# Turnstile Streaming Algorithms Might as Well Be Linear Sketches

Yi Li
Max-Planck Institute for Informatics
yli@mpi-inf.mpg.de

Huy L. Nguyễn
Princeton University
hlnguyen@princeton.edu

David P. Woodruff
IBM Research, Almaden
dpwoodru@us.ibm.com

## Abstract

In the turnstile model of data streams, an underlying vector $x \in \{-m, -m+1, \ldots, m-1, m\}^n$ is presented as a long sequence of arbitrary positive and negative integer updates to its coordinates. A randomized algorithm seeks to approximate a function $f(x)$ with constant probability while only making a single pass over this sequence of updates and using a small amount of space. All known algorithms in this model are linear sketches: they sample a matrix $A$ from a distribution on integer matrices in the preprocessing phase, and maintain the linear sketch $A \cdot x$ while processing the stream. At the end of the stream, they output an arbitrary function of $A \cdot x$. One cannot help but ask: *are linear sketches universal?*

In this work we answer this question by showing that any 1-pass constant probability streaming algorithm for approximating an arbitrary function $f$ of $x$ in the turnstile model can also be implemented by sampling a matrix $A$ from the uniform distribution on $O(n \log m)$ integer matrices, with entries of magnitude $\mathrm{poly}(n)$, and maintaining the linear sketch $A \cdot x$. Furthermore, the logarithm of the number of possible states of $A \cdot x$, as $x$ ranges over $\{-m, -m+1, \ldots, m\}^n$, plus the amount of randomness needed to store $A$, is at most a logarithmic factor larger than the space required of the space-optimal algorithm. Our result shows that to prove space lower bounds for 1-pass streaming algorithms, it suffices to prove lower bounds in the simultaneous model of communication complexity, rather than the stronger 1-way model. Moreover, the fact that we can assume we have a linear sketch with polynomially-bounded entries further simplifies existing lower bounds, e.g., for frequency moments we present a simpler proof of the $\tilde{\Omega}(n^{1-2/k})$ bit complexity lower bound without using communication complexity.

# 1 Introduction

In the turnstile model of data streams [28, 34], there is an underlying $n$-dimensional vector $x$ which is initialized to $\vec{0}$ and evolves through an arbitrary finite sequence of additive updates to its coordinates. These updates are fed into a streaming algorithm, and have the form $x_i \leftarrow x_i + \delta_t$, changing the $i$-th coordinate by the value $\delta_t$ in the $t$-th update, where $\delta_t$ is an arbitrary positive or negative integer. At the end of the stream, $x$ is guaranteed to satisfy the promise that $x \in \{-m, -m+1, \ldots, m\}^n$, where throughout we assume that $m \geq 2n$. The goal of the streaming algorithm is to make one pass over the data and to use limited memory to compute functions of $x$, such as the frequency moments [1], the number of distinct elements [19], the empirical entropy [12], the heavy hitters [14, 17], and treating $x$ as a matrix, various quantities in numerical linear algebra such as a low rank approximation [15]. Since computing these quantities exactly or deterministically often requires a prohibitive amount of space, these algorithms are usually randomized and approximate.

Curiously, all known algorithms in the turnstile model have the following form: they sample an $r \times n$ matrix $A$ from a distribution on integer matrices in a preprocessing phase, and then maintain the "linear sketch" $A \cdot x$ during the stream[1]. For known solutions to the above problems, the sketch $A \cdot x$ is maintained over the integers (rather, than say, a finite field). From $A \cdot x$, the algorithm then computes an arbitrary function of $A \cdot x$, which should with constant probability, be a good approximation to the desired function $f(x)$. Linear sketches are well-suited for turnstile streaming algorithms since given the state $A \cdot x$ and an additive update $x_i \leftarrow x_i + \delta_t$, the new state $A \cdot (x + e_i \cdot \delta_t)$ can be computed as $A \cdot x + \delta_t A_i$, where $e_i$ is the $i$-th standard unit vector and $A_i$ is the $i$-th column of $A$. This raises the natural question of whether or not all turnstile streaming algorithms need to in fact be linear sketches.

Several works already consider the question of lower bounds on the dimension of linear sketches themselves, rather than trying to obtain bit complexity lower bounds [4, 27, 32, 36]. For instance, Andoni et. al [4] prove a lower bound on the dimension of a randomized linear sketch for estimating frequency moments and state "We stress that essentially all known algorithms in the general streaming model are in fact linear estimators." The relationship between the space complexity of the optimal turnstile algorithm and that of a linear sketch is thus important for understanding the applicability of lower bounds for linear sketches.

Some progress has been made on this question. Ganguly [22] shows that for approximating the $\ell_1$-heavy hitters, any deterministic streaming algorithm might as well be a linear sketch over the integers. Ganguly also generalized this reduction to deterministic streaming algorithms for approximating the $\ell_p$-heavy hitters, for every $1 \leq p \leq 2$ [23].

In a related work, Feldman et. al [18] introduced the MUD model for computational tasks, which stands for massive, unordered, distributed algorithms, which contains linear sketches as a special case. The authors show that any deterministic streaming algorithm for computing a symmetric (order-invariant) total single-valued function exactly can be simulated by a MUD algorithm. They partially extend this to approximation algorithms, but require that the approximating function is also a symmetric function of its input, rather than only requiring that the function being approximated is symmetric. For randomized algorithms they require that for each fixed random string, the streaming algorithm computes a symmetric function of its input.

Thus, existing results on this problem were either for deterministic algorithms and specific problems, or did not work for arbitrary approximation algorithms.

**Our Contributions:** We show that any 1-pass constant probability streaming algorithm for approximating an arbitrary function $f$ of $x$ in the turnstile model can be implemented by sampling a matrix $A$ from the uniform distribution with support on $O(n \log m)$ integer matrices, each with entries of magnitude

[1] There are algorithms in the update-only model, where all the $\delta_i$s are required to be positive, which are not linear sketches, e.g., the algorithm for frequency moments given by Alon et. al [1].

$\mathrm{poly}(n \log m)$, and maintaining the linear sketch $A \cdot x$ over the integers. Furthermore, the logarithm of the number of possible states of $A \cdot x$, as $x$ ranges over $\{-m, -m+1, \ldots, m\}^n$, plus the number of random bits needed to sample $A$, is at most an $O(\log m)$ factor larger than the space required of the space-optimal algorithm for approximating $f$ in the turnstile model. Our result helps explain why all known streaming algorithms in the turnstile model are linear sketches.

We note that the logarithmic factor loss in our result is necessary for some problems, e.g., for the function $f(x) = x_1 \bmod 2$. In this case, the optimal algorithm has two states and maintains $x_1 \bmod 2$, while the optimal sketching algorithm $Ax$ over the integers must have at least $\log m$ states since with large constant probability, $A \cdot (1, 0, \ldots, 0) \neq 0$, and by scaling, $A \cdot (i, 0, \ldots, 0)$ results in a different state for each $i \in [m]$.

While the distribution on sketching matrices $A$ that we create can be sampled from with an $O(\log n + \log \log m)$ bit random seed, in general the $O(n \log m)$ matrices in its support cannot be represented in small space. The output function of the sketching algorithm, given $A \cdot x$ and the random bits used to sample $A$, may also not be computable in small space. These issues can be handled by allowing the sketching algorithm to be non-uniform, though there are other ways of addressing them as well. These are discussed further below, and do not affect one of our main applications to lower bounds described next.

*Communication Complexity:* One consequence of our result is for proving lower bounds on the space required of algorithms in the turnstile model, which is often done using communication complexity. Typically one creates a communication problem in which there are two or more players $P_1, \ldots, P_k$, each with an input $X_i$, and lower bounds the communication required among the players to compute a function $f(X_1, \ldots, X_k)$. For 1-pass lower bounds, it suffices to consider 1-way communication, in which $P_1$ speaks to $P_2$ who speaks to $P_3$, etc., and $P_k$ outputs the answer. To obtain lower bounds in the turnstile model, each player $P_i$ creates a data stream $\sigma_i$ from his/her input $X_i$. Then $P_1$ runs a data stream algorithm $A$ on $\sigma_1$, and transmits the memory contents of $A$ to $P_2$, who continues the computation of $A$ on $\sigma_2$, etc. At the end, $A$ will have been executed on the stream $\sigma_1 \circ \sigma_2 \circ \cdots \circ \sigma_k$. If the output of $A$ determines $f$ with constant probability, then the space of the streaming algorithm is at least the randomized communication complexity of $f$ divided by $k$.

While for some problems randomized 1-way communication lower bounds are easy via a reduction from the Indexing problem [31], other problems such as $k$-player Set Disjointness and Gap-$\ell_\infty$ [7, 13, 26, 30] are not much easier to prove than 2-way communication lower bounds. A weaker model than the 1-way model is the simultaneous communication model. In this model there are $k$ players $P_1, \ldots, P_k$, each with an input $X_i$, but the communication is even more restricted. There is an additional player, called the referee, and each of the players $P_i$ can only send a single message, and only to the referee, though they may share a common random string. The referee announces the output, which should equal $f(X_1, \ldots, X_k)$ with constant probability. Before our work, one could not use randomized communication lower bounds for $f$ to obtain space lower bounds for streaming algorithms since there is no simulation in which the state of a streaming algorithm can be passed sequentially among the players. However, since our result shows that the optimal streaming algorithm can be implemented using a sketching matrix $A$, up to a logarithmic factor, each player $P_i$ can compute $A \cdot x_i$, where $x_i$ is the underlying vector associated with the stream $\sigma_i$ that $P_i$ creates from his/her input $X_i$ to the communication game. The referee uses linearity to compute $A \cdot (x_1 + x_2 + \cdots + x_k)$ from which it can compute the output of the streaming algorithm.

