Electronic Colloquium on Computational Complexity, Report No. 122 (2022)

Efficient Linearization Implies the Multiphase Conjecture

Young Kun Ko*

*Department of Computer Science and Engineering, Penn State

August 29, 2022

Abstract

The main motivation for studying linear data structures and circuits is the intuition that non-linear advice cannot help in computing a linear operator. Jukna and Schnitger [JS11, Juk12] formalized this as a conjecture which states that all circuits computing a linear operator can be "linearized," with only a constant size blow-up. We show that if we assume strengthening of this intuition to data structures (to some field $\mathbb{F}$), then this implies Pătraşcu's Multiphase Conjecture [Pat10] for such $\mathbb{F}$. Furthermore, we show that this conjecture is an intermediate conjecture between NOF-Multiphase Conjecture [Pat10] and the Multiphase Conjecture, formalizing why Pătraşcu's original approach to the Multiphase Conjecture is hard.

Our main technical ingredient is proving unconditional space-time tradeoff for the following static data structure problem for any given field $\mathbb{F}$: Let $M \in \mathbb{F}^{m\times n}$ be fixed. Data structure preprocesses input $X \in \mathbb{F}^n$ using s-cells (dependant on M), each of which can store an arbitrary element in $\mathbb{F}$. When $i \in [m]$ is revealed, the data structure can output $\langle M_i, X\rangle$ using t_q probes. We show that there exists $M \in \{0,1\}^{m\times n}$ such that if the functions used by the data structure are all linear and $s \leq \tilde{o}(m)$ then $t_q \geq \tilde{\Omega}(n)$.

As a corollary, we show that Pătraşcu's Multiphase Conjecture [Pat10] when restricted to dynamic linear data structure holds (with unlimited preprocessing) over any field $\mathbb{F}$. This exhibits an explicit dynamic data structure which requires polynomial update time $t_u \geq \tilde{\Omega}(n)$ or query time $t_q \geq \tilde{\Omega}(n)$. This also improves upon the breakthrough work of Larsen [Lar14] which showed a polynomial lower bound for dynamic data structure under the weaker group model.

ISSN 1433-8092

1 Introduction

We study the following concrete static data structure problem of computing a linear operator over a given field $\mathbb{F}$:

Problem 1.1. *Fix some $M \in \mathbb{F}^{m \times n}$. Let $x \in \mathbb{F}^n$ be given as an input. Preprocess x into $h \in \mathbb{F}^s$, so that for any given $i \in [n]$, one can output $\langle M_i, x \rangle$ using t-probes (adaptively) into x and h.*[1]

This remarkably simple problem, which addresses the complexity of computing a linear operator $x \mapsto Mx$ (or matrix-vector multiplication) (with preprocessing) is in the crossroad of Circuit Complexity [Val77, Val92], Static and Dynamic Data Structure Lower Bound [Pat10, DGW19, KW19], Index Coding [BYBJK11] and Cryptoanalysis [CGK19]. Yet we do not fully understand the optimal trade-off between h and the required probe t.

There are two naive solutions on two extremes: (a) Precompute all answers to $\mathcal{Q} = [m]$ using $|\mathcal{Q}|$ additional space. Any query then can be answered using just a single probe into the data structure ($t = 1$); (b) just store the input $s = n$ with no preprocessing. Then for any query $i \in \mathcal{Q}$, read the whole input x to compute the answer $\langle M_i, x \rangle$ to the given query thus $t \geq n$. As example (a) and (b) illustrate, there is a trade-off between space s and time t. The goal of static[2] data structure lower bound is to prove **unconditional** