Electronic Colloquium on Computational Complexity, Report No. 175 (2024)

Optimality of Frequency Moment Estimation

Mark Braverman
Princeton University

Or Zamir
Tel Aviv University

Abstract

Estimating the second frequency moment of a stream up to $(1 \pm \varepsilon)$ multiplicative error requires at most $O(\log n/\varepsilon^2)$ bits of space, due to a seminal result of Alon, Matias, and Szegedy. It is also known that at least $\Omega(\log n + 1/\varepsilon^2)$ space is needed. We prove an optimal lower bound of $\Omega\left(\log\left(n\varepsilon^2\right)/\varepsilon^2\right)$ for all $\varepsilon = \Omega(1/\sqrt{n})$. Note that when $\varepsilon > n^{-1/2+c}$, where $c > 0$, our lower bound matches the classic upper bound of AMS. For smaller values of ε we also introduce a revised algorithm that improves the classic AMS bound and matches our lower bound. Our lower bound holds also for the more general problem of p-th frequency moment estimation for the range of $p \in (1, 2]$, giving a tight bound in the only remaining range to settle the optimal space complexity of estimating frequency moments.

1 Introduction

An extensive body of literature is devoted to the streaming model of computation, which is important for the analysis of massive datasets and in network traffic monitoring. A central problem in this model is the *frequency moment estimation* problem: Elements from a universe U are given to the algorithm one-by-one, defining a vector of frequencies — that is, $f_x \in \mathbb{N}$ is the number of times the element $x \in U$ appeared in the stream; Finally, the algorithm has to return, with good probability, a $(1 \pm \varepsilon)$-estimation of $F_p := \sum_{x \in U} f_x^p$ — the *p-th frequency moment* of the stream. We generally denote the length of the stream by n and assume that $|U| = \text{poly}(n)$. The main complexity parameter studied in this model is how much *space* is needed for the algorithm to succeed. The study of both the streaming model and of frequency moment estimation in it was initiated in the seminal 1996 work of Alon, Matias, and Szegedy [AMS96].

The case of $p = 2$, or *second moment estimation*, is of particular importance. It is often called the *repeat rate* or *surprise index*, and is used in various tasks such as database query optimization [AGMS99], network traffic anomaly detection [KSZC03], approximate histogram maintenance [GGI$^+$02] and more. Other moments of particular interest are $p = 1$, corresponding to the *approximate counting* problem [Mor78, NY22], and $p = 0$, corresponding to the *distinct elements* problem [FM85, IW03, KNW10b]. Among these special cases, only the space complexity of the first remains not fully understood. The original algorithm for F_2-estimation given by Alon, Matias, and Szegedy uses $O(\log n/\varepsilon^2)$ bits of space; while the highest known lower bound due to Woodruff in 2004 [Woo04] is $\Omega(\log n + 1/\varepsilon^2)$ — leaving up to a quadratic gap between the upper and lower bounds for certain choices of ε.

While F_p-estimation for $p \leq 2$ uses amount of space that is only logarithmic in the length of the stream, it was shown that for $p > 2$ at least $\Omega(n^{1-2/p}/\text{poly}(\varepsilon))$ space is needed [BYJKS04, CKS03b] — which is polynomial in the stream's length. A long list of works [IW05, BGKS06, MW10, AKO11, BO10, And17, Gan11b, WZ12, Gan11a, LW13] resulted in a nearly-tight bound of $\tilde{\Theta}\left(n^{1-2/p}/\varepsilon^2\right)$ for F_p-estimation for every $p > 2$ (not necessarily an integer) and ε, for some ranges of parameters the bounds are tight — in others there is a gap between the bounds that is poly-logarithmic in the bound itself.

For $p \leq 2$ the space complexity is not as well understood. Woodruff [Woo04] showed a lower bound of $\Omega(\log n + 1/\varepsilon^2)$ for every $p \neq 1$, this is optimal in terms of ε alone and is also known to be optimal for the distinct elements problem (that is, $p = 0$). For the special case of approximate counting (that is, $p = 1$), a tight bound of $\Theta(\log\log n + \log\varepsilon^{-1})$ is known [NY22]. For the range of $p \in [0, 1)$, the upper bound of

ISSN 1433-8092

$0 \leq p < 1$	$p = 1$	$1 < p \leq 2$	$p > 2$
$\tilde{\Theta}(\log n + 1/\varepsilon^2)$	$\Theta(\log \log n + \log \varepsilon^{-1})$	$\Theta(\log n/\varepsilon^2)$	$\tilde{\Theta}(n^{1-2/p}/\varepsilon^2)$

Figure 1: Space complexity of F_p-estimation.

AMS was improved by Jayaram and Woodruff who presented a nearly-tight $\tilde{O}\left(\log n + 1/\varepsilon^2\right)$ bound in that range [JW19]. This leaves $p \in (1, 2]$ as the last remaining range within no nearly-tight bounds are known.

For certain generalizations more is known: When the stream is *randomly shuffled* and given in random order, then in the range $p \in (1, 2)$ (excluding $p = 2$) [BVWY18] showed an improved upper bound of $\tilde{O}\left(\log n + 1/\varepsilon^2\right)$. When *updates* are allowed in the stream, that is, elements can also be deleted and not only added to it, then [KNW10a] showed that $\Theta(\log n/\varepsilon^2)$ is optimal for $p \leq 2$.

In this work, we settle the space complexity of frequency moment estimation in the entire remaining range of $p \in (1, 2]$, including the special case of second frequency moment estimation. For $p = 2$, we show that the AMS bound is essentially tight.

Theorem. *Let $\mathcal{A}$ be a streaming algorithm that gives an $(1 \pm \varepsilon)$ multiplicative approximation to the F_2 of its input stream and succeeds with probability $\geq 2/3$, for some $\varepsilon = \Omega(1/\sqrt{n})$. Then, the space used by $\mathcal{A}$ is $\Omega\left(\log\left(\varepsilon^2 n\right)/\varepsilon^2\right)$.*

Note that the range $\varepsilon < 1/\sqrt{n}$ is less interesting as $O\left(\min\{n \log n, |U|\}\right)$ space suffices for exactly maintaining the vector of frequencies. We observe that in the range where ε is very close to $1/\sqrt{n}$ our lower bound is (slightly) lower than the AMS upper bound, we show that this is inherent by introducing a modification of the AMS algorithm that matches our lower bound in this range.

Theorem. *For $\varepsilon = \Omega(1/\sqrt{n})$, we can get a $(1 \pm \varepsilon)$-approximation of the F_2 of a stream of length n using $O\left(\log\left(\varepsilon^2 n\right)/\varepsilon^2\right)$ space with success probability $> 2/3$.*

We also extend our lower bound to the range $p \in (1, 2]$, which settles the space complexity of F_p-estimation for all values of p. See Figure 1 for a summary of the space complexity of F_p-estimation for all $p \geq 0$.

Theorem. *Fix $p \in (1, 2]$. Let $\mathcal{A}$ be a streaming algorithm that gives an $(1 \pm \varepsilon)$ approximation to the F_p of its input stream, for some $\varepsilon \in \left(\Omega\left(n^{-1/p}\right), 4^{-1/(p-1)}\right)$, and succeeds with probability $\geq 2/3$. Then, the space used by $\mathcal{A}$ is $\Omega\left(\log\left(\varepsilon^{1/p} n\right)/\varepsilon^2\right)$.*

Most of the lower bounds for streaming problems are based on reductions from *communication complexity*. In [JW19], a natural barrier to prove a better than $\tilde{\Omega}(1/\varepsilon^2)$ lower bound was shown: Even in a very strong model of communication, $O\left(1/\varepsilon^2 \cdot \left(\log \log n + \log d + \log \varepsilon^{-1}\right)\right)$ bits of communication suffice for the players to correctly produce a F_p estimation, where d is the diameter of the communication graph. This means that problems who reduce to F_p-estimation have a too low communication complexity to improve the existing lower bounds. To overcome this natural barrier, we present a new type of a *direct sum theorem* that takes place at the level of the streaming algorithm rather than the level of the communication model — informally, we pack many instances of problems with communication complexity $\Theta(1/\varepsilon^2)$ into a single stream, and then directly show that a successful streaming algorithm must solve them all. In Section 2 we give a detailed high-level overview of our proofs. The lower bound is presented in Section 4 and Section 5, and then extended from $p = 2$ to $p \in (1, 2]$ in Section 5.3. The improved upper bound is presented in Section 6. We conclude and present remaining open problems in Section 7.

2 High-Level Overview

In this Section we give a high-level overview of the components used in our lower and upper bounds. The lower bound is then proven in Section 4 and Section 5, and the upper bound in Section 6.

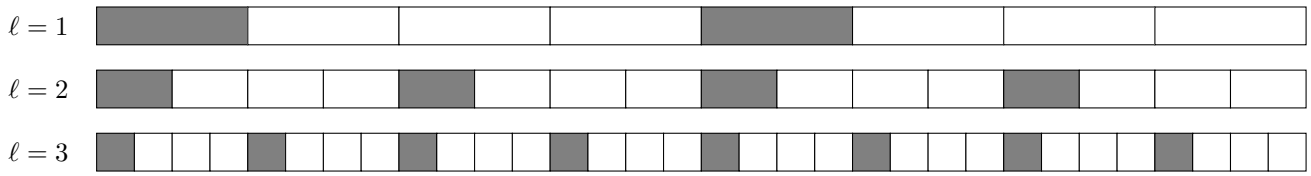

Figure 2: The parts of the stream corresponding to the players of a 2^ℓ-party Exam Disjointness instance, for $\ell = 1, 2, 3$.

2.1 Exam Communication Model

As in many other streaming lower bounds, our initial building blocks are reductions from communication complexity problems. However, instead of using the classic communication model, we consider a slight variant which we call *the exam model*. In the classic numbers-in-hand communication model, t-players receive different parts $x_1, x_2, \ldots, x_t$ of an input and have to compute some function $f(x_1, \ldots, x_n)$ of the entire input by communicating with each other. In our model, we introduce an additional player, *a referee*, who has an additional input y which we think of as a *question* about the players' inputs $\{x_i\}$. The players still receive their parts of the input and are allowed to communicate with each other, but not with the referee; finally, a player sends a single message to the referee, and the referee then has to compute some function $g(x_1, \ldots, x_t, y)$ of *both* the input and her "secret" question y.

In the classic *Disjointness* problem, each of the t players receives a set S_i and they have to determine whether or not all of their input sets are pairwise disjoint. It is known that solving this problem remains hard even under the promise that the input sets are either all disjoint or contain a unique element appearing in all input sets and do not intersect further. We introduce a variant we call *Exam Disjointness* in which each of the players still receives a set S_i, and the referee receives a single element y from the universe; based on the message she receives the referee needs to decide whether y appears in *all* input sets S_i. In other words, instead of determining whether there was a unique intersection between their input sets, the players now have to send enough information to determine if a specific y given to the referee is that unique intersection. In Section 4 we prove that solving Exam Disjointness requires at least as much information about the input as the standard Disjointness problem.

This communication model turns out to be very useful for reducing a communication problem to a one-pass streaming algorithm problem. Intuitively, only the suffix a of the stream would depend on the referee's input y, which would in turn mean that the algorithm processing the rest of the stream has to learn enough information to answer any possible question of the referee that might appear in the exam. In Section 5.1, we use the Exam Disjointness problem to recover the existing $\Omega(1/\varepsilon^2)$ lower bound for F_2-estimation, with a simple and short proof.

2.2 Direct Sum for Dependent Instances

A standard tool for proving lower bounds is proving *a direct sum theorem*; this means showing that solving several *independent* instances of a certain problem is as hard as solving every one of them separately. In our proof, we introduce an unintutive variant of a direct sum theorem in which several instances *are* dependent, and nevertheless solving them all is as hard as solving them separately.

As stated above, we first show that we can reduce the t-party Exam Disjointness problem to estimating the F_2 of a certain stream. We then introduce a distribution over input stream that encodes within it instances of t-party Exam Disjointness for multiple different values of t. See Figure 2 for an example of a stream in which Exam Disjointness instances with two, four, and eight players are encoded, using different parts of the stream to encode each player; while at the same level all players get a disjoint part of the stream — players at different levels occasionally intersect. We would describe a construction containing roughly $\log(\varepsilon^2 n)$ such levels, such that solving the Exam Disjointness instance corresponding to a single level requires $\Omega(1/\varepsilon^2)$ bits of space. If instances of different levels were disjoint, a direct sum theorem would

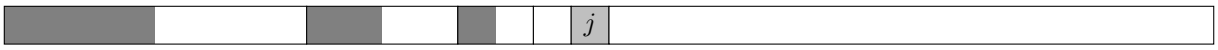

Figure 3: An index j and its preceding players from several different levels.

yield our desired $\Omega\left(\log\left(\varepsilon^2 n\right)/\varepsilon^2\right)$ lower bound; we obtain a similar bound despite the intersection of different levels.

The core idea of this part is the following observation: Consider an average index j within the stream; at each level, look at the part of the stream that encodes the last player appearing before index j in that level. Although some of these players might intersect each other, usually there is a large subset of them who are pairwise disjoint. See Figure 3 for an illustration. At index j we must have information about each level, as otherwise we would not be able to finally solve the instance of that level. For the levels in which the last players are pairwise disjoint, we would be able to show that the information known about each of them by j is independent. This would be enough to prove a "local" direct sum for only a subset of the levels at each index j, and an average over all indices would then give us a bound as good as a "global" direct sum. The exact details are given in Section 5.2.

Another way to look at the direct sum across levels is that at each level ℓ we get a lower bound on the amount of information the stream typically needs to store about the $\sim 2^\ell$ preceding stream elements. It is then shown that these lower bounds add up, as these different-scaled "pasts" are essentially disjoint. This multi-scale phenomenon is arguably the "real reason" behind the $\log n$ multiplicative factor in the lower bound. A similar phenomenon was used to show that estimating the majority of n random bits requires $\Omega(\log n)$ memory [BGW20].

2.3 Improved Algorithm for The Small Error Regime

The classic algorithm of AMS gives us a bound of $O(\log n/\varepsilon^2)$ for $(1\pm\varepsilon)$-estimating the F_2 of a stream. Evidently, this matches our $\Omega\left(\log\left(\varepsilon^2 n\right)/\varepsilon^2\right)$ lower bound only when ε is polynomially larger than $n^{-1/2}$. In Section 6 we introduce a slight variant of the AMS algorithm that matches our lower bound. The modification is rather simple: we randomly partition the universe of elements into roughly $1/\varepsilon^2$ disjoint subsets, and then run the AMS algorithm separately (and simultaneously) on the subsets of the stream contained in each of these parts. Each subset of the stream only contains around $\varepsilon^2 n$ elements, which would result in the improved bound. A similar modification was used before in algorithms aiming to reduce the memory-probes-per-update complexity of the AMS algorithm [TZ04].

3 Mutual Information and Streaming Algorithms

We frequently use the notions of information complexity, mutual information and entropy, see [BYJKS04] for example for more background on these notions.

A *one-pass* streaming algorithm receives a sequence of inputs $X=(X_1,\ldots,X_n)$ one by one, and eventually outputs an answer. Denote by $M_1,\ldots,M_n$ the *memory transcript* of the streaming algorithm. That is, M_i is the memory state of the algorithm immediately after receiving the i-th input X_i. We observe that M_i is drawn from a distribution that depends solely on M_{i-1} and X_i. From now on, we think of the inputs as a distribution, thus every X_i and every M_i is a random variable.

The following classic observation highlights the benefit of studying the mutual information between the input distribution and the memory transcript.

Observation 3.1. *The number of memory bits used by a streaming algorithm is at least*

$$\max_j \left(I\left(M_j\ ;\ X\right)\right).$$

Proof. It follows immediately as $I(M_j\ ;\ X)\le H(M_j)\le \log\left|\text{Support}\left(M_j\right)\right|$. □

We next introduce a useful lemma.