Our result thus shows that it suffices to consider simultaneous communication complexity to prove lower bounds. This makes progress on Question 19 on Sketching versus Streaming in the IITK Open Problems in Data Streams from 2006, which asks to show that any symmetric function that admits a good streaming algorithm also admits a sketching algorithm. Our result shows this even for non-symmetric functions (in turnstile streaming model). One well-studied problem for which it is easier to prove lower bounds in the simultaneous model is $k$-player Set Disjointness, see Theorem 18 of [6] which predates the lower bound [7] in the one-way model. This problem is used to prove lower bounds for approximating the $p$-th frequency moment $F_p = \sum_{i=1}^n |x_i|^p$, $p > 2$, in the turnstile model, for which it is known that $\tilde{\Theta}(n^{1-2/p})$ bits of space is necessary and sufficient. This problem has a long history [1, 2, 3, 7, 10, 11, 13, 16, 20, 21, 24, 25, 29, 33, 38]. While we can use the simultaneous lower bound for $k$-player Set Disjointness to prove an $\tilde{\Omega}(n^{1-2/p})$ bound, we give an even simpler proof for linear sketches with polynomially bounded entries without using communication complexity at all in Appendix C. By our reduction, this gives a bit complexity lower bound for turnstile algorithms in general.

*Non-uniformity.* It may not be possible to represent the sketching matrix $A$ or compute the output given $Ax$ in small space, even though $A$ is sampleable with only $O(\log n + \log\log m)$ random bits. A natural way of addressing this is to allow the streaming algorithm to be non-uniform, meaning for each $n$ it has the uniform distribution on our $O(n \log m)$ matrices $A$ hardwired into a read-only tape. We could further hardwire the output of $Az$ for each possible value of $Az$ and each of the $O(n \log m)$ possible $A$. The number of such outputs is $(mn)^{O(\text{rank}(A))}$, which since the algorithm uses $\Theta(\text{rank}(A) \log(mn))$ bits of space to maintain $Ax$, the algorithm does not need additional space to index into this list of outputs. This hardwiring is analogous to the distinction between circuits and Turing machines, and is sometimes the definition used for streaming algorithms, see, e.g., [9].

A second way of addressing this is, since our procedure for finding these $O(n \log m)$ matrices $A$ with associated outputs is constructive, when processing an update $x_i \leftarrow x_i + \delta_t$ the algorithm could run this procedure to reconstruct $A_i$ and add $\delta_t A_i$ to the sketch. This process is consistent across different updates since the algorithm stores the same random seed. Similarly, it could run a constructive procedure to compute its output. Before and after processing each update or output, the state of the streaming algorithm consists only of its random seed together with its sketch $Ax$ for the current $x$. However, the algorithm is allowed more space while processing an update or computing the output.

This non-uniformity does not affect our application to using simultaneous communication complexity to prove streaming lower bounds since the parties can locally create the same $O(n \log m)$ matrices $A$, and use the common random seed to sample an $A$. Moreover, the local space and time complexities are not counted in the communication lower bounds, and so the referee can use additional space to compute the output.

**Our Techniques:** Our starting point is an elegant work of Ganguly [22] which introduces concepts such as path-reversibility and path-independence for modeling a streaming algorithm by a deterministic automaton. One property of deterministic automata is that if the input vector $x$ goes to multiple different states (induced by distinct input streams), then these states can be "merged" and the output of the new state can be chosen to be the output of any of the states being merged. We deal with automata which only have large success probability on an input distribution and so this is no longer true, i.e., if the merging is not done carefully our success probability could drop dramatically. We instead partition the state space into connected components and perform random walks in these components, where the walks correspond to variable-length streams with underlying input $\vec{0}$. Our outputs are defined by samples from the stationary distribution of these walks.

For each fixed random string of the automaton we obtain a deterministic automaton whose state space is isomorphic to the quotient module $\mathbb{Z}^n/M$ for a submodule $M$ of $\mathbb{Z}^n$. In Ganguly's work [22], for his specific problem of approximating the $\ell_1$-heavy hitters he could then remove torsion from $\mathbb{Z}^n/M$ so that this quotient module is free. This is not possible for general problems (e.g., when computing $x_1 \bmod 2$ the quotient module would be $\mathbb{Z}/2\mathbb{Z}$ and removing torsion would make the quotient module 0). We instead crucially rely on the Smith Normal Form of $\mathbb{Z}^n/M$, which allows us to write the states of the automaton as $Bx \bmod q$ for an integer matrix $B$ with $r$ rows, where $q = (q_1, \ldots, q_r)$ is an integer vector. The remainder of our proof has two main ingredients. The first is in showing how to go from $Bx \bmod q$ to a linear sketch $A$ over the integers whose number of states is not much larger than the original number of states. The main difficulty here is that each of these states should be the image of an $x \in \{-m, -m+1, \ldots, m\}^n$, not of an arbitrary $x \in \mathbb{Z}^n$, and we do not know how large the set $\{Bx \bmod q \mid x \in \{-m, -m+1, \ldots, m\}^n\}$ is. The second ingredient is to show that while the entries of $A$ may be exponentially large, we can construct an integer matrix with polynomially bounded entries without increasing the number of states by much. Our procedure reduces the coefficients of random linear combinations of the rows of $A$ modulo random small primes.

## 2 Preliminaries

**Notation.** Let $\mathbb{Z}$ denote the set of integers and $\mathbb{Z}_{|m|} = \{-m, \ldots, m\}$. For a random variable $X$, we write $X \sim \mathcal{D}$ if $X$ is subject to distribution $\mathcal{D}$. We shall denote automata by script letters $\mathcal{A}, \mathcal{B}, \ldots$ and matrices and modules by the regular italic letters $A, B, \ldots$. We also define $a \bmod 0 = a$ for all $a \in \mathbb{Z}$ and allow us to say that 0 is divisible by any integer $a$, denoted as $a|0$, even for $a = 0$.

**Modules and Smith Normal Forms.** Throughout this paper we assume the 'scalars' of a module corresponds to the ring of integers, $\mathbb{Z}$, and we accordingly tailor the definitions.

**Definition 1** (Module). *A $\mathbb{Z}$-module (or module for short) is an additive abelian group $A$ together with a function $\mathbb{Z} \times A \to A$ (the image of $(r, a)$ being denoted by $ra$) such that for all $r, s \in \mathbb{Z}$ and $a, b \in A$ these four properties hold: (1) $r(a + b) = ra + rb$, (2) $(r + s)a = ra + sa$, (3) $r(sa) = (rs)a$, (4) $1a = a$.*

Notions such as submodules, module homomorphisms and finitely generated modules can be defined similarly as for groups and we omit the definitions. Analogously to vector spaces, we can define the notions of linear independence, basis, submodule spanned by a set, etc., and we omit the definitions here. However, since $\mathbb{Z}$ is not a field, a $\mathbb{Z}$-module $A$ may not have a basis. In case it has a basis, we call $A$ a *free $\mathbb{Z}$-module*, or a free module, for short. Note that every submodule of a free $\mathbb{Z}$-module is free. Also, if $F$ is a free module, then any bases of $F$ have the same cardinality. Hence we can define the *dimension* (or *rank*) of $F$ to be the cardinal number of any basis of $F$. As in the case of vector spaces, if we have a linearly independent set of $F$ we can always extend it to a basis of $F$.

The following is an important structure theorem of modules.

**Theorem 1** (Structure Theorem). *Let $A$ be a finitely generated module over $\mathbb{Z}$. Then $A \simeq \mathbb{Z}_{a_1} \oplus \mathbb{Z}_{a_2} \oplus \cdots \oplus \mathbb{Z}_{a_r} \oplus \mathbb{Z}^t$ for some integers $a_1|a_2|\cdots|a_r$. The numbers $r$, $t$ and $a_1, \ldots, a_r$ are uniquely determined by $A$.*

The structure of a module is closely related to the Smith Normal Form. We explore the relation below.

**Definition 2.** *A matrix $A \in \mathbb{Z}^{n\times n}$ is called* unimodular *if* $\det A = 1$ *or* $-1$.

Note that the inverse of a unimodular matrix is an integer matrix and also unimodular.

**Theorem 2** (Smith Normal Form). *Let $A \in \mathbb{Z}^{m\times n}$, then $A$ can be written as $A = SDT$ where $S \in \mathbb{Z}^{m\times m}$ and $T \in \mathbb{Z}^{n\times n}$ are unimodular, and $D \in \mathbb{Z}^{m\times n}$ is a rectangular diagonal matrix with diagonal elements $a_1, \ldots, a_r, 0, \ldots, 0$ for some $r \le \min\{m, n\}$ and nonzero $a_1, \ldots, a_r$ with $a_1|\cdots|a_r$. The matrix $D$ is called the* Smith Normal Form *of $A$ and the diagonal entries $a_1, \ldots, a_r$ are uniquely determined by $A$.*

To connect free modules with Theorem 2, consider the following problem: Suppose that $M$ is a submodule of a free module $F$. We want to find a basis $\{f_1, \ldots, f_n\}$ of $F$ with integers $a_1, \ldots, a_r \in \mathbb{Z}$ with $a_1|\cdots|a_r$ such that $\{a_1 f_1, \ldots, a_r f_r\}$ is a basis of $M$. We say $\{f_1, \ldots, f_n\}$ is a *compatible basis* of $F$ and $M$.

Pick an arbitrary basis $e_1, \ldots, e_n$ of $F$ and suppose that $M$ admits a basis $b_1, \ldots, b_m$ with coefficient matrix $A \in \mathbb{Z}^{m\times n}$ with respect to $e_1, \ldots, e_n$, that is, $b_i = \sum_j A_{ij} e_j$. Reduce $A$ to its Smith Normal Form $A = SDT$, then

$$\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} = S \cdot D \cdot T \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} =: S \cdot D \cdot \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$$

where $D$ is the Smith Normal Form of $A$. It is clear that $f_1, \ldots, f_n$ is a basis of $F$ and easy to check that $\{a_1 f_1, \ldots, a_r f_r\}$ is a basis of $M$.

Finally, in connection with Theorem 1, we note that $F/M \simeq \mathbb{Z}/(a_1) \oplus \cdots \mathbb{Z}/(a_r) \oplus \mathbb{Z}^{n-m}$. This fact will be repeatedly used in Section 4. Our convention that 0 divides 0 allows us to write $a_1 | \cdots | a_{r+n-m}$, where $a_{r+1} = \cdots = a_{r+n-m} = 0$, in a more unified notation, $F/M \simeq \mathbb{Z}/(a_1) \oplus \cdots \oplus \mathbb{Z}/(a_{r+n-m})$.

**Deterministic Stream Automata.** The results in this section are largely due to Ganguly [22].

We consider only the problems in which the input is a vector $v \in \mathbb{Z}^n$ but represented as a data stream $\sigma = (\sigma_1, \sigma_2, \ldots)$ in which each element $\sigma_i$ is an element of $\Sigma = \{e_1, \ldots, e_n, -e_i, \ldots, -e_n\}$ (where the $e_i$'s are canonical basis vectors) such that $\sum_i \sigma_i = v$, and in this case, we write $v = \operatorname{freq} \sigma$.

**Definition 3.** *A deterministic stream automaton $\mathcal{A}$ is a deterministic Turing machine that uses two tapes, a one-way (unidirectional) read-only input tape and a (bidirectional) two way work-tape. The input tape contains the input stream $\sigma$. After processing its input, the automaton writes an output, denoted by $\phi_{\mathcal{A}}(\sigma)$, on the work-tape.*

A configuration of a stream automaton $\mathcal{A}$ is modeled as a triple $(q, h, w)$, where, $q$ is a state of the finite control, $h$ the current head position of the work-tape and $w$ the content of the work-tape. The set of configurations of a stream automaton $\mathcal{A}$ that are reachable from the initial configuration $o$ on some input stream is denoted as $C(\mathcal{A})$. A stream automaton is a tuple $(n, C, o, \oplus, \phi)$, where $n$ specifies the dimension of the underlying vector, $\oplus : C \times \Sigma \to C$ is the configuration transition function, $o$ is the initial position of the automaton and $\phi : C \to \mathbb{Z}^{p(n)}$ is the output function and $p(n)$ is the dimension of the output. For a stream $\sigma$ we also write $\phi(o \oplus \sigma)$ as $\phi(\sigma)$ for simplicity.

The set of configurations of an automaton $\mathcal{A}$ that is reachable from the origin $o$ for some input stream $\sigma$ with $\|\operatorname{freq} \sigma\|_\infty \leq m$ is denoted by $C(\mathcal{A}, m)$. The space of the automaton $A$ with stream parameter $m$ is defined as $S(A, m) = \log |C(\mathcal{A}, m)|$.

**Definition 4.** *Let $\mathcal{A}$ and $\mathcal{B}$ be two stream automata. We say $\mathcal{B}$ is an output restriction of $\mathcal{A}$ if for every stream $\sigma$ there exists a stream $\sigma'$ such that* $\operatorname{freq} \sigma = \operatorname{freq} \sigma'$ *and $\phi_{\mathcal{B}}(\sigma) = \phi_{\mathcal{A}}(\sigma')$.*

A problem over a data stream is characterized by a family of binary relations $P_n \subseteq \mathbb{Z}^{p(n)} \times \mathbb{Z}^n$, $n \geq 1$. We say an automaton $A$ *solves* a problem $P$ (with domain size $n$) if for every input stream $\sigma$ it holds that $(\phi_{\mathcal{A}}(\sigma), \operatorname{freq} \sigma) \in P_n$. It is clear that if $\mathcal{A}$ solves a problem and $\mathcal{B}$ is an output restriction of $\mathcal{A}$, then $\mathcal{B}$ solves the same problem.

Now we introduce more concepts of different kinds of automata. Suppose $\sigma$ and $\tau$ are two streams. Let $\sigma \circ \tau$ be the stream obtained by concatenating $\tau$ to the end of $\sigma$, so $\operatorname{freq}(\sigma \circ \tau) = \operatorname{freq} \sigma + \operatorname{freq} \tau$. The *inverse stream* of $\sigma$ is denoted by $\sigma^{-1}$ and defined inductively as follows: $e_i^{-1} = -e_i$, $-e_i^{-1} = e_i$ and $(\sigma \circ \tau)^{-1} = \tau^{-1} \circ \sigma^{-1}$.

**Definition 5.** *A stream automaton $\mathcal{A}$ is said to be* path independent *if for each configuration $s$ and input stream $\sigma$, $s \oplus \sigma$ is dependent only on* $\operatorname{freq} \sigma$ *and $s$. A stream automaton $\mathcal{A}$ is said to be* path-reversible *if for every stream $\sigma$ and configuration $s$, $s \oplus (\sigma \circ \sigma^{-1}) = s$.*

Suppose that $\mathcal{A}$ is a path independent automaton. We can define a function $+ : \mathbb{Z}^n \times C \to C$ as $x + a = a \oplus \sigma$, where $\operatorname{freq} \sigma = x$. Since $\mathcal{A}$ is a path independent automaton, the function $+$ is well-defined. In [22] it is proved that

**Theorem 3.** *Suppose that $\mathcal{A}$ is a path independent automaton with initial configuration $o$. Let $M = \{x \in \mathbb{Z}^n : x + o = 0 + o\}$, then $M$ is a submodule of $\mathbb{Z}^n$, and the mapping $x + M \mapsto x + o$ is a set isomorphism between $\mathbb{Z}^n/M$ and the set of reachable configurations $\{x + o : x \in \mathbb{Z}^n\}$.*

The submodule $M$ above is called the *kernel* of $\mathcal{A}$. The theorem implies that $\mathcal{A}$ gives the same output for all vectors in $x + M$, and the transition function of $\mathcal{A}$ is the canonical addition on $\mathbb{Z}^n/M$. Conversely, given a module $M \subseteq \mathbb{Z}^n$ one can construct an automaton $\mathcal{A}$ whose states are the cosets of $M$ and the transition function is the canonical addition on $\mathbb{Z}^n/M$. Furthermore, $\mathcal{A}$ has $\mathrm{poly}(n)$ states in its finite control. This construction uses the optimal space.

## 3 Randomized Stream Automata

In this section, we shall extend the notions and results of deterministic stream automata to randomized stream automata. The following definition first appeared in [23] but the author did not define its space complexity.

**Definition 6.** *A randomized stream automaton is a deterministic stream automaton with one additional tape for the random bits. The random bit string $R$ is initialized on the random bit tape before any input record is read; thereafter the random bit string is used in a two way read-only manner. The rest of the execution proceeds as in a deterministic stream automaton.*

A randomized stream automaton $\mathcal{A}$ is said to be path-independent if for each randomness $R$ the deterministic instance $\mathcal{A}_R$ is path-independent (path-reversible). The space complexity of $\mathcal{A}$ with stream width parameter $m$ is defined as

$$S(\mathcal{A}, m) = \max_R \{|R| + S(\mathcal{A}_R, m)\}.$$

Next we show how to reduce a general automaton to a path independent automaton. The reduction of deterministic automata is due to Ganguly [22], where he first reduces a general automaton to a path reversible automaton and thence to a path independent automaton. We take the same approach for randomized automata. The difficulty is in defining the output of the reduced automaton.

**Theorem 4.** *Suppose that a randomized path reversible automaton $\mathcal{A}$ succeeds in solving problem $P$ on any stream with probability at least $1-\delta$. Let $\Pi$ be an arbitrary distribution over streams. There exists a deterministic path independent automaton $\mathcal{B}$ with $S(\mathcal{B}, m) \le S(\mathcal{A}, m) + O(\log n)$ which succeeds with probability at least $1 - 2\delta$ on $\Pi$.*

*Proof.* Let $S$ be the set of all streams of length at most $L$ whose frequency vector is $\vec{0}$. We choose $L$ large enough such that for any two states $o_1$ and $o_2$ of $\mathcal{A}$, if there exists a stream $\sigma$ with $\mathrm{freq}\,\sigma = \vec{0}$ such that $o_1 \oplus \sigma = o_2$, then at least one such $\sigma$ is included in $S$. It suffices to take $L = |C(\mathcal{A}, m)| \cdot m^n$. We also include in $S$ the empty stream. Pick $W$ to be a number large enough such that a random walk on an undirected graph of $|C(\mathcal{A}, m)|$ vertices and $n^L$ edges would mix. An upper bound for $W$ is $2^{O((2n)^L)}$. Construct a distribution $\Pi'$ as follows. For each stream $\tau \in \Pi$, we include new streams $\tau \circ \sigma_1 \circ \cdots \circ \sigma_W$ in $\Pi'$ for all $\sigma_1, \ldots, \sigma_W \in S$.

By Yao's minimax principle, there exists a choice of randomness $R$ and thus a deterministic automaton $\mathcal{A}_R$ such that $\mathcal{A}_R$ succeeds on $\Pi'$ with probability at least $1-\delta$. Let $G$ be the associated multi-graph of the states of $\mathcal{A}_R$ where two vertices $o_1, o_2$ are connected by an arc if there exists $\sigma \in S$ such that $o_1 \oplus \sigma = o_2$. Since $\mathcal{A}_R$ is path reversible, the graph $G$ is undirected. Since $S$ contains the empty stream, a random walk in a connected component $C$ of $G$ converges to a stationary distribution $\pi_C$.

Construct a randomized automaton $\mathcal{B}$ as follows. Define a state of $\mathcal{B}$ for each connected component of $G$. Let $o_1$ and $o_2$ be two states of $\mathcal{A}_R$ in the same connected component in $G$, i.e., there exists a stream $\sigma \in S$ such that $o_1 \oplus \sigma = o_2$. Let $o_3 = o_1 \oplus e_i$ and $o_4 = o_2 \oplus e_i$. Since $\mathcal{A}_R$ is path reversible, $o_3 \oplus -e_i = o_1$, so $o_3 \oplus (-e_i \circ \sigma \circ e_i) = o_4$, which implies that $o_3$ and $o_4$ are in the same connected component. Therefore, the transitions among states of $\mathcal{B}$ are well-defined and it is clear that $\mathcal{B}$ is path-reversible. Furthermore, $\mathcal{B}$ is path independent because there is no stream of frequency $\vec{0}$ changing $\mathcal{B}$ from a state to a different state. The output of $B$ on a state is picked randomly from the outputs of $\mathcal{A}$ according to $\pi_C$.

Fix a $\sigma \in \Pi$. We analyze the probability of $\mathcal{A}_R$ succeeding over all choices of $s_1, \ldots, s_W$. Let $C$ be the connected component of $G$ containing the state of $\mathcal{A}_R$ after processing $\sigma$. Because $W$ is large enough, the distribution of the final state of $\mathcal{A}_R$ is close (within $\delta$ in statistical distance) to $\pi_C$. Thus, the probability that $\mathcal{A}_R$ succeeds is at most $\delta$ more than the expected success probability of $\mathcal{B}$.

By an averaging argument, there exists a deterministic automaton $\mathcal{B}$ achieving success probability at least as high as that of the randomized $\mathcal{B}$, which is at least $1 - 2\delta$. □

**Theorem 5.** *Suppose that a randomized algorithm $\mathcal{A}$ solves $P$ on any stream with probability at least $1-\delta$. Let $\Pi$ be an arbitrary distribution over streams. There exists a deterministic path reversible automaton $B$ with $S(\mathcal{B}, m) \leq S(\mathcal{A}, m) + O(\log n)$ which solves $P$ with probability at least $1 - 3\delta$ on $\Pi$.*

*Proof.* Let $S$ be the set of all streams of length at most $L$ whose frequency vector is $\vec{0}$. We choose $L$ large enough such that for any two states $o_1$ and $o_2$ of $\mathcal{A}$, if there exists a stream $\sigma$ with $\operatorname{freq} \sigma = \vec{0}$ such that $o_1 \oplus \sigma = o_2$, then at least one such $\sigma$ is included in $S$. It suffices to take $L = 2^{S(\mathcal{A},m)} m^n$. We also include in $S$ the empty stream. Pick $W$ to be a number large enough such that a lazy random walk from a fixed vertex on a directed graph of $2^{S(\mathcal{A},m)}$ vertices and $n^L$ edges and a positive probability of staying in every step would get to within statistical distance $\delta$ from a stationary distribution. Note that for the same graph, the stationary distribution is dependent on the fixed starting vertex but this is not a problem for the proof. Construct $\Pi'$ as follows. For each stream $\tau \in \Pi$, we include new streams $s_1 \circ s_2 \circ \cdots \circ s_W \circ \sigma \circ s_{W+1} \circ \cdots \circ s_{2W}$ in $\Pi'$ for all $s_1, \ldots, s_{2W} \in S$.

Let $G$ be the associated multi-graph of the states of $\mathcal{A}_R$ where two vertices $o_1, o_2$ are connected by an arc if there exists $\sigma \in S$ such that $o_1 \oplus \sigma = o_2$. Since $S$ contains the empty stream, every vertex in $G$ has a self-loop.

Construct a randomized automaton $\mathcal{B}$ as follows. Let $G' = (V', E')$ be the directed acyclic graph where each vertex represents a strongly connected component of $G$. Let $rep : V' \to V$ be a map such that $rep(v)$ is a (fixed) arbitrary vertex in the strongly connected component $v$ and $com : V \to V'$ a map from a vertex $v \in G$ to its strongly connected component. We call a strongly connected component of $G$ (correspondingly a vertex in $G'$) *terminal* if there is no edge from it to the rest of the graph. Define a map $\alpha : V' \to V'$ where $\alpha(v)$ is an arbitrary terminal vertex reachable from $v$. Let the states of $\mathcal{B}$ be the set of terminal vertices of $G'$. Define the transition function $\oplus'$ on the states of $\mathcal{B}$ as

$$\alpha(s) \oplus' e_i = \alpha(com(rep(\alpha(s)) \oplus e_i)).$$

It is clear that the transition function $\oplus'$ is well-defined: for any $s, t$ with $\alpha(s) = \alpha(t)$, we have $\alpha(s) \oplus' e_i = \alpha(t) \oplus' e_i$. It is easy to see that $\mathcal{B}$ is path reversible. The proof is postponed to Appendix A.

**Lemma 6.** *Consider a terminal vertex $u \in V'$. We have $u \oplus' e_i \circ -e_i = u$.*

Next we set the initial state of $\mathcal{B}$. Let $u_0$ be the initial state of $\mathcal{A}_R$ and let $\pi$ be the stationary distribution for the random walk starting from $u_0$ in $G$. Note that $G$ is directed so the stationary distribution is dependent on the initial state. Notice that $\pi$ is a mixture of stationary distributions of random walks in terminal components reachable from that initial state. The initial state of $\mathcal{B}$ is picked randomly from $V'$ according to the mixing weight of the terminal components in $\pi$.

Finally, the output of each state of $\mathcal{B}$ is picked randomly from the outputs of the states in the corresponding terminal component in $G$ according to the stationary distribution of a random walk in that component.

We now show that the expected failure probability of $\mathcal{B}$ is no more than $3\delta$. It then follows from an averaging argument that there exists a deterministic $\mathcal{B}$ with failure probability at most $3\delta$.

We shall further need the following two lemmata, whose proofs are postponed to Appendix A.

**Lemma 7.** *Let $s$ be a vertex in a terminal component of $G$. For any stream $\sigma$, there is only one terminal component reachable from $s \oplus \sigma$ via streams of frequency $\vec{0}$.*

**Lemma 8.** *If $\mathcal{A}_R$ starts from a state $u \in G$ in a terminal component $C$, and $\mathcal{B}$ starts from $C$, then after every transition, the state of $\mathcal{B}$ always corresponds to the unique terminal component reachable from the state of $\mathcal{A}$ via streams of frequency $\vec{0}$.*

Fix $\sigma \in \Pi$. We analyze the probability of $\mathcal{A}_R$ succeeding over all choices of $s_1, \ldots, s_{2W}$. Because of the aforementioned mixing time of $G$, the distribution of $u_0 \oplus s_1 \circ \cdots \circ s_W$ is within statistical distance $\delta$ from $\pi$. Thus, the statistical distance between the marginal distribution of $u_0 \oplus s_1 \circ \cdots \circ s_W$ over strongly connected components of $G$ and the distribution of the initial state of $B$ is at most $\delta$.

Consider a fixing of $s_1, \ldots, s_W$ such that $u_1 = u_0 \oplus s_1 \circ \cdots s_W$ belongs to a terminal component $C$ of $G$. We compare the success probability of $\mathcal{A}$ with the success probability of $\mathcal{B}$ when its initial state is $C$. By Lemma 7, there is only one terminal component $D$ reachable from $u_1$ via streams of frequency $\vec{0}$. Again by the mixing time of $G$, the distribution of $u_1 \oplus s_{W+1} \circ \cdots \circ s_{2W}$ is within statistical distance $\delta$ from the stationary distribution of the random walk in $D$, which, by Lemma 8, is the state of $\mathcal{B}$. Therefore, the success probability of $\mathcal{A}$ and $\mathcal{B}$ differ by at most $\delta$.

In summary, over all random choices, the success probability of $\mathcal{A}$ and $\mathcal{B}$ differ by at most $2\delta$ and the desirable conclusion follows. □

Both theorems above conclude with the existence of a deterministic automaton that succeeds with probability $\geq 1 - \delta$ on $\Pi$ for any given distribution $\Pi$ over the inputs. By Yao's minimax theorem, there exist a randomized automaton that succeeds with probability $\geq 1-\delta$ on any input. But, the number of random bits needed by the randomized automaton, i.e., the number of different deterministic path independent automata used by the randomized automaton, could be unbounded. However, following an argument due to Newman [35], it suffices for the randomized automaton to pick uniformly at random one of $O(n \log m)$ deterministic automata. Therefore, the additional space needed for the random bits is only $O(\log n + \log \log m)$. For completeness, we include the argument below.

**Theorem 9.** *Let $\mathcal{A}$ be a randomized automaton solving problem $P$ on $\mathbb{Z}^n_{|m|}$ with failure probability at most $\delta$. There is a randomized automaton $\mathcal{B}$ that only needs to pick uniformly at random one of $O(n)$ deterministic instances of $\mathcal{A}$ and solves $P$ with failure probability at most $2\delta$.*

*Proof.* Let $\mathcal{A}_1, \ldots, \mathcal{A}_{O(n\delta^{-2}\log m)}$ be independent draws of deterministic automata picked by $\mathcal{B}$. Fix an input $x \in \mathbb{Z}^n_{|m|}$. Let $p_{\mathcal{A}}(x)$ be the fraction of the automata among $A_1, \ldots, A_{O(n\delta^{-2}\log m)}$ that solve problem $P$ correctly on $x$ and $p_{\mathcal{B}}(x)$ be the probability that $\mathcal{B}$ solves $P$ on $x$ correctly. By a Chernoff bound, we have that $\Pr\{|p_{\mathcal{A}}(x) - p_{\mathcal{B}}(x)| \geq \delta\} \leq \exp(-O(n \log m)) < (2m+1)^{-2n}$. Taking a union bound over all choices of $x \in \mathbb{Z}^n_{|m|}$, we have $\Pr\{|p_{\mathcal{A}}(x) - p_{\mathcal{B}}(x)| \geq \delta \text{ for all } x\} > 0$. Therefore, there exists a set of $\mathcal{A}_1, \ldots, \mathcal{A}_{O(n\delta^{-2}\log m)}$ such that $|p_{\mathcal{A}}(x) - p_{\mathcal{B}}(x)| \leq \delta$ for all $x \in \mathbb{Z}^n_{|m|}$. The automaton $\mathcal{B}$ simply samples uniformly at random from this set of deterministic algorithms. □

# 4 Reduction to Linear Sketches

**Lemma 10.** *Let $M \subseteq \mathbb{Z}^n$ be a module. Then there exists a submodule $M' \subseteq M$ such that*

1. *For all $x, y \in \mathbb{Z}^n_{|m|}$, it holds that $x + M' = y + M'$ whenever $x + M = y + M$.*
2. *$M'$ admits a basis $b_1, \ldots, b_t$ such that $\|b_i\|_\infty \leq C 2^i m$ for some absolute constant $C > 0$.*

The algorithm to find the basis is standard, so the proof is postponed to Appendix B.

**Lemma 11.** *Suppose that $m \geq 1$ and $M \subseteq \mathbb{Z}^n$ is a module. Then there exists a module $M' \supset M$ such that the following are true.*

1. *For all $x, y \in \mathbb{Z}^n_{|m|}$, it holds that $x + M = y + M$ whenever $x + M' = y + M'$.*
2. *The compatible basis of $M$ and $\mathbb{Z}^n$ is also a compatible basis of $M'$ and $\mathbb{Z}^n$.*
3. *Suppose that $\mathbb{Z}^n/M' \simeq \mathbb{Z}/(q_1) \oplus \mathbb{Z}/(q_2) \oplus \cdots \oplus \mathbb{Z}/(q_r)$ for some $q_1 | q_2 | \cdots | q_r$, where $q_i \neq 1$ for all $i$. It then holds that $\left|\left\{[x + M'] : x \in \mathbb{Z}^n_{|2mn|}\right\}\right| \geq 2^r$.*

**Algorithm 1** Extending the module $M$

```
    // $d_{s+1} = \cdots = d_n = 0$
 1: $d'_i \leftarrow d_i$ for $i = 1, \ldots, \ell$
 2: for $i \leftarrow \ell + 1$ to $n$ do
 3:     $M_i \leftarrow \langle d'_1 f_1, \ldots, d'_{i-1} f_{i-1}, d_i f_i, d'_{i+1} f_{i+1}, \ldots, d_n f_n \rangle$
 4:     if $(k f_i + M_i) \cap \mathbb{Z}^n_{|m|} = \emptyset$ for all $k = 1, \ldots, d_i - 1$ then    ▷ When $d_i = 0$ it means for all $k \neq 0$
 5:         $d'_i \leftarrow 1$
 6:     else
 7:         $d'_i \leftarrow d_i$
 8:     end if
 9: end for
10: return $M' \leftarrow \langle d'_1 f_1, \ldots, d'_n f_n \rangle$
```

*Proof.* Suppose that $(f_1, \ldots, f_n)$ is a compatible basis of $M$ and $\mathbb{Z}^n$, and $\{d_1 f_1, \ldots, d_s f_s\}$ is a basis of $M$, where $d_1 | d_2 | \cdots | d_s$ and $d_1 = \cdots = d_\ell = 1$ for some $\ell \leq s$. Run Algorithm 1 and we claim that the returned $M'$ is as desired.

We first show that Property 1 is maintained throughout the loop. In addition to $M_i$ defined on Line 3, we also define

$$M'_i = \langle d'_1 f_1, \ldots, d'_{i-1} f_{i-1}, f_i, d_{i+1} f_{i+1}, \ldots, d_n f_n \rangle.$$

If $(k f_i + M_i) \cap \mathbb{Z}^n_{|2m|} = \emptyset$ for all $k = 1, \ldots, d_i - 1$, then for any $z \in \mathbb{Z}^n_{|2m|}$ it holds that $z \in M_i$ whenever $z \in M'_i$. Suppose that $z \in M'_i$, we write

$$z = c_1 d'_1 f_1 + \cdots + c_{i-1} d'_{i-1} f_{i-1} + c_i f_i + c_{i+1} d_{i+1} f_{i+1} + \cdots + c_s d_s f_s =: k f_i + m,$$

for some $k \in \{0, \ldots, d_i - 1\}$ and $m \in M_i$. Since $(k f_i + M_i) \cap \mathbb{Z}^n_{|2m|} = \emptyset$ for all $k = 1, \ldots, d_i - 1$, it must hold that $k = 0$ and thus $z \in M_i$. Property 1 can be proved inductively using the above argument as the inductive step. Property 2 is immediate.

Now we prove Property 3. Let $S = (f_1, \ldots, f_n)^{-1}$ and $I = \{i : d'_i = d_i \neq 1\} =: \{i_1, \ldots, i_r\}$, then the map $g(x) = ((Sx)_{i_1} \bmod d_{i_1}, \ldots, (Sx)_{i_r} \bmod d_{i_r}), \quad x \in \mathbb{Z}^n$, induces an isomorphism $\mathbb{Z}^n / M' \simeq \mathbb{Z}/(d_{i_1}) \oplus \cdots \oplus \mathbb{Z}/(d_{i_r})$, where it holds automatically that $d_{i_1} | d_{i_2} | \cdots | d_{i_r}$. To simplify the notation, without loss of generality, we assume that $i_j = j$ for $1 \leq i \leq r$. For each $j \in [r]$, it follows from the algorithm that there exists some $k_j \in \{1, \ldots, d_{i_j}\}$ such that $(k_j f_{i_j} + M'_{i_j}) \cap \mathbb{Z}^n_{|2m|} \neq \emptyset$. Pick an arbitrary $x_j \in (k_j f_{i_j} + M'_{i_j}) \cap \mathbb{Z}^n_{|2m|}$, then $g(x_j) = (0, \ldots, 0, k_j, 0, \ldots, 0)$. We then have $2^r$ distinct vectors in $\mathbb{Z}^n_{|2mn|}$, namely, $\sum_{j=1}^r \sigma_j x_j$ with $\sigma \in \{0,1\}^r$, that correspond to $2^r$ distinct cosets of $M'$. □

**Lemma 12.** *Suppose that $m \geq 2n$. Given a deterministic path-independent automaton $\mathcal{A}$, one can construct a deterministic free automaton $\mathcal{B}$ such that*

1. *$\mathcal{B}$ is an output restriction of $\mathcal{A}$ on $\mathbb{Z}^n_{|m/(2n)|}$;*
2. *$S(\mathcal{B}, m/(2n)) \leq S(\mathcal{A}, m) \log m + O(\log n)$;*
3. *The sketching matrix $\Phi_{\mathcal{B}}$ satisfies $\|\Phi_{\mathcal{B}}\|_{\max} \leq \exp(O(n^2 + n \log m))$.*

*Proof.* Let $M_0$ be the kernel of $\mathcal{A}$ and $\dim M_0 = d$. By Lemma 10 we may assume that $M_0$ has a basis $b_1, \ldots, b_d$ such that $\|b_i\|_\infty = O(2^i m)$. Let $B = (b_1, \ldots, b_d) \in \mathbb{Z}^{n \times d}$ and $RBT = \operatorname{diag}(c_1, \ldots, c_d)$ (where $c_1 | \cdots | c_d$) be the Smith Normal Form of $B$, where $R \in \mathbb{Z}^{n \times n}$ and $T \in \mathbb{Z}^{d \times d}$ are unimodular matrices. Now invoking Lemma 11, we can extend $M_0$ to a module $M$ such that

(a) $\mathbb{Z}^n / M \simeq \mathbb{Z}/(q_1) \oplus \cdots \mathbb{Z}/(q_r) \oplus \mathbb{Z}^t$ for some $q_1 | q_2 | \cdots | q_r$ and $q_1 \geq 2$,

(b) there are at least $2^{r+t}$ cosets intersecting $\mathbb{Z}^n_{|m/(2n)|}$

(c) there exists a submatrix $S \in \mathbb{Z}^{(r+t)\times n}$ such that the isomorphism in (a) is given by

$$x \mapsto ((Sx)_1 \bmod q_1, \dots, (Sx)_r \bmod q_r, (Sx)_{r+1}, \dots, (Sx)_{r+t}).$$

From the proof of Lemma 11, we actually have that $(q_1, \dots, q_r)$ is a subsequence of $(c_1, \dots, c_d)$ and $S$ is formed by $r + t$ rows of $R$. Without loss of generality, we can further assume that for each $i \le r$, the entries of $i$-th row of $S$ are contained in $\{0, 1, \dots, q_i - 1\}$ by taking the remainders modulo $q_i$. Then for each $i \le r$, $|Sx|_i \le q_i m/2$ for all $x \in \mathbb{Z}^n_{|m/(2n)|}$, so given some $0 \le k < q_i$ there are at most $m/2$ different $(Sx)_i$ such that $(Sx)_i \bmod q_i = k$.

We construct a new automaton $\mathcal{B}$ whose states correspond to $X_{\mathcal{B}} = \{Sx : x \in \mathbb{Z}^n_{|m/(2n)|}\}$, it is clear that $\mathcal{B}$ is a free automaton and can be made an output restriction of $\mathcal{A}$. This proves **conclusion 1**.

Let $X_{\mathcal{A}}$ be the set of states of $\mathcal{A}$. For each $(k_1, \dots, k_r, k_{r+1}, \dots, k_t) \in X_{\mathcal{A}}$, where $0 \le k_i < q_i$ for all $1 \le i \le r$, there are at most $(m/2)^r$ different states $Sx$ of $\mathcal{B}$ such that

$$((Sx)_1 \bmod q_1, \dots, (Sx)_r \bmod q_r, (Sx)_{r+1}, \dots, (Sx)_t) = (k_1, \dots, k_r, k_{r+1}, \dots, k_{r+t}).$$

This implies that $|X_{\mathcal{B}}| \le (m/2)^r |X_{\mathcal{A}}|$ and thus

$$\begin{aligned} S(\mathcal{B}, (m/2n)) &\le r \log(m/2) + \log|X_{\mathcal{A}}| + O(\log n) \\ &\le \log|X_{\mathcal{A}}| \log(m/2) + \log|X_{\mathcal{A}}| + O(\log n) \le S(\mathcal{A}, m) \log m + O(\log n), \end{aligned}$$

where we used the fact (b) that $S(\mathcal{A}, m/(2n)) \ge r + t$ for the second inequality. This proves **conclusion 2**.

Next we show **Conclusion 3**. Note that $\|B\|_{\max} = O(2^n m)$. By [37, Chapter 8], we can find $S$ such that $\|S\|_{\max} \le n^{2n+5}(\sqrt{n}\|B\|_{\max})^{4n}\|B\|_{\max} = n^{4n+5}\|B\|_{\max}^{4n+1}$. It follows immediately that $\|\Phi_{\mathcal{B}}\|_{\max} = \|R\|_{\max} \le \|S\|_{\max} = \exp(O(n^2 + n\log m))$. □

**Lemma 13.** *Suppose that $\mathcal{A}$ is a deterministic automaton with sketching matrix $\Phi_{\mathcal{A}}$ such that $\|\Phi_{\mathcal{A}}\|_{\max} \le 2^M$ for some $M \ge \max\{20\log_2(mn), 100\}$. Then there exists a deterministic automaton $\mathcal{B}$ such that*

1. *$\mathcal{B}$ is an output restriction of $\mathcal{A}$ on $\mathbb{Z}^n_{|m|}$;*
2. *$S(\mathcal{B}, m) \le 2(3 + \log_M(mn))S(\mathcal{A}, m) + O(\log n)$;*
3. *$\Phi_{\mathcal{B}}$ has integer entries such that $\|\Phi_{\mathcal{B}}\|_{\max} \le M^3$.*

*Proof.* Suppose that $x \in \{-4m^2n^32^M, \dots, 4m^2n^32^M\} \setminus \{0\}$ and $P = \{p \text{ is a prime} : p \le M^{10}\}$. Choose $p$ uniformly at random from $P$. Since $x$ has most $\log_2(4m^2n^32^M) \le 1.2M$ distinct prime factors and $|P| \ge M^3/(11\ln M)$, we see that $\Pr\{p \text{ divides } x\} \le (13.2\ln M)/M^2$.

Suppose that $\Phi_{\mathcal{A}}$ has $r$ rows and $\operatorname{rank}(\Phi_{\mathcal{A}}) = r$. Let $X = \{\Phi_{\mathcal{A}}x : x \in \mathbb{Z}^n_{|m|}\}$. Let $p_1, \dots, p_t$ be i.i.d. uniform on $P$ and $S$ be a random $t \times r$ matrix of entries i.i.d uniform over $\{0, \dots, 2M-1\}$, where $t = 2\log_M|X|$. Construct a matrix $\Phi_{\mathcal{B}}$ such that $(\Phi_{\mathcal{B}})_{ij} = (S\Phi_{\mathcal{A}})_{ij} \bmod p_i$. To show that $\mathcal{B}$ can be realized as an output restriction of $\mathcal{A}$, we shall show that if for any $x, y \in \mathbb{Z}^n_{|m|}$, if $\Phi_{\mathcal{A}}x \ne \Phi_{\mathcal{A}}y$ then $\Phi_{\mathcal{B}}x \ne \Phi_{\mathcal{B}}y$. Let $z = x - y$, then $\Phi_{\mathcal{A}}z \ne 0$. Then for each $i$, $\Pr_{\rho_i}\{\langle\rho_i, \Phi_{\mathcal{A}}z\rangle = 0\} \le 1/(2M)$. Since $|\langle\rho_i, \Phi_{\mathcal{A}}z\rangle| \le 4m^2n^32^M$, it follows from the argument at the beginning of the proof that

$$\Pr_{p_i}\left\{\langle\rho_i, \Phi_{\mathcal{A}}z\rangle \bmod p_i = 0 \middle| \langle\rho_i, \Phi_{\mathcal{A}}z\rangle \ne 0\right\} \le (13.2\ln M)/M^2 < 1/(2M)$$

Summing the two probabilities above, we see that

$$\Pr_{p_i}\{\langle\rho_i, \Phi_{\mathcal{A}}x\rangle = 0\} \le \Pr_{p_i}\{\langle\rho_i, \Phi_{\mathcal{A}}x\rangle \bmod p_i = 0\} < 1/M.$$

Thus

$$\Pr_{p_1,\dots,p_t,\rho_1,\dots,\rho_t}\{\Phi_{\mathcal{B}}z = 0\} < (1/M)^t = 1/|X|^2.$$

Now, taking a union bound over at most $|X|^2$ pairs of $(x, y)$ such that $\Phi_{\mathcal{A}}x \ne \Phi_{\mathcal{A}}y$, we see that there exists a choice of $p_1, \dots, p_t$ and $\rho_1, \dots, \rho_t$ such that the associated $\Phi_{\mathcal{B}}$ meets our requirement.

Finally,

$$\begin{aligned} S(\mathcal{B},m) &\le \log_2 \left|\{\Phi_{\mathcal{B}} x : x \in \mathbb{Z}^n_{|m|}\}\right| + O(\log n) \le \log_2(\|\Phi_B\|_{\max} \cdot mn)^t + O(\log n) \\ &\le t \log_2(M^3 mn) + O(\log n) \le 2 \log_2 |X| \cdot (3 + \log_M(mn)) + O(\log n) \\ &\le 2(3 + \log_M(mn)) S(\mathcal{A}, m) + O(\log n). \quad \square \end{aligned}$$

Combining Lemma 12 and Lemma 13, we conclude this section with

**Theorem 14.** *Suppose that $m \ge 2n$. Given a deterministic path-independent automaton $\mathcal{A}$, one can construct a deterministic free automaton $\mathcal{B}$ such that*

1. *$\mathcal{B}$ is an output restriction of $\mathcal{A}$ on $\mathbb{Z}^n_{|m/(2n)|}$;*
2. *$S(\mathcal{B}, m/(2n)) \le 8S(\mathcal{A}, m) \log m + O(\log n)$;*
3. *The sketching matrix $\Phi_{\mathcal{B}}$ satisfies $\|\Phi_{\mathcal{B}}\|_{\max} = O((n^2 + n \log m)^3)$.*

**Acknolwedgments:** Yi Li and David Woodruff would like to thank the Simons Institute for the Theory of Computing where part of this work was done.

# References

[1] Noga Alon, Yossi Matias, and Mario Szegedy. The Space Complexity of Approximating the Frequency Moments. *JCSS*, 58(1):137–147, 1999.

[2] Alexandr Andoni. High frequency moment via max stability. Available at http://web.mit.edu/andoni/www/papers/fkStable.pdf.

[3] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Streaming algorithms via precision sampling. In *FOCS*, pages 363–372, 2011.

[4] Alexandr Andoni, Huy L. Nguyên, Yury Polyanskiy, and Yihong Wu. Tight lower bound for linear sketches of moments. In *ICALP (1)*, pages 25–32, 2013.

[5] Ziv Bar-Yossef. *The Complexity of Massive Data Set Computations*. PhD thesis, University of California, Berkeley, 2002.

[6] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. Information theory methods in communication complexity. In *IEEE Conference on Computational Complexity*, pages 93–102, 2002.

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

[8] A. Barvinok. *Integer Points in Polyhedra*. Contemporary mathematics. European Mathematical Society, 2008.

[9] Paul Beame, T. S. Jayram, and Atri Rudra. Lower bounds for randomized read/write stream algorithms. In *STOC*, pages 689–698, 2007.

[10] Lakshminath Bhuvanagiri, Sumit Ganguly, Deepanjan Kesh, and Chandan Saha. Simpler algorithm for estimating frequency moments of data streams. In *SODA*, pages 708–713, 2006.

[11] Vladimir Braverman and Rafail Ostrovsky. Recursive sketching for frequency moments. *CoRR*, abs/1011.2571, 2010.

[12] Amit Chakrabarti, Khanh Do Ba, and S. Muthukrishnan. Estimating Entropy and Entropy Norm on Data Streams. In *STACS*, pages 196–205, 2006.

[13] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *CCC*, pages 107–117, 2003.

[14] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *ICALP*, pages 693–703, 2002.

[15] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *STOC*, pages 205–214, 2009.

[16] Don Coppersmith and Ravi Kumar. An improved data stream algorithm for frequency moments. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 151–156, 2004.

[17] Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.

[18] Jon Feldman, S. Muthukrishnan, Anastasios Sidiropoulos, Clifford Stein, and Zoya Svitkina. On distributing symmetric streaming computations. *ACM Transactions on Algorithms*, 6(4), 2010.

[19] Philippe Flajolet and G. Nigel Martin. Probabilistic counting. In *Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 76–82, 1983.

[20] Sumit Ganguly. Estimating frequency moments of data streams using random linear combinations. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM)*, pages 369–380, 2004.

[21] Sumit Ganguly. A hybrid algorithm for estimating frequency moments of data streams, 2004. Manuscript.

[22] Sumit Ganguly. Lower bounds on frequency estimation of data streams. In *Proceedings of the 3rd international conference on Computer science: theory and applications*, CSR'08, pages 204–215, 2008.

[23] Sumit Ganguly. Distributing frequency-dependent data stream computations. In *Proceedings of the Fifteenth Australasian Symposium on Computing: The Australasian Theory - Volume 94*, CATS '09, pages 163–170, 2009.

[24] Sumit Ganguly. Polynomial estimators for high frequency moments. *CoRR*, abs/1104.4552, 2011.

[25] Sumit Ganguly. A lower bound for estimating high moments of a data stream. *CoRR*, abs/1201.0253, 2012.

[26] Andre Gronemeier. Asymptotically optimal lower bounds on the nih-multi-party information complexity of the and-function and disjointness. In *STACS*, pages 505–516, 2009.

[27] Moritz Hardt and David P. Woodruff. How robust are linear sketches to adaptive inputs? In *STOC*, pages 121–130, 2013.

[28] Piotr Indyk. Sketching, streaming and sublinear-space algorithms, 2007. Graduate course notes available at `http://stellar.mit.edu/S/course/6/fa07/6.895/`.

[29] Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In *STOC*, pages 202–208, 2005.

[30] T. S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *APPROX-RANDOM*, pages 562–573, 2009.

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

[32] Yi Li, Huy L. Nguyen, and David P. Woodruff. On sketching matrix norms and the top singular vector. In *SODA*, 2014.

[33] Morteza Monemizadeh and David P. Woodruff. 1-pass relative-error $l_p$-sampling with applications. In *SODA*, 2010.

[34] S. Muthukrishnan. Data Streams: Algorithms and Applications. *Foundations and Trends in Theoretical Computer Science*, 1(2):117–236, 2005.

[35] Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, pages 67–71, 1991.

[36] Eric Price and David P. Woodruff. Lower bounds for adaptive sparse recovery. In *SODA*, pages 652–663, 2013.

[37] Arne Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Eidgenössische Technische Hochschule Zürich, 2000.

[38] David P. Woodruff. Optimal space lower bounds for all frequency moments. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 167–175, 2004.

# A Omitted Details in the Proof of Theorem 5

*Proof of Lemma 6.* Let $v = u \oplus' e_i = \alpha(com(rep(u) \oplus e_i))$. Let $\sigma$ be a stream with $\text{freq}(\sigma) = \vec{0}$ such that $(rep(u) \oplus (e_i)) \oplus \sigma = rep(v)$. Note that $\text{freq}(e_i \circ \sigma \circ -e_i) = \vec{0}$ and $u$ is terminal, we have $\alpha(com(rep(u) \oplus e_i \circ \sigma \circ -e_i)) = u$. Thus,

$$u \oplus' e_i \circ -e_i = \alpha(com(rep(v) \oplus -e_i)) = \alpha(com(rep(u) \oplus e_i \circ \sigma \circ -e_i)) = u. \qquad \square$$

*Proof of Lemma 7.* Let $u, v$ be two vertices in some (possibly different) terminal components reachable from $s \oplus \sigma$. Assume that $s \oplus \sigma \oplus \sigma_u = u$, $s \oplus \sigma \oplus \sigma_v = v$, where $\text{freq}(\sigma_u) = \text{freq}(\sigma_v) = \vec{0}$. Since $s$ belongs to a terminal component and $\text{freq}(\sigma \circ \sigma_u \circ \sigma^{-1}) = \vec{0}$, there is a stream $\sigma'_u$ of frequency $\vec{0}$ such that $u \oplus \sigma^{-1} \circ \sigma'_u = s$. We have $u \oplus \sigma^{-1} \circ \sigma'_u \circ \sigma \circ \sigma_v = v$ and $\text{freq}(\sigma^{-1} \circ \sigma'_u \circ \sigma \circ \sigma_v) = \vec{0}$. This implies that $u, v$ must belong to the same terminal component since they are both in terminal components. □

*Proof of Lemma 8.* We prove by induction. Let $u'$ be the current state of $\mathcal{A}_R$ and $C'$ the current state of $\mathcal{B}$. By the induction hypothesis, $C' = \alpha(com(u'))$. Consider the transition induced by processing $e_i$. Because $C'$ is the unique terminal component reachable from $u'$, there is a stream $\sigma$ of frequency $\vec{0}$ such that $(u' \oplus e_i) \oplus -e_i \circ \sigma = rep(C')$. By definition, the state of $B$ after the transition is $\alpha(com(rep(C') \oplus e_i)) = \alpha(com((u' \oplus e_i) \oplus -e_i \circ \sigma \circ e_i))$, which is the unique terminal component reachable from $u' \oplus e_i$ via streams of frequency $\vec{0}$. □

**Algorithm 2** Finding an independent set of $M$

```
 1: $X \leftarrow \{0\}$
 2: $t \leftarrow 0$
 3: while $(M \setminus X) \cap \mathbb{Z}^n_{|2m|} \neq \emptyset$ do
 4:     Pick $y \in (M \setminus X) \cap \mathbb{Z}^n_{|2m|}$
 5:     if $t = 0$ then
 6:         $b_1 \leftarrow y$
 7:     else
 8:         $\rho \leftarrow d(y, \mathcal{P}(b_1, \ldots, b_t))$
 9:         $Y \leftarrow \{x \in M \setminus X : d(x, \mathcal{P}(b_1, \ldots, b_t)) \leq \rho\}$
10:         $b_{t+1} \leftarrow \arg\min_{z \in Y} d(z, \mathcal{P}(b_1, \ldots, b_t))$      ▷ pick an arbitrary one if there is a tie
11:     end if
12:     $t \leftarrow t + 1$
13:     $X = \text{span}\{b_1, \ldots, b_t\}$
14: end while
15: return $b_1, \ldots, b_t$
```

# B Proof of Lemma 10

*Proof.* We run Algorithm 2 to find linearly independent vectors $b_1, \ldots, b_t \in M$ for some $t$. Since $Y$ is a finite set, it is always possible to find a $b_{i+1}$ in Line 10. The algorithm will always terminate since $\mathbb{Z}^n_{|2m|}$ is a finite set, and it is easy to see that the returned set of vectors $b_1, \ldots, b_t$ are linearly independent. It is not difficult to verify inductively that $\|b_i\|_\infty \leq C2^i m$ for some absolute constant $C > 0$.

