell)}$ is a convex combination of random variables where each of them is the summation of $\geq k/30$ uniform-over-two-elements variables. Repeating a very similar argument as in [16], we conclude that $X_k^{(\ell)}$ is $e^{-\Omega(\sqrt{k})}$ close to uniform.

The overall error probability of above arguments is bounded by $1/p$, which sums up to $\leq \ell/p$ for $X_k^{(L)}$ via a standard union bound. ◁

7 Lower bound for Hamming Norm Estimation

In this section we present a space lower bound for single-pass streaming algorithms for $(1 \pm \varepsilon)$-approximating the Hamming norm L_0, which is denoted by T_ε as in Section 1.1. Recall that the underlying vector is N-dimensional and there are m updates each of magnitude $[\pm M]$.

▶ **Theorem 12.** *For error tolerance $\varepsilon < 1/3$ and $\varepsilon = \max\left\{\Omega\left(\sqrt{\frac{\log k}{k}}\right), \frac{1}{N^{0.49}}\right\}$, any single-pass streaming algorithm solving T_ε with probability $\geq 2/3$ in the strict turnstile model must use $\Omega\left(\varepsilon^{-2}\log(N)\log\log(mM)\right)$ bits of space.*

Proof sketch. We present a proof sketch here, with the detailed proof left to the full paper. First we introduce the $\mathrm{GHSE}_{n,k}$ problem, which is a composition of the n-dimensional gap Hamming weight problem $\textsc{Gap-Hamming}_n$ over the results of n copies of k-player $\textsc{Sum-Equal}_k$ instances, i.e., the result of $\mathrm{GHSE}_{n,k}$ is 1 if there are $\geq (1+\varepsilon)n/2$ underlying $\textsc{Sum-Equal}$ instances outputting 1, and 0 if $\leq (1-\varepsilon)n/2$ instances outputting 1.

The hard problem for our lower bound is the augmented index version of $\mathrm{GHSE}_{n,k}$, which we denote by $\textsc{Aug-Index-GHSE}^t_{n,k}$. In particular, $\textsc{Aug-Index-GHSE}^t_{n,k}$ has $t = \Theta(\log n)$ many $\mathrm{GHSE}_{n,k}$ instances embedded, where the last player P_k is given an index $i \in [t]$ together with the results of $\mathrm{GHSE}^{(i+1)}_{n,k}, \ldots, \mathrm{GHSE}^{(t)}_{n,k}$, and P_k is required to output the result of $\mathrm{GHSE}^{(i)}_{n,k}$. Following the reduction in Theorem 4.1 of [2] it suffices to prove our space lower bound in the simultaneous communication model, where each of $P_1, \ldots, P_{k-1}$ sends a single message to the referee P_k.

In the reduction from $\textsc{Aug-Index-GHSE}^t_{n,k}$ to T_ε, the input integers to underlying $\textsc{Sum-Equal}_k$ instances are processed as updates to distinct elements. Furthermore, every $\textsc{Sum-Equal}_k$ instance of $\mathrm{GHSE}^{(j)}_n$ embedded in the $\textsc{Aug-Index-GHSE}^t_{n,k}$ problem is given frequency 100^{j-1}, i.e., is counted as 100^{j-1} distinct elements. Thus the universe has $N := n + 100 \cdot n + \cdots + 100^{t-1} \cdot n \leq 100^t n/99$ distinct elements in total, and the final Hamming norm is a weighted sum $F := \sum_{j=1}^t 100^{j-1} f_j$, where f_j is the Hamming weight of $\textsc{Sum-Equal}_k$ instances of $\mathrm{GHSE}^{(j)}_n$ for every $j \in [t]$. An algorithm solving T_ε will give a $(1 \pm \varepsilon)$-estimate $\widetilde{F}$ of F, such that $(1-\varepsilon)F \leq \widetilde{F} \leq (1+\varepsilon)F$. From the estimate $\widetilde{F}$ we need to determine the result of $\mathrm{GHSE}^{(i)}_n$ for the given index i. Since the referee can precisely remove the influence of $\mathrm{GHSE}^{(i+1)}_{n,k}, \ldots, \mathrm{GHSE}^{(t)}_{n,k}$ using the auxiliary input before computing $\widetilde{F}$, it suffices to consider the case $i = t$ and the estimation of f_t. Indeed we prove that $\widetilde{F}$ is also a good approximation to $100^{t-1} f_t$ with high probability, as long as the additive error $\sum_{j=1}^{t-1} 100^{j-1} f_j$ is significantly less than the variance of $100^{t-1} f_t$. More specifically,

$$\mathbf{RCC}^{sim}_{k,1/3}\left(\mathrm{T}_\varepsilon\right) \geq \mathbf{RCC}^{sim}_{k,0.4}\left(\textsc{Aug-Index-GHSE}^t_{n,k}\right) \tag{1}$$

for our specified input distribution, which induces variance $O(n)$ on every f_j while our gap in advantage is $\Omega(n)$.