Lemma 3.2. *If the inputs X_i are mutually independent, then*

$$I(M\ ;\ X) = \sum_{i=1}^{n} I\left(X_i\ ;\ M_i \mid M_{i-1}\right).$$

Proof. Consider the following chain of equalities,

$$\begin{aligned} I(X_1,\ldots,X_n\ ;\ M_1,\ldots,M_n) &= \sum_{i=1}^{n} I\left(X_1,\ldots,X_n\ ;\ M_i \mid M_{<i}\right) && (1)\\ &= \sum_{i=1}^{n} \left(I\left(X_i\ ;\ M_i \mid M_{<i}\right) + I\left(X\ ;\ M_i \mid M_{<i}, X_i\right)\right) && (2)\\ &= \sum_{i=1}^{n} \left(I\left(X_i\ ;\ M_i \mid M_{i-1}\right) + 0\right). && (3) \end{aligned}$$

Inequalities (1) and (2) are applications of the chain rule of mutual information. The second part of Inequality (3), that is $I\left(X\ ;\ M_i \mid M_{<i}, X_i\right) = 0$, follows as X is independent of the algorithm's random coins while M_i is fully determined by $M_{<i}, X_i$ and these random coins. The first part of Inequality (3), that is $I\left(X_i\ ;\ M_i \mid M_{<i}\right) = I\left(X_i\ ;\ M_i \mid M_{i-1}\right)$, follows as to both X_i and M_i conditioning on $M_{<i}$ or on M_{i-1} is equivalent, this uses the independence of X_i from $X_{<i}$. □

4 Exam Variant of Multi-Party Set Disjointness

At the core of our lower bound is a variant of the Multi-Party Set Disjointness problem (denoted DISJ from now on). The DISJ_t communication problem is defined as follows. There are t players, each player $i \in [t]$ receives a set S_i of elements from some universe U; We are promised that either all sets are disjoint, or otherwise there is exactly one element that appears in all of the sets and beside it all other elements are disjoint; the players need to decide in which of the two cases their input sets are. We focus on the *one-way communication model* in which there is a predetermined order of the players; In its turn, a player can look at its own input set and on the communication it received from the previous player, work for an unbounded amount of time, and then send a communication to the next player; The last player needs to answer whether the sets were disjoint or contained a shared element. Chakrabarti et al. [CKS03a] proved that the total length (in bits) of the message communicated by the players throughout the protocol must be at least $\Omega\left(\frac{1}{t}\sum_{i=1}^{t}|S_i|\right)$ if they succeed with good probability. Intuitively, this follows as otherwise the total amount of communicated information is smaller than even a single player's set, and thus even a single set cannot be fully described throughout the protocol's transcript. Their lower bound is optimal and holds even in a stronger communication model called *the blackboard model* in which each player can see the messages communicated by all previous players, rather than just the preceding one. The same lower bound holds even in the stronger model of communication in which players can talk several times in arbitrary order and not only in a one-way fashion [Gro09, Jay09].

For our needs, we introduce a slight variant of the one-way DISJ_t problem which we call *Exam DISJ_t* and denote by EDISJ_t. In this version, we still have t players with input sets S_i that can be disjoint or have a unique common intersection, but we also have an additional special player, which we call *the referee*. The referee receives as input a single universe element $x \in U$. The t players still communicate one by one according to order, and the last player communicates a message to the referee. Then, the referee has to decide (and succeed with probability $\geq 2/3$) whether there was an intersection between the players' sets *and* that intersection is exactly the referee's input x. We think of this as an "exam" for the players: If they claim there was an intersection, they also need to know enough about the intersection to tell whether or not

it is x. In this section, we extend the lower bound for DISJ_t and prove that EDISJ_t also requires as much communication.

Let $\mathcal{P}$ be a protocol that solves EDISJ_t, and denote by $\Pi = \Pi(X)$ its *transcript* on an input $X = (S_1, S_2, \ldots, S_t, x)$; the transcript consists of the message communicated by each of the t players during its turn to speak. Note that Π is a random variable that depends only on $X' := (S_1, \ldots S_t)$, as the referee does not speak and thus x does not affect the transcript.

We define an input distribution for the players of EDISJ_t. Let m be a size parameter, and let U be a universe of size $|U| \geq m^4$. Each input set S_i is drawn randomly and independently as a subset of U of size $\lceil \frac{m}{t} \rceil$. We note that $m \leq \sum_{i=1}^{t} |S_i| < m + t$ and that with high probability all t sets are disjoint. Denote this distribution of X' by $\mu = \mu(t, m, U)$.

In this section, we prove the following lower bound on the information complexity of EDISJ_t on the distribution μ. We also assume that $t < \frac{m}{4 \ln m}$.

Theorem 4.1. *Let $\mathcal{P}$ be a protocol that solves $EDISJ_t$ with success probability $\geq 2/3$ for any input with $|S_i| \leq \lceil \frac{m}{t} \rceil$ for $i \in [t]$. Let $X' = (S_1, \ldots, S_t) \sim \mu$ and denote by $\Pi = \Pi(X')$ the transcript of $\mathcal{P}$ on X'. Then,*

$$I(X' \,;\, \Pi) = \Omega\left(\frac{m}{t}\right).$$

The proof of Theorem 4.1 is similar to standard lower bound proof for multi-party set disjointness. It proceeds in two steps: first, derive an appropriate information bound for the AND function – which is the one-bit version of disjointness; second, apply a direct sum argument to obtain the lower bound statement.

4.1 AND Lower Bounds

All proofs for the hardness of the standard Disjointess problem go through a direct sum between instances of the AND problem [BYJKS04, CKS03b, Jay09]. In the AND^t communication problem, there are t players and each receives a single input bit $Y_i \in \mathbb{F}_2$; the players need to compute the "and" function $\bigwedge_{i=1}^{t} Y_i$. Also denote by $Y = (Y_1, \ldots, Y_t)$. In the DISJ_t problem, we may think of each player's input as the characteristic vector of its set $S_i \subset U$, and denote its coordinates by $Y_{i,u} = 1$ if $u \in S_i$ and otherwise 0. Then, the answer to DISJ_t is simply $\bigvee_{u \in U} \bigwedge_{i=1}^{t} Y_{i,u}$. Due to this structure, techniques introduced by [BYJKS04] show it suffices to give a lower bound to the information complexity of AND^t to imply a lower bound for DISJ_t. Consider the probability distribution ν over inputs to the AND^t problem that gives probability $\frac{1}{2}$ for all input bits being 0, and probability $\frac{1}{2}$ for exactly one coordinate chosen at random being 1 and the rest 0. Let Π be the transcript of a communication protocol that solves AND^t with probability $> 2/3$, the citations above show that $I(Y \,;\, \Pi) = \Omega\left(\frac{1}{t}\right)$. Denote by Π_t the last message communicated in the protocol Π, from which the answer has to be deduced (in a one-way communication setting this is simply the message sent by the last player). In slightly more detail, they show that as long as there is a non-negligible statistical difference between the distributions of Π_t when the input is 0^t and when the input is 1^t (which there must be for solving AND^t correctly), then the above information bound holds.

Lemma 4.2 (e.g., Corollary 8 of [Jay09][1])**.** *If $\|\Pi_t(1^t) - \Pi_t(0^t)\|_1 > 0.1$, then $I_\nu(Y \,;\, \Pi) = \Omega\left(\frac{1}{t}\right)$.*

For our proof, we need a slight variant of Lemma 4.2. Let $p \leq \frac{1}{t}$ be some probability. We consider a distribution $\mu = \mu_p$ in which every input bit Y_i is independently chosen to be 1 with probability p and otherwise 0. We prove the following adaptation of Lemma 4.2. While it is straightforward to deduce it by repeating any of the previous proofs of Lemma 4.2, we give a proof that uses them in a black-box manner.

Lemma 4.3. *Suppose $\|\Pi_t(1^t) - \Pi_t(0^t)\|_1 > 0.1$, then $I_\mu(Y; \Pi) = \Omega(p)$.*

Proof. We observe that $\mu(0^t) = (1-p)^t \geq \left(1 - \frac{1}{t}\right)^t \geq \frac{1}{4}$, and that for every $i \in [t]$ also $\mu(e_i) = p \cdot (1-p)^{t-1} \geq \frac{1}{4t}$. Therefore, we may define a Boolean random variable D such that: (1) $\Pr[D = 1] = \Theta(p \cdot t)$;

[1] In [Jay09] this statement is phrased in terms of the Hellinger distance between the two distributions rather than the total variation distance, but in the constant regime the two distances are equivalent; See for example Proposition A.2 in [BYJKS04].

(2) $\mu|_{D=1} = \nu$. That is, conditioned on the event D happening, μ becomes ν. Note that D is independent of Π. Using Lemma 4.2, we get

$$\begin{aligned} I_\mu(Y;\Pi) &= I_\mu(Y, D\ ;\ \Pi) - I_\mu(D;\Pi|Y) = I_\mu(Y, D\ ;\ \Pi) = I_\mu(D;\Pi) + I_\mu(Y;\Pi|D) \\ &= I_\mu(Y;\Pi|D) \geq \Pr[D=1] \cdot I_\mu(Y;\Pi|D=1) = \Pr[D=1] \cdot I_\nu(Y;\Pi) \\ &\geq \Theta(p \cdot t) \cdot \Omega(1/t) = \Omega(p). \end{aligned}$$

□

We denote the problem from Lemma 4.3 with this distribution μ by AND_p^t.

4.2 Proof of Theorem 4.1

For the remainder of the proof set

$$p := \frac{m}{2t|U|}.$$

Let $Y := \{Y^j\}_{j=1}^{|U|}$ be $|U|$ instances of inputs to AND^t. Let $Y_i := (Y_i^j)_{j=1}^{|U|}$ be the i-th player's input, which consists of the i-th player's inputs in all of the $|U|$ instances. Given a protocol Π that solves EDISJ_t, we construct a protocol $\Pi'(Y_1, \ldots, Y_t)$ that we will then use to solve all of the instances of AND^t defined above:

1. Each player $i \in [t]$ constructs an input set S_i as follows:
   (a) Set $S_i' := \{j \in U :\ Y_i^j = 1\}$;
   (b) If $|S_i'| > \lceil \frac{m}{t} \rceil$, send 'Fail', and the protocol fails;
   (c) Otherwise, set $S_i \subset U$ by adding $\lceil \frac{m}{t} \rceil - |S_i'|$ random elements from $U \setminus S_i'$ to S_i'.
2. Run $\Pi(S_1, \ldots, S_t)$ and return the message sent to the referee as the output Π_t'.

We first observe that if every instance Y^j is drawn from μ_p then w.h.p no player will 'Fail'.

Lemma 4.4. *When every Y^j is independently distributed as μ_p, then the probability that any S_i is larger than m/t is at most $te^{-m/(2t)} < \frac{1}{4m \ln m}$.*

Proof. The size of S_i is distributed as $\mathrm{Binomial}(|U|, \frac{m}{2t|U|})$ and therefore a Chernoff bound gives that

$$\Pr\left(|S_i| > \frac{m}{t}\right) \leq e^{-m/(2t)}.$$

We then take a union bound over the t indices. □

Then, we also observe that the protocol's output Π_t' can be used to answer each of the AND^t instances.

Lemma 4.5. *Let $j \in U$ be a universe element. Let Y^j be an input to the AND^t problem. If for every $u \neq j$ we draw Y^u independently from μ_p and then run the above protocol Π' on $Y_1, \ldots, Y_t$, then Π_t' implies the correct solution to $\mathrm{AND}^t(Y^j)$ with probability $> 2/3 - o(1)$.*

Proof. Using Π_t', we may simulate the referee's part of the EDISJ_t protocol with the choice of j as the exam question. This correctly computes $\mathrm{AND}^t(Y^j)$ as long as no player mistakenly added j to its set in line (2) of the reduction, or if the reduction returned 'Fail'. Nonetheless, the former happens with probability $\leq p \cdot m = o(1)$ and the latter happens with probability $\leq \frac{1}{4m \ln m} = o(1)$. □

For any $j \in U$, Lemma 4.5 implies that Π'_t conditioned on $Y^j = 0^t$ must be at statistical distance $> 1/3 - o(1)$ from Π'_t conditioned on $Y^j = 1^t$. Therefore, by Lemma 4.3 we have that when $Y^j \sim \mu_p$ (like all other instances Y^u) then

$$I(Y^j; \Pi') = \Omega(p) = \Omega\left(\frac{m}{t|U|}\right).$$

Thus if every $Y^j \sim \mu_p$ independently, we also have

$$I(Y; \Pi') = \sum_{j \in U} I(Y^j; \Pi' | Y^{<j}) = \sum_{j \in U} I(Y^j \ ; \ \Pi', Y^{<j}) \geq \sum_{j \in U} I(Y^j; \Pi') = \Omega\left(\frac{m}{t}\right).$$

Let F denote the Boolean random variable such that $F = 1$ if Π' fails (that is, any player outputs 'Fail'). We observe that conditioned on $F = 0$, the distribution of $X' := (S_1, \ldots, S_t)$, the input to the portocol Π, is exactly $\mu = \mu(t, m, U)$. Since Π runs on the input X' when we do not fail, then $I(Y; \Pi' \mid X', F = 0) = 0$ as there is no additional information about Y seen by the protocol besides X'. We therefore have,

$$\begin{aligned} I_\mu(\Pi; X') &= I(\Pi'; X' | F = 0) = I(\Pi'; X', Y | F = 0) - I(\Pi'; Y | X'; F = 0) \\ &= I(\Pi'; X', Y | F = 0) \geq I(\Pi'; Y | F = 0) \geq I(\Pi'; Y) - H(F) - \Pr[F = 1] \cdot I(\Pi'; Y | F = 1) \\ &\geq \Omega\left(\frac{m}{t}\right) - 1 - t \exp\left(-\frac{m}{2t}\right) \cdot m \log |U| = \Omega\left(\frac{m}{t}\right). \end{aligned}$$

The last inequality holds since even in the failure event, each player who transmits anything but 'Fail' only holds at most m/t elements, bounding the total entropy of their inputs by $m \log |U|$.

□

5 Lower Bound

We begin with Section 5.1, in which we show how to reduce an instance of EDISJ to the problem of F_2 estimation on a stream. This reduction already gives a simple proof of Woodruff's $\Omega(1/\varepsilon^2)$ lower bound [Woo04] in the entire range of $\varepsilon = \Omega(1/\sqrt{n})$. Then, in Section 5.2 we show that we can pack $\Theta\left(\log\left(\varepsilon^2 n\right)\right)$ instances of EDISJ into a *single* stream of length n. Solving each single instance would require $\Omega\left(\frac{1}{\varepsilon^2}\right)$ space; We prove that solving *all* $\Theta\left(\log\left(\varepsilon^2 n\right)\right)$ instances requires $\Omega\left(\log\left(\varepsilon^2 n\right)/\varepsilon^2\right)$ memory bits, as much as it would take to solve each of them independently. Crucially, our instances are not going to be independent of each other, but to share certain stream elements. Finally, we prove the following, which is the main theorem of this paper.

Theorem 5.1. *Let $\mathcal{A}$ be a streaming algorithm that gives an $(1 \pm \varepsilon)$ approximation to the F_2 of its input stream and succeeds with probability $\geq 2/3$, for some $\varepsilon = \Omega(1/\sqrt{n})$. Then, the space used by $\mathcal{A}$ is $\Omega\left(\log\left(\varepsilon^2 n\right)/\varepsilon^2\right)$.*

In Section 5.3 we extend the found to F_p estimation for $p \in (1, 2]$.

5.1 Reducing EDISJ to F_2

Fix some n, $\varepsilon > \frac{2}{\sqrt{n}}$ and $|U| > n^3$. Denote by $t := \lfloor \varepsilon\sqrt{n} \rfloor \geq 2$, for simplicity we assume that t divides n. We reduce EDISJ_t to estimating F_2 on a stream of length $(1 \pm o(1))\, n$ up to error ε. Let $X = (S_1, \ldots, S_t, x)$ be a valid input to EDISJ_t, for simplicity we also assume that $|S_i| = \frac{n}{t}$ for every $i \in [t]$ (otherwise we may pad the sets). We define a stream $s(X)$ with elements from U that encodes X as follows: First, we write down the elements of S_1 in arbitrary order, afterwards we write down the elements of S_2, then of S_3 and so on until those of S_t; Finally, we write down x repeatedly $k := \lceil \frac{t}{\varepsilon} \rceil = \Theta(\sqrt{n})$ times. We note that the length of $s(X)$ is $\left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right) n$, and that the stream is naturally partitioned to $t + 1$ parts such that each part depends on the input of a single player (where the $(t+1)$-th 'player' is the referee).

Lemma 5.2. *A $(1 \pm \Theta(\varepsilon))$-approximation to $F_2(s(X))$ implies the answer to EDISJ$_t(X)$.*

Proof. If EDISJ$_t(X)$='No' then there are two possible cases: First, there is no intersection to the input sets, and thus x might appear in at most one of the sets; Hence, $F_2(s(X)) \leq (n-1)\cdot 1^2 + 1\cdot(1+k)^2 = n+k^2+2k$. Second, there is a unique intersection to all input sets, but this intersection is not x, who might still appear in at most one of the sets; Hence, $F_2(s(X)) \leq (n-t-1)\cdot 1^2 + 1\cdot t^2 + 1\cdot(1+k)^2 = n+k^2+2k+t^2-t$. On the other hand, if EDISJ$_t(X)$='Yes' then there is a unique intersection to all input sets who is also x, and thus $F_2(s(X)) = (n-t)\cdot 1^2 + (t+k)^2 = n+t^2-t+k^2+2tk$. Due to our choice of parameters, $F_2(s(X)) = O(n)$ in all cases; Furthermore, the additive gap between the 'Yes' case and the largest possible 'No' case is

$$2tk - 2k = 2k(t-1) \geq kt \geq \frac{t^2}{\varepsilon} \geq \varepsilon n.$$

□

Next, we observe that a streaming algorithm approximating $F_2(s(X))$ naturally implies a communication protocol for EDISJ$_t(X)$. Let $\mathcal{A}$ be a streaming algorithm that gives a $(1 \pm \Theta(\varepsilon))$-approximation to F_2, we define a communication protocol $\mathcal{P}$ as follows. The first player, who knows S_1, can construct the part of $s(X)$ corresponding to it and start running $\mathcal{A}$ on that prefix of the stream. Denote by M_1 the memory in $\mathcal{A}$ at the end of that part. The first player communicates M_1 to the second player, who has S_2 and can thus construct the second part of $s(X)$; as it also has the communicated M_1 it can now continue the run of $\mathcal{A}$ until the end of the second part of the stream and communicate the memory in its end, M_2, to the next player. We continue similarly until the t-th player communicates M_t to the referee, who has x and can thus construct the last part of $s(X)$ and complete $\mathcal{A}$'s run. The transcript of $\mathcal{P}$ is $\Pi := (M_1, \ldots, M_t)$, note that it depends only on $X' := (S_1, \ldots, S_t)$ and not on x, or equivalently, only on the first n elements of $s(X)$. Lemma 5.2 implies that if $\mathcal{A}$ correctly approximates F_2 with error $\Theta(\varepsilon)$ then $\mathcal{P}$ correctly solves EDISJ$_t$ with the same success probability. We conclude using Theorem 4.1 that over the distribution $X' \sim \mu(t, n, U)$ we have

$$I(X' ;\ \Pi) = \Omega\left(\frac{n}{t}\right) = \Omega\left(\frac{\sqrt{n}}{\varepsilon}\right).$$

Corollary 5.3. *For any $\varepsilon = \Omega(1/\sqrt{n})$, approximating F_2 up to multiplicative error $(1 \pm \varepsilon)$ requires $\Omega(1/\varepsilon^2)$ bits of space.*

Proof. The length of $\Pi = (M_1, \ldots, M_t)$ is at most t times the size M of the space used by the streaming algorithm $\mathcal{A}$ in bits. Thus,

$$\Omega\left(\frac{\sqrt{n}}{\varepsilon}\right) \leq I(X' ;\ \Pi) \leq H(\Pi) \leq tM,$$

and therefore

$$M \geq \Omega\left(\frac{\sqrt{n}}{\varepsilon t}\right) = \Omega\left(\frac{1}{\varepsilon^2}\right).$$

□

For the consecutive parts of our lower bound, we need a slightly more flexible reduction from EDISJ to F_2. In particular, we show that with a slight modification the above reduction yields the same lower bound even if we reduce from EDISJ$_t$ for any $2 \leq t \leq \varepsilon\sqrt{n}$ rather than exactly $t = \lfloor \varepsilon\sqrt{n} \rfloor$. Fix any such t. Denote by $d := \lfloor \frac{\varepsilon^2 n}{t^2} \rfloor \geq 1$. For simplicity, we assume that $d \cdot t$ divides n. We still use the universe U for the stream elements, but our EDISJ$_t$ instances are defined over the larger universe U^d of d-tuples of elements from U. We add the *promise* that we only need to solve instances of EDISJ$_t$ in which no two distinct elements of U^d appearing in the input intersect each other. We consider instances of EDISJ$_t$ in which the set sizes are $|S_i| = \frac{n}{dt}$. For an instance $X = (S_1, \ldots, S_t, x)$ the stream $s(X)$ is constructed almost identically to the previous construction, except that when we write an element of U^d on the stream we write

replace it with the d elements of U it consists of. We still repeat the referee's element $x \in U^d$ at the end of the stream for $k = \lceil \frac{t}{\varepsilon} \rceil$ times. Thus, the size of the stream corresponding to X' is still $t \cdot \frac{n}{dt} \cdot d = n$ and its elements are still elements from U. The size of the suffix of the stream corresponding to the referee is $k \cdot d = O\left(\frac{td}{\varepsilon}\right) = O\left(\frac{t\varepsilon^2 n}{\varepsilon t^2}\right) = O\left(\frac{\varepsilon n}{t}\right)$ which is at most $O(\varepsilon n)$. We next observe that Lemma 5.2 still holds: In both the 'Yes' and 'No' cases, we have $F_2\left(s\left(X\right)\right) \leq d \cdot \left(\frac{n}{d} + t^2 - t + k^2 + 2tk\right) = O\left(n\right)$. Furthermore, the gap between the 'Yes' case and the largest possible 'No' case is at least $d \cdot (2tk - 2k) \geq \Omega(dtk) \geq \Omega\left(\varepsilon n\right)$. We conclude as before that for $\Pi = (M_1, \ldots, M_t)$ and input distribution $X' \sim \mu\left(t, \frac{n}{d}, U^d\right)$ we have

$$I\left(X' \ ; \ \Pi\right) = \Omega\left(\frac{n/d}{t}\right) = \Omega\left(\frac{t}{\varepsilon^2}\right).$$

Remark 5.4. *This modification implies that the $\Omega(1/\varepsilon^2)$ lower bound follows even from the Exam version of the 2-party DISJ problem.*

5.2 Packing many EDISJ instances into a single F_2 instance

We now construct a stream of length n that encodes within it many instances of EDISJ. Fix some $n, \varepsilon > \frac{4}{\sqrt{n}}$ and $|U| > n^3$. For simplicity, assume that $\varepsilon^{-}1$ is a power of two and that n is a power of four (otherwise we may round to the nearest ones); In particular, $\varepsilon\sqrt{n}$ is also a power of two and divides n.

Let $\ell \in [1, \log\left(\varepsilon\sqrt{n}\right) - 2]$ be a *level* index. We construct a stream of length n that encodes an instance of EDISJ_t for $t = 2^\ell$, similar to that of Section 5.1. Denote by $d := \frac{\varepsilon^2 n}{4t^2}$. Denote by $X' = (S_1, \ldots, S_t), X = (X', x)$ a valid instance to EDISJ_t with set sizes $|S_i| = \frac{n}{4dt}$ and universe U^d. As before, the only valid inputs are those in which all sets $S_i \subset U^d$ are either disjoint or have a unique intersection, and furthermore every distinct elements of U^d appearing in the input do not intersect each other (i.e., each element of U is used in at most one element of U^d appearing in the input sets).

We start by partitioning the stream of length n into named parts. We partition the stream into $2^{\ell+2}$ consecutive blocks of size $\frac{n}{2^{\ell+2}}$ each. We call each such block *a bucket*; The buckets of indices divisible by four are called *active buckets* and the buckets of other indices are called *inactive buckets*, denote by $B_1^{(\ell)}, B_2^{(\ell)}, \ldots, B_{2^\ell}^{(\ell)}$ the t active buckets, each of them is a consecutive subsequence of the stream of length $n/2^{\ell+2} = n/4t$. Within each active bucket, we again partition the stream into consecutive blocks of size d each; We call each such sub-block *a super-element*. Every active bucket contains $\frac{n}{2^{\ell+2} \cdot d} = \frac{n}{4dt}$ super-elements. We refer the reader to Figure 2 for an illustration of the active buckets in several levels.

Reduction from DISJ_t:

We follow the reduction of Section 5.1 in our current terminology. Given a valid input X, we put the set S_i for $i \in [t]$ in place of the active bucket $B_i^{(\ell)}$ in the stream, where each element $x \in S_i \subset U^d$ fills an entire super-element within the active bucket. Every element of the stream not within an active bucket is drawn uniformly and independently from U. We append to the described stream a suffix of length $d \cdot k$, for $k = \frac{t}{\varepsilon}$, which we fill with k repetitions of x written as a super-element each time.

We observe that with high probability approximating the F_2 of the described stream up to error $\Theta(\varepsilon)$ also solves $\text{EDISJ}_t(X)$: This is because with high probability (at least $1 - \frac{1}{n}$) the random elements of U placed in the non-active buckets are completely disjoint to any element appearing in X, and thus subtracting $\frac{3}{4}n$ from the F_2 of the entire stream leaves us with the F_2 of the sub-stream containing only the active buckets — which is exactly the stream analyzed in Section 5.1.

Let $\mathcal{A}$ be an algorithm that estimates the F_2 of a stream up to $\Theta(\varepsilon)$ error. Let $0 < j_1 < j_2 < \ldots < j_t < n$ be any t indices such that j_i is located between the end of the i-th active bucket and before the beginning of the $(i+1)$-th active bucket (or before the referee's suffix of the stream when $i = t$). Denote by M_{j_i} the state of $\mathcal{A}$'s memory after processing the j_i-th stream element while running over the stream $S = S(X)$ we get from the reduction. Due to the stated above, $\Pi = (M_{j_1}, \ldots, M_{j_t})$ is the transcript of a communication protocol that solves $\text{EDISJ}_t(X)$.

Denote by ξ the distribution of a stream of length n where each element is drawn uniformly and independently from U. We observe that when $X' \sim \mu\left(t, \frac{n}{4d}, U^d\right)$, the resulting stream without the referee's suffix $S(X')$ is exactly distributed according to ξ. Thus, from Theorem 4.1 we conclude the following.

Corollary 5.5. *For indices $j_1 < j_2 < \ldots < j_t$ as above, we have*

$$I\left(\ \left(B_1^{(\ell)}, B_2^{(\ell)}, \ldots, B_t^{(\ell)}\right)\ ;\ \left(M_{j_1}, \ldots, M_{j_t}\right)\ \right) = \Omega\left(\frac{t}{\varepsilon^2}\right),$$

where the memory states of the algorithm $\mathcal{A}$ and the active buckets are with respect to a stream drawn from ξ.

Information measure for a single level ℓ:

Following the above, we define an information measure that encapsulates how much information about the active buckets of level ℓ is given by the memory of the algorithm in an average index. From now on, we only consider the input stream distribution ξ for $\mathcal{A}$, and denote the stream itself by $X = (X_1, \ldots, X_n) \sim \xi$.

Definition 5.6. *For level ℓ, denote by*

$$I_\ell := \sum_{j=1}^{n} I\left(X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}\ ;\ M_j\ \mid\ M_{\left(j-\frac{n}{2^\ell}\right)}\right).$$

We next give a lower bound for I_ℓ, based on Corollary 5.5.

Lemma 5.7. $I_\ell > \Omega\left(\frac{n}{\varepsilon^2}\right)$.

Proof. Let j_1 be any index in the range $\left(\frac{3}{4}\cdot\frac{n}{2^\ell}, \frac{n}{2^\ell}\right)$. For $1 < i \leq 2^\ell$, denote by $j_i := j_{i-1} + \frac{n}{2^\ell}$. We observe that each j_i is between the end of the i-th active bucket $B_i^{(\ell)}$ and the beginning of the next, so by Corollary 5.5 we have

$$I\left(\ \left(B_1^{(\ell)}, B_2^{(\ell)}, \ldots, B_{2^\ell}^{(\ell)}\right)\ ;\ \left(M_{j_1}, \ldots, M_{j_{2^\ell}}\right)\ \right) = \Omega\left(\frac{2^\ell}{\varepsilon^2}\right).$$

By applying Lemma 3.2 this also implies

$$\sum_{i=1}^{2^\ell} I\left(B_i^{(\ell)}\ ;\ M_{j_i}\ \mid\ M_{j_{(i-1)}}\right) > \Omega\left(2^\ell \cdot \frac{1}{\varepsilon^2}\right),$$

where we abuse notation and treat M_{j_0} as an empty conditioning. Furthermore, each active bucket $B_i^{(\ell)}$ is fully contained within $X_{\left(j_i-\frac{n}{2^\ell},\ j_i-\frac{n}{2^{\ell+1}}\right]}$; Hence,

$$I\left(X_{\left(j_i-\frac{n}{2^\ell},\ j_i-\frac{n}{2^{\ell+1}}\right]}\ ;\ M_{j_i}\ \mid\ M_{j_{i-1}}\right) \geq I\left(B_i^{(\ell)}\ ;\ M_{j_i}\ \mid\ M_{j_{i-1}}\right).$$

The sequences $j_1, j_2, \ldots, j_{2^\ell}$ corresponding to each different $j_1 \in \left(\frac{3}{4}\cdot\frac{n}{2^\ell}, \frac{n}{2^\ell}\right)$ are disjoint, and there are $\Omega\left(\frac{n}{2^\ell}\right)$ of those. We thus conclude that

$$I_\ell \geq \Omega\left(\frac{n}{2^\ell}\right) \cdot \Omega\left(2^\ell \frac{1}{\varepsilon^2}\right) = \Omega\left(n \cdot \frac{1}{\varepsilon^2}\right).$$

□

Combining multiple levels:

Crucially, while the analysis leading to the lower bound of I_ℓ depended on the specific level ℓ, the input stream's distribution is the same across all levels. Our final component is a type of direct-sum over all $\Theta\left(\log\left(\varepsilon^2 n\right)\right)$ information measures I_ℓ.

Lemma 5.8. *For any index $j \in [n]$, we have*

$$I\left(X_1, X_2, \ldots, X_{j-1}\ ;\ M_j\right) \geq \sum_{\ell=1}^{\log\left(\varepsilon\sqrt{n}\right)-2} I\left(X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}\ ;\ M_j\ \mid\ M_{\left(j-\frac{n}{2^\ell}\right)}\right).$$

Proof. The subsets $X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}$ of $X_1, \ldots, X_{j-1}$ are disjoint, and we may thus use the chain law of mutual information to get

$$I\left(X_1, X_2, \ldots, X_{j-1}\ ;\ M_j\right) \geq \sum_{\ell=1}^{\log\left(\varepsilon\sqrt{n}\right)-2} I\left(X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}\ ;\ M_j\ \mid\ X_{\leq\left(j-\frac{n}{2^\ell}\right)}\right).$$

We conclude the proof by observing that

$$I\left(X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}\ ;\ M_j\ \mid\ X_{\leq\left(j-\frac{n}{2^\ell}\right)}\right) = I\left(X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}\ ;\ M_j\ \mid\ M_{\left(j-\frac{n}{2^\ell}\right)}\right),$$

as to both $X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}$ and M_j the conditioning on either $X_{\leq\left(j-\frac{n}{2^\ell}\right)}$ or $M_{\left(j-\frac{n}{2^\ell}\right)}$ is equivalent. □

We next define a global measure of information between the algorithm's memory and the inputs, then bound it using the single-level information measures.

Definition 5.9. *Denote by*

$$\bar{I} := \frac{1}{n}\sum_{j=1}^{n} I\left(X_{<j}\ ;\ M_j\right).$$

Lemma 5.10. $\bar{I} \geq \frac{1}{n}\sum_{\ell=1}^{\log\left(\varepsilon\sqrt{n}\right)-2} I_\ell$.

Proof. Using Lemma 5.8 we have

$$\begin{aligned}
\sum_{j=1}^{n} I\left(X_{<j}\ ;\ M_j\right) &= \sum_{j=1}^{n}\sum_{\ell=1}^{\log\left(\varepsilon\sqrt{n}\right)-2} I\left(X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}\ ;\ M_j\ \mid\ M_{\left(j-\frac{n}{2^\ell}\right)}\right) \\
&\geq \sum_{\ell=1}^{\log\left(\varepsilon\sqrt{n}\right)-2}\sum_{j=1}^{n} I\left(X_{\left(j-\frac{n}{2^\ell},\ j-\frac{n}{2^{\ell+1}}\right]}\ ;\ M_j\ \mid\ M_{\left(j-\frac{n}{2^\ell}\right)}\right) \\
&= \sum_{\ell=1}^{\log\left(\varepsilon\sqrt{n}\right)-2} I_\ell.
\end{aligned}$$

□

Proof of Theorem 5.1. By Lemma 5.10 and Lemma 5.7 we have

$$\bar{I} \geq \sum_{\ell=1}^{\log\left(\varepsilon\sqrt{n}\right)-2} \frac{1}{n} I_\ell \geq \sum_{\ell=1}^{\log\left(\varepsilon\sqrt{n}\right)-2} \Omega\left(\frac{1}{\varepsilon^2}\right) \geq \Omega\left(\frac{\log\left(\varepsilon^2 n\right)}{\varepsilon^2}\right).$$

We conclude the proof by using Observation 3.1. □

5.3 Generalization to $p \in (1, 2]$

The lower bound presented in this section naturally translates to the problem of estimating the fractional p-th norm (that is, $\sum_i f_i^p$ for the frequency vector $\{f_i\}$ or equivalently its L_p norm), for $p \in (1, 2]$.

Theorem 5.11. *Fix $p \in (1, 2]$. Let $\mathcal{A}$ be a streaming algorithm that gives an $(1 \pm \varepsilon)$ approximation to the F_p of its input stream, for some $\varepsilon \in \left(\Omega\left(n^{-1/p}\right), 4^{-1/(p-1)}\right)$, and succeeds with probability $\geq 2/3$. Then, the space used by $\mathcal{A}$ is $\Omega\left(\log\left(\varepsilon^{1/p} n\right)/\varepsilon^2\right)$.*

We only give a short sketch of this generalization, as it only requires setting the parameters of the proof for $p = 2$ differently and the rest of the proof remains as-is.

We consider level indices in the range $\ell \in [1, \log\left(\varepsilon n^{1/p}\right) - 2]$, all construction parameters remain as before besides d which we set to $d = (\varepsilon^p n)/(4t^p)$ instead. In particular, the places of the active buckets in each level remain as before. The appropriate parallel of Lemma 5.2 still holds, as the F_p of the stream is in all cases dominated by $O\left(n + d \cdot k^p\right) = O(n)$ as

$$d \cdot k^p = \left(\frac{\varepsilon^p n}{4t^p}\right) \cdot \left(\frac{t}{\varepsilon}\right)^p \leq n;$$

Furthermore, the gap between the 'Yes' and 'No' cases is at least

$$\begin{aligned} d \cdot ((t+k)^p - t^p - (k+1)^p) &= \left(\frac{\varepsilon^p n}{4t^p}\right) \cdot \left(\left(t + \frac{t}{\varepsilon}\right)^p - t^p - \left(\frac{t}{\varepsilon} + 1\right)^p\right) \\ &= \frac{1}{4} n \cdot \left((1+\varepsilon)^p - \varepsilon^p - \left(1 + \frac{1}{t}\varepsilon\right)^p\right) \\ &\geq \frac{1}{4} n \cdot \left((1+\varepsilon)^p - \left(1 + \frac{1}{2}\varepsilon\right)^p - \varepsilon^p\right) \\ &\geq \frac{1}{4} n \cdot \left(\frac{p}{2}\varepsilon - \varepsilon^p\right) = \Omega(\varepsilon n), \end{aligned}$$

where we used that $(1+x)^p - \left(1 + \frac{x}{2}\right)^p \geq \frac{p}{2}x$ for every $x \geq 0, p > 1$, and also that $\frac{p}{2}\varepsilon - \varepsilon^p \geq \frac{p}{4}\varepsilon$ when $\varepsilon \leq 4^{-1/(p-1)}$.

6 Upper Bound

In this section we give a new tight upper bound on the problem of F_2 estimation. The upper bound is similar to the original AMS construction [AMS96], with the modification that the random vectors onto which we calculate the projections have (random) disjoint support. To simplify the presentation we assume that the algorithm has access to public randomness – similarly to prior works this assumption can be relaxed by replacing true randomness with k-wise independence for a constant k. The same algorithm with a different choice of parameters appeared in [TZ04] (see also [CCFC02]), in the context of improving the update time of AMS. We include a full analysis here for completeness.

Algorithm 1 Partition-based F_2 algorithm

$P \leftarrow \lceil 4/\varepsilon^2 \rceil + 1$;
Let $H : U \to [P]$ and $\gamma : U \to \{-1, 1\}$ be two random hash functions;
Initialize an array $A[1..P]$ of integer counters to 0;
for elements x in the stream **do**
    $A[H(x)] \leftarrow A[H(x)] + \gamma(x)$
Output $A = \sum_{i=1}^{P} A[i]^2$;

6.1 Correctness analysis of Algorithm 1

For the analysis, let A be the random variable representing the algorithm's output, and A_i the random variable representing the i-th cell of the array.

Suppose the stream of length n consists of elements $\{x_j \in U\}_{j=1}^n$ with frequencies $\{f_j \geq 0\}_{j=1}^n$[2].

Let $\gamma_j := \gamma(x_j)$, and let $B_{ji} \in \{0,1\}$ be the indicator random variable specifying whether x_j got mapped to $A[i]$:

$$B_{ji} := \mathbf{1}_{H(x_j)=i}.$$

Then we have

$$A_i = \sum_{j=1}^{n} f_j \cdot B_{ji} \cdot \gamma_j; \quad A = \sum_{i=1}^{P} A_i^2$$

Next, we compute A's expectation and variance over the choices of the H and γ hash functions.

Observe that for each i,

$$\begin{aligned}
\mathsf{E}[A_i^2] &= \mathsf{E}\left[\sum_{j\in[n]} f_j^2 \cdot B_{ji}^2 \cdot \gamma_j^2 + \sum_{j_1 \neq j_2 \in [n]} f_{j_1} f_{j_2} \cdot B_{j_1 i} B_{j_2 i} \cdot \gamma_{j_1}\gamma_{j_2}\right] \\
&= \mathsf{E}\left[\sum_{j\in[n]} f_j^2 \cdot B_{ji}\right] \\
&= \frac{1}{P} \cdot \sum_{j\in[n]} f_j^2
\end{aligned}$$

And for each $i_1 \neq i_2$:

$$\begin{aligned}
\mathsf{E}[A_{i_1} \cdot A_{i_2}] &= \mathsf{E}\left[\sum_{j\in[n]} f_j^2 \cdot B_{ji_1} \cdot B_{ji_2} \cdot \gamma_j^2 + \sum_{j_1 \neq j_2 \in [n]} f_{j_1} f_{j_2} \cdot B_{j_1 i_1} B_{j_2 i_2} \cdot \gamma_{j_1}\gamma_{j_2}\right] \\
&= 0,
\end{aligned}$$

since for all j, $B_{ji_1} \cdot B_{ji_2} = 0$.

Therefore,

$$E[A] = \sum_{i=1}^{P} E[A_i^2] = \sum_{j\in[n]} f_j^2 = F_2,$$

and A is an unbiased estimator of the F_2 moment.

Next, we will calculate the variance $\mathsf{Var}[A]$. Let $i_1 \leq i_2 \leq i_3 \leq i_4 \in [P]$ be any four (not necessarily distinct) elements. We have

$$\mathsf{E}[A_{i_1} \cdot A_{i_2} \cdot A_{i_3} \cdot A_{i_4}] = \sum_{j_1,j_2,j_3,j_4 \in [n]} f_{j_1} f_{j_2} f_{j_3} f_{j_4} \cdot \mathsf{E}[B_{j_1 i_1} B_{j_2 i_2} B_{j_3 i_3} B_{j_4 i_4}] \cdot \mathsf{E}[\gamma_{j_1}\gamma_{j_2}\gamma_{j_3}\gamma_{j_4}]$$

This equals to 0 unless $i_1 = i_2$ and $i_3 = i_4$: for example, if $i_1 < i_2 \leq i_3 \leq i_4$, then when $j_1 \in \{j_2, j_3, j_4\}$ we have $B_{j_1 i_1} B_{j_2 i_2} B_{j_3 i_3} B_{j_4 i_4} = 0$, and when $j_1 \notin \{j_2, j_3, j_4\}$, we have $\mathsf{E}[\gamma_{j_1}\gamma_{j_2}\gamma_{j_3}\gamma_{j_4}] = 0$.

Therefore, we only need to consider terms of the form $\mathsf{E}[A_{i_1}^2 \cdot A_{i_2}^2]$.

[2]We allow some frequencies to be 0 so that the counter on j goes from 1 to n

If $i_1 < i_2$, we have

$$\begin{aligned}
\mathsf{E}[A_{i_1}^2 \cdot A_{i_2}^2] &= \sum_{j_1,j_2,j_3,j_4 \in [n]} f_{j_1} f_{j_2} f_{j_3} f_{j_4} \cdot \mathsf{E}[B_{j_1 i_1} B_{j_2 i_1} B_{j_3 i_2} B_{j_4 i_2}] \cdot \mathsf{E}[\gamma_{j_1} \gamma_{j_2} \gamma_{j_3} \gamma_{j_4}] \\
&= \sum_{j_1 \neq j_3 \in [n]} f_{j_1}^2 f_{j_3}^2 \cdot \mathsf{E}[B_{j_1 i_1} B_{j_3 i_2}] \cdot \mathsf{E}[\gamma_{j_1}^2 \gamma_{j_3}^2] \\
&= \sum_{j_1 \neq j_3 \in [n]} f_{j_1}^2 f_{j_3}^2 / P^2 \\
&= \frac{1}{P^2} \cdot \left(F_2^2 - \sum_j f_j^4 \right) \\
&= \frac{1}{P^2} \cdot \left(F_2^2 - F_4 \right)
\end{aligned}$$

If $i_1 = i_2$, we have

$$\begin{aligned}
\mathsf{E}[A_i^4] &= \sum_{j_1,j_2,j_3,j_4 \in [n]} f_{j_1} f_{j_2} f_{j_3} f_{j_4} \cdot \mathsf{E}[B_{j_1 i} B_{j_2 i} B_{j_3 i} B_{j_4 i}] \cdot \mathsf{E}[\gamma_{j_1} \gamma_{j_2} \gamma_{j_3} \gamma_{j_4}] \\
&= 6 \cdot \sum_{j_5 < j_6 \in [n]} f_{j_5}^2 f_{j_6}^2 \cdot \mathsf{E}[B_{j_5 i} B_{j_6 i}] + \sum_{j \in [n]} f_j^4 \cdot \mathsf{E}[B_{ji}] \\
&= \frac{3}{P^2} \cdot \left(F_2^2 - \sum_j f_j^4 \right) + \frac{1}{P} \cdot \sum_j f_j^4 \\
&= \frac{3}{P^2} \cdot \left(F_2^2 - F_4 \right) + \frac{1}{P} \cdot F_4
\end{aligned}$$

We are now ready to compute $\mathsf{Var}[A]$:

$$\begin{aligned}
\mathsf{Var}[A] &= \mathsf{E}[A^2] - F_2^2 \\
&= \sum_{i_1 \neq i_2 \in [P]} \mathsf{E}[A_{i_1}^2 \cdot A_{i_2}^2] + \sum_{i \in [P]} \mathsf{E}[A_i^4] - F_2^2 \\
&= P \cdot (P-1) \cdot \frac{1}{P^2} \cdot (F_2^2 - F_4) + P \cdot \left(\frac{3}{P^2} \cdot \left(F_2^2 - F_4 \right) + \frac{1}{P} \cdot F_4 \right) - F_2^2 \\
&= \frac{2}{P} \cdot F_2^2 - \frac{2}{P} \cdot F_4 \\
&< \frac{2}{P} \cdot F_2^2
\end{aligned}$$

Theorem 6.1. *The expected relative ℓ_2-error of Algorithm 1 is bounded by ε:*

$$\mathsf{E}\left[\left(\frac{A - F_2}{F_2} \right)^2 \right] < \varepsilon^2 \tag{1}$$

Proof. Since $\mathsf{E}[A] = F_2$, we have

$$\mathsf{E}[(A - F_2)^2] = \mathsf{Var}[A] < \frac{2}{P} \cdot F_2^2 < \varepsilon^2 \cdot F_2^2,$$

implying (1). □

Memory analysis of Algorithm 1 In the word model, or if each element $A[i]$ of the array is stored using $O(\log n)$ bits, the memory cost of the algorithm is $O(P \log n) = O(\varepsilon^{-2} \cdot \log n)$ – matching the memory cost of [AMS96]. We observe that in some regimes – when $\log(\varepsilon^2 n) \ll \log n$ – Algorithm 1 requires asymptotically less memory, matching our lower bounds above.

Claim 6.2. *Suppose the stream length is $n \geq \Omega(\varepsilon^{-2})$. Then Algorithm 1 can be implemented using $O(P \log(n/P)) = O(\varepsilon^{-2} \cdot \log(\varepsilon^2 n))$ memory.*

Proof. Note that the state of the memory consists of P integers satisfying $\sum_i |A[i]| \leq n$. Each integer $A[i]$ can be stored using a prefix-free code that only requires $2\log(|A[i]| + 1) + O(1)$ bits[3].

The total memory cost of concatenating P such prefix-free encodings is therefore at bounded by

$$2 \sum_{i \in [P]} \log(|A[i]| + 1) + O(P) \leq 2P \cdot \log\left(\frac{\sum_{i \in [P]} |A[i]|}{P} + 1\right) + O(P) = O(P \log(n/P)).$$

Here the inequality follows from the concavity of the log function. □

7 Summary and Open Problems

After this work, the space complexity of F_p estimation is understood with nearly-tight bounds for the entire range of $p \geq 0$, as illustrated in Figure 1. Nonetheless, several natural questions remain open.

For second moment estimation, we prove a tight space bound for all $\varepsilon = \Omega(1/\sqrt{n})$, which is $\Theta(n)$ when $\varepsilon = \Theta(1/\sqrt{n})$. By simply maintaining the frequency vector, an exact computation of the second moment is possible with $O\left(\min\{n \log n, |U|\}\right)$ space. A gap between $\Theta(n \log n)$ and $\Theta(n)$ space remains for the range of $0 < \varepsilon < 1/\sqrt{\varepsilon}$. What is the exact space complexity of F_2 estimation in the regime of very low error?

Contrasting our result with the $\tilde{O}\left(\log n + 1/\varepsilon^2\right)$ upper bound in *random-ordered* streams for $p \in (1, 2)$ of [BVWY18], we end up with the first proven gap between the space complexity of F_p estimation in adversarial and random-ordered streams for any range of p. For $p > 2$, it is known that no such gap exists [GH09]. Therefore, only the case of $p = 2$ remains open: Is F_2-estimation "cheaper" in random-ordered streams?

Acknowledgments

The authors would like to deeply thank David Woodruff for several helpful conversations.
MB's research is supported in part by the NSF Alan T. Waterman Award, Grant No. 1933331.
OZ's research is supported in part by the Israel Science Foundation, Grant No. 1593/24.

References

[AGMS99] Noga Alon, Phillip B Gibbons, Yossi Matias, and Mario Szegedy. Tracking join and self-join sizes in limited storage. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 10–20, 1999.

[AKO11] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Streaming algorithms via precision sampling. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 363–372. IEEE, 2011.

[AMS96] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29, 1996.

[3] In fact, $\log |A[i]| + O(\log \log |A[i]|)$ suffices

[And17] Alexandr Andoni. High frequency moments via max-stability. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6364–6368. IEEE, 2017.

[BGKS06] Lakshminath Bhuvanagiri, Sumit Ganguly, Deepanjan Kesh, and Chandan Saha. Simpler algorithm for estimating frequency moments of data streams. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 708–713, 2006.

[BGW20] Mark Braverman, Sumegha Garg, and David P Woodruff. The coin problem with applications to data streams. In *2020 ieee 61st annual symposium on foundations of computer science (focs)*, pages 318–329. IEEE, 2020.

[BO10] Vladimir Braverman and Rafail Ostrovsky. Recursive sketching for frequency moments. *arXiv preprint arXiv:1011.2571*, 2010.

[BVWY18] Vladimir Braverman, Emanuele Viola, David P Woodruff, and Lin F Yang. Revisiting frequency moment estimation in random order streams. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2018.

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

[CCFC02] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer, 2002.

[CKS03a] A. Chakrabarti, S. Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.*, pages 107–117, 2003.

[CKS03b] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.*, pages 107–117. IEEE, 2003.

[FM85] Philippe Flajolet and G Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of computer and system sciences*, 31(2):182–209, 1985.

[Gan11a] Sumit Ganguly. A lower bound for estimating high moments of a data stream. *arXiv preprint arXiv:1201.0253*, 2011.

[Gan11b] Sumit Ganguly. Polynomial estimators for high frequency moments. *arXiv preprint arXiv:1104.4552*, 2011.

[GGI+02] Anna C Gilbert, Sudipto Guha, Piotr Indyk, Yannis Kotidis, Sivaramakrishnan Muthukrishnan, and Martin J Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 389–398, 2002.

[GH09] Sudipto Guha and Zhiyi Huang. Revisiting the direct sum theorem and space lower bounds in random order streams. In *Automata, Languages and Programming: 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I 36*, pages 513–524. Springer, 2009.

[Gro09] 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*, pages 505–516. IBFI Schloss Dagstuhl, 2009.

[IW03] Piotr Indyk and David Woodruff. Tight lower bounds for the distinct elements problem. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 283–288. IEEE, 2003.

[IW05] Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 202–208, 2005.

[Jay09] TS Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 562–573. Springer, 2009.

[JW19] Rajesh Jayaram and David P Woodruff. Towards optimal moment estimation in streaming and distributed models. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2019.

[KNW10a] 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*, pages 1161–1178. SIAM, 2010.

[KNW10b] 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.

[KSZC03] Balachander Krishnamurthy, Subhabrata Sen, Yin Zhang, and Yan Chen. Sketch-based change detection: Methods, evaluation, and applications. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 234–247, 2003.

[LW13] Yi Li and David P Woodruff. A tight lower bound for high frequency moment estimation with small error. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 623–638. Springer, 2013.

[Mor78] Robert Morris. Counting large numbers of events in small registers. *Communications of the ACM*, 21(10):840–842, 1978.

[MW10] Morteza Monemizadeh and David P Woodruff. 1-pass relative-error lp-sampling with applications. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1143–1160. SIAM, 2010.

[NY22] Jelani Nelson and Huacheng Yu. Optimal bounds for approximate counting. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 119–127, 2022.

[TZ04] Mikkel Thorup and Yin Zhang. Tabulation based 4-universal hashing with applications to second moment estimation. In *SODA*, volume 4, pages 615–624, 2004.

[Woo04] David P Woodruff. Optimal space lower bounds for all frequency moments. In *SODA*, volume 4, pages 167–175. Citeseer, 2004.

[WZ12] David P Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 941–960, 2012.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il