Now let $M' = \langle b_1, \ldots, b_t \rangle$, the module generated by $b_1, \ldots, b_t$. It is clear that $M'$ is a submodule of $M$. Suppose that $x, y \in \mathbb{Z}^n_{|m|}$ and $x + M = y + M$. We want to show that $x + M' = y + M'$. Let $z = x - y$. By the termination condition of the algorithm, there exist $\alpha_1, \ldots, \alpha_t \in \mathbb{R}$ such that $z = \alpha_1 b_1 + \cdots + \alpha_t b_t$. It is a standard argument as in [8, Lemma 10.2] to show that all $\alpha_i$'s are integers and thus $z \in M'$. We include the argument below for completeness. Let

$$z' = [\alpha_1]b_1 + \cdots + [\alpha_t]b_t \in M'.$$

It is clear that $z - z' \in M$. If $z - z' \neq 0$, then there exists a maximum $i$ such that $\alpha_i$ is not an integer, so

$$z - z' = (\alpha_1 - [\alpha_1])b_1 + \cdots + (\alpha_i - [\alpha_i])b_i,$$

where $\alpha_i - [\alpha_i] > 0$. Let $\tilde{b}_i$ be the orthogonal projection of $b_i$ onto $\text{span}\{b_1, \ldots, b_{i-1}\}^\perp$. Then

$$d(z - z', \text{span}\{b_1, \ldots, b_{i-1}\}) = (\alpha_i - [\alpha_i])\|\tilde{b}_i\|_2$$

$$d(b_i, \text{span}\{b_1, \ldots, b_{i-1}\}) = \|\tilde{b}_i\|_2$$

so

$$d(z - z', \text{span}(b_1, \ldots, b_{i-1})) < d(b_i, \text{span}(b_1, \ldots, b_{i-1})).$$

Now, let $x$ be the orthogonal projection of $z - z'$ onto $\text{span}(b_1, \ldots, b_{i-1})$ such that

$$x = \beta_1 b_1 + \cdots + \beta_{i-1} b_{i-1}, \quad \beta_1, \ldots, \beta_{i-1} \in \mathbb{R}$$

and let

$$x' = [\beta_1]b_1 + \cdots + [\beta_{i-1}]b_{i-1},$$

then $x' \in M \cap \operatorname{span}(b_1, \ldots, b_{i-1})$, $x - x' \in \mathcal{P}(b_1, \ldots, b_{i-1})$ and $z - z' - x' \in M \setminus X$. It follows that

$$\begin{aligned} d(z - z' - x', \mathcal{P}(b_1, \ldots, b_{i-1})) &\le d(z - z' - x', x - x') \\ &= d(z - z', x) \\ &= d(z - z', \operatorname{span}\{b_1, \ldots, b_{i-1}\}) \\ &< d(b_i, \operatorname{span}\{b_1, \ldots, b_{i-1}\}) \\ &\le d(b_i, \mathcal{P}(b_1, \ldots, b_{i-1})). \end{aligned}$$

Observe that our choice of $b_i$ actually guarantees that

$$d(b_i, \mathcal{P}(b_1, \ldots, b_{i-1})) \le d(w, \mathcal{P}(b_1, \ldots, b_{i-1})), \forall w \in M \setminus \operatorname{span}\{b_1, \ldots, b_{i-1}\}$$

We meet a contradiction. Therefore, $z = z'$ and $\alpha_i \in \mathbb{Z}$ for all $i$. □

# C Frequency Moments

Provided that streaming algorithms can be implemented by a linear sketch without much loss in space complexity, a lower bound of frequency moment problem follows easily, without resorting to complicated communication complexity.

## C.1 Information Theory and Hellinger Distance

Suppose that $X$ is a discrete random variable on $\Omega$ with distribution $p(x)$. Then the entropy $H(X)$ of the random variable $X$ is defined by $H(X) = -\sum_{x\in\Omega} p(x)\log_2 p(x)$. The joint entropy $H(X,Y)$ of a pair of discrete random variables $(X,Y)$ with joint distribution $p(x,y)$ is defined as $H(X,Y) = -\sum_x \sum_y p(x,y)\log p(x,y)$. We also define the conditional entropy $H(X|Y)$ as $H(X|Y) = \sum_y H(X|Y = y)\Pr\{Y = y\}$, where $H(X|Y = y)$ is the entropy of the conditional distribution of $X$ given the event $\{Y = y\}$. The mutual information $I(X;Y)$ is the relative entropy between the joint distribution and the product distribution $p(x)p(y)$ (where $p(x)$ and $p(y)$ are marginal distributions), i.e., $I(X;Y) = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$. The following are the basic properties regarding entropy and mutual information.

**Proposition 15.** *Let $X, Y, Z$ be discrete random variables defined on $\Omega_X, \Omega_Y, \Omega_Z$ respectively and $f$ a function defined on $\Omega$. Then*

1. $0 \le H(X) \le \log|\Omega_X|$, *the right equality is attained iff $X$ is uniform on $\Omega_X$.*
2. *Conditioning reduces entropy: $H(X|Y) \le H(X)$.*
3. $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \ge 0$ *and the equality is attained iff $X$ and $Y$ are independent;*
4. *Chain rule for mutual information: $I(X,Y;Z) = I(X;Z) + I(X;Y|Z)$;*
5. *If $X, Y$ are jointly independent of $Z$ then $H(X|Y,Z) = H(X|Y)$.*

**Definition 7.** *The Hellinger distance $h(P,Q)$ between probability distributions $P$ and $Q$ on a domain $\Omega$ is defined by*

$$h^2(P,Q) = 1 - \sum_{\omega\in\Omega}\sqrt{P(\omega)Q(\omega)} = \frac{1}{2}\sum_{\omega\in\Omega}(\sqrt{P(\omega)} - \sqrt{Q(\omega)})^2.$$

One can verify that the Hellinger distance is a metric satisfying the triangle inequality, see, e.g., [7]. The following proposition connects the Hellinger distance and the total variation distance.

**Proposition 16.** *(see, e.g., [5]) $h^2(P,Q) \le d_{TV}(P,Q) \le \sqrt{2}h(P,Q)$.*

In connection with mutual information, we have that

**Lemma 17** ([7])**.** *Let $F_{z_1}$ and $F_{z_2}$ be two random variables. Let $Z$ denote a random variable with uniform distribution in $\{z_1, z_2\}$. Suppose $F(z)$ is independent of $Z$ for each $z \in \{z_1, z_2\}$. Then, $I(Z;F(Z)) \ge h^2(F_{z_1}, F_{z_2})$.*

### C.2 Lower bound

Suppose that $x \in \mathbb{R}^n$. We say that a data stream algorithm solves the $(\epsilon, p)$-NORM problem if its output $X$ satisfies $(1-\epsilon)\|x\|_p^p \leq X \leq (1+\epsilon)\|x\|_p^p$ with probability $\geq 1-\delta$.

Let $\mu$ be a distribution $\mathbb{Z}^n$ defined as follows. Choose $x$ uniformly at random from $\{0,1\}^n$ and $i$ uniformly at random from $\{1,\ldots,n\}$. With probability 1/2, let $x_i = (2n)^{1/p}$. Let $\mu$ be the distribution of the resulting $x$.

**Theorem 18.** *For any $p > 2$, any randomized free automaton that solves $(c,p)$-NORM problem ($c < \sqrt{2}$) for $x \sim \mu$ with probability $\geq 19/20$, where $m \leq \operatorname{poly}(n)$, requires $\Omega(n^{1-2/p}/\log^2 n)$ space.*

*Proof.* In the light of Theorem 14, we may increase the space complexity by a $\log m$ factor but assume that the algorithm takes linear sketches and the sketching matrix $\Phi \in \mathbb{Z}^{k\times n}$ satisfies $\|\Phi\|_{\max} = O(\operatorname{poly}(n))$. Suppose that $\Phi = (\phi_1,\ldots,\phi_n)$ and $x \sim \{0,1\}^n$. We shall show that

$$I(\Phi x; x) = \Omega(n^{1-2/p}). \tag{1}$$

Assume this is true for the moment, on the other hand, since entries of $\Phi$ have size $O(n^3)$

$$I(\Phi x; x) \leq H(\Phi x) \leq \log((\operatorname{poly}(n))^k) = O(k\log n),$$

it follows that

$$k = \Omega(n^{1-2/p}/\log n)$$

as desired.

Next we prove (1). Observe that

$$\begin{aligned}
I(\Phi x; x) &= \sum_i I(\Phi x; x_i | x_1,\ldots,x_{i-1}) \\
&= \sum_i H(x_i|x_1,\ldots,x_{i-1}) - H(x_i|\Phi x, x_1,\ldots,x_{i-1}) \\
&= \sum_i H(x_i) - H(x_i|\Phi x, x_1,\ldots,x_{i-1}) \quad \text{(since } x_i\text{'s are independent)} \\
&\geq \sum_i H(x_i) - H(x_i|\Phi x) \quad \text{(conditioning reduces entropy)} \\
&= \sum_i I(\Phi x; x_i) \\
&= \sum_i I(x_i\phi_i + R_i; x_i)
\end{aligned}$$

where $R_i = \sum_{j\neq i} x_j\phi_j$. It suffices to show that $I(x_i\phi_i + R_i; R_i) = \Omega(n^{-2/p})$ for $\Omega(n)$ $i$'s.

By correctness of the sketch, it must hold that

$$h^2((2n)^{1/p}\phi_i + R_i, j_0\phi_i + R_i) = \Omega(1) \quad \text{for } \Omega(n)\ i\text{'s}$$

for either $j_0 = 0$ or 1. Observe that $h^2(j\phi_i + R_i, (j-1)\phi_i + R_i) = h^2(\phi_i + R_i, R_i)$ for all $j \geq 1$, no matter whether $j_0 = 0$ or 1, we have that

$$\Omega(1) = h^2((2n)^{1/p}\phi_i + R_i, j_0\phi_i + R_i) \leq \left(\sum_{j=j_0}^{(2n)^{1/p}} h(j\phi_i + R_i, (j-1)\phi_i + R_i)\right)^2 = O(n^{2/p}h^2(\phi_i + R_i, R_i)),$$

whence it follows immediately that

$$I(x_i\phi_i + R_i; x_i) \geq h^2(\phi + R_i, R_i) = \Omega(1/n^{2/p}),$$

where we invoked Lemma 17 for the inequality, noting that $R_i$ and $x_i$ are independent. □