Then we prove that the communication cost of solving the augmented index version of t copies of $\mathrm{GHSE}_{n,k}$ is equal to simultaneously solving $\Omega(t)$ many copies.

$$\mathbf{RCC}^{sim}_{k,0.4}\left(\textsc{Aug-Index-GHSE}^t_{n,k}\right) \geq \Omega\left(t \cdot \mathbf{RCC}^{sim}_{k,0.01}\left(\mathrm{GHSE}_{n,k}\right)\right) \tag{2}$$

The proof relies on the direct sum property of one-way communication for the GHSE problem. The intuition is that all necessary information for computing $\mathrm{GHSE}^{(1)}_{n,k}, \ldots, \mathrm{GHSE}^{(t)}_{n,k}$ must be included in the messages to the referee, since every instance $\mathrm{GHSE}^{(i)}_{n,k}$ can be determined at the referee's position by changing the referee's input alone (without tampering the messages).

Next we prove an $\Omega\left(\varepsilon^{-2}k\log\log k\right)$ lower bound for $\mathbf{RCC}^{sim}_{k,0.1}\left(\mathrm{GHSE}_{n,k}\right)$ and $mM = \mathsf{poly}(k)$. Consider the input $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ to the $\mathrm{GHSE}_{n,k}$ problem, where each player P_j gets $\mathbf{x}_j = \left(\mathbf{x}_j^{(1)}, \ldots, \mathbf{x}_j^{(n)}\right) \in \mathbb{Z}^n$, and for every $i \in [n]$, $Z^{(i)} := \textsc{Sum-Equal}_k\left(\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_k^{(i)}\right)$ denotes the result of the i-th $\textsc{Sum-Equal}$ instance and the range is $\{\pm 1\}$. Let $\mathrm{HSE}(\mathbf{x}) := \sum_{i=1}^{k} Z^{(i)}$ denote the bias of the underlying vector for the $\textsc{Gap-Hamming}_n$ problem embedded in $\mathrm{GHSE}_{n,k}(\mathbf{x})$. Recall that $\mathrm{GHSE}_{n,k}$ distinguishes $\mathrm{HSE}(\mathbf{x}) \geq \varepsilon n$ and $\mathrm{HSE}(\mathbf{x}) \leq -\varepsilon n$, where the gap becomes $\sqrt{n'}$ for $\mathrm{GHSE}_{n',k}$ and $n' = 1/\varepsilon^2$. With random universal hash functions specified by the public randomness, we prove that

$$\mathbf{RCC}^{sim}_{k,0.01}\left(\mathrm{GHSE}_{n',k}\right) \geq \mathbf{RCC}^{sim}_{k,0.1}\left(\textsc{Aug-Index-Sum-Equal}_k^{n''}\right) \tag{3}$$

where $n'' = \Theta\left(n'\right)$ and $\textsc{Aug-Index-Sum-Equal}_k^{n''}$ is the augmented index version of n'' instances of the $\textsc{Sum-Equal}_k$ problem.

Furthermore, the lower bound holds for a distribution μ over $\mathbb{Z}^{n' \times k}$ such that for $\mathbf{x} \sim \mu$ the conditional expectation satisfies $\mathbf{Var}\left(\mathrm{HSE}(\mathbf{x})\right) \leq n'$, $\mathbf{E}\left[\mathrm{HSE}(\mathbf{x}) \mid \mathrm{GHSE}_{n',k}(\mathbf{x}) = 1\right] = 10\sqrt{n'}$ and $\mathbf{E}\left[\mathrm{HSE}(\mathbf{x}) \mid \mathrm{GHSE}_{n',k}(\mathbf{x}) = 0\right] = -10\sqrt{n'}$. More specifically, let each player specify independent hash functions for every $\textsc{Sum-Equal}_k$ instance, and send the majority of those hash values to the referee. The referee can guess the input and corresponding hash value of any specific $\textsc{Sum-Equal}_k$ instance, such that the conditional distribution of the majority of hash values has a $\Theta\left(1/\sqrt{n''}\right)$ bias under correct guesses. Therefore by taking $n' = \Theta\left(n''\right)$ independent instances of the majority of hash values and conditioned on the correctness of the guesses, the expected number of agreements of the majority and the guessed hash value has a gap of $\Theta\left(n'/\sqrt{n''}\right) = \Theta\left(\sqrt{n'}\right)$, while in both cases the variance is linear in n'. For convenience we shift $\mathrm{HSE}(\mathbf{x})$ to $\pm 10\sqrt{n'}$ by padding and hence the vector of majority instances becomes an input to $\textsc{Gap-Hamming}_{n'}$.

For $\mathbf{RCC}^{sim}_{k,0.1}\left(\textsc{Aug-Index-Sum-Equal}_k^{n''}\right)$, i.e., k-player 0.1-error simultaneous communication complexity of $\textsc{Aug-Index-Sum-Equal}_k^{n''}$, the lower bound follows Theorem 13.

▶ **Theorem 13.** *Let Π be an δ-error randomized simultaneous communication protocol for $\textsc{Aug-Index-Sum-Equal}_k^{m'}$, where $m' \leq \frac{k\log\log k}{20\log k}$ and the error tolerance $\delta < 1/6$. Then Π must have simultaneous communication cost $\mathbf{RCC}^{sim}_{k,\delta}(\Pi) = \Omega\left(m'k\log\log k\right)$. Furthermore, the lower bound holds when the inputs to the $\textsc{Sum-Equal}_k$ problems are drawn from $\left([a]^{m'}\right)^{k-1} \times [\pm ka]^{m'}$ and the sum of inputs to each copy of $\textsc{Sum-Equal}_k$ is promised to be 0 or q, where $a = O\left(\log k\right)$ and $q = 2^{O(a)} \leq k^{1/8}$ is a multiple of all integers in $[a]$.*

Here we present the proof intuition of Theorem 13, while the proof appears in the full paper. Suppose that in a simultaneous communication protocol, a player P_1 encodes multiple instances of $\textsc{Sum-Equal}_k$ independently in a message, say t_1 bits for $\textsc{Sum-Equal}_k^{(1)}$, t_2 bits for $\textsc{Sum-Equal}_k^{(2)}$, and so on. Then many $\textsc{Sum-Equal}_k$ instances will be irrecoverable if the message length $\sum_{i=1}^{m'} t_i$ is significantly less than necessary for handling m' instances in parallel, say $\sum_{i=1}^{m'} t_i \leq 0.1 \cdot m' \cdot \mathbf{RCC}^{sim}_{k,\delta}\left(\textsc{Sum-Equal}_k\right)/k$, which means the $\textsc{Aug-Index-Sum-Equal}_k^{m'}$ cannot be solved with small error. Of course the full argument is much more involved, since the information in different $\textsc{Sum-Equal}$ instances can be combined in the message, which we deal with via a dedicated rectangle argument for conditional distributions. Combining (1), (2), (3), and Theorem 13, we get $\mathbf{RCC}^{sim}_{k,1/3}\left(\mathrm{T}_\varepsilon\right) \geq \Omega\left(tn''k\log\log k\right)$. Recalling Theorem 4.1 of [2] and $t = \Theta\left(\log n\right) = \Omega\left(\log N\right)$, $n'' = \Theta\left(n'\right) = \Theta\left(\varepsilon^{-2}\right)$, $mM = \mathsf{poly}(k)$, we conclude that $\overrightarrow{\mathbf{RSC}}_{k,1/3}\left(\mathrm{T}_\varepsilon\right) = \Omega\left(\varepsilon^{-2}\log(N)\log\log(mM)\right)$. ◀

References

1 Farid Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theoretical Computer Science*, 157(2):139–159, 1996.

2 Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. New Characterizations in Turnstile Streams with Applications. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 20:1–20:22, 2016.

3 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, June 2004.

4 Paul Beame, Toniann Pitassi, Nathan Segerlind, and Avi Wigderson. A Direct Sum Theorem for Corruption and the Multiparty NOF Communication Complexity of Set Disjointness. In *20th Annual IEEE Conference on Computational Complexity (CCC 2005), 11-15 June 2005, San Jose, CA, USA*, pages 52–66, 2005.

5 Mark Braverman and Ankit Garg. Public vs Private Coin in Bounded-Round Information. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 502–513, 2014.

6 Joshua Brody, Harry Buhrman, Michal Koucký, Bruno Loff, Florian Speelman, and Nikolay K. Vereshchagin. Towards a Reverse Newman's Theorem in Interactive Information Complexity. *Algorithmica*, 76(3):749–781, 2016 (also CCC 2013).

7 Graham Cormode, Mayur Datar, Piotr Indyk, and S. Muthukrishnan. Comparing Data Streams Using Hamming Norms (How to Zero In). *IEEE Trans. Knowl. Data Eng.*, 15(3):529–540, 2003.

8 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, February 26-28, 2009, Freiburg, Germany, Proceedings*, pages 505–516, 2009.

9 Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006.

10 T. S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of AND. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques. APPROX '09 / RANDOM '09, Berkeley, CA, USA, August 21 - 23, 2009*, pages 562–573, 2009.

11 Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff. Fast moment estimation in data streams in optimal space. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 745–754, 2011.

12 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 41–52, 2010.

13 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the Exact Space Complexity of Sketching and Streaming Small Norms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1161–1178, 2010.

14 Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.

15 Christos H. Papadimitriou and Michael Sipser. Communication complexity. *Journal of Computer and System Sciences*, 28(2):260–269, 1984.

16 Emanuele Viola. The communication complexity of addition. *SODA*, 2013.

17 David P. Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 941–960, 2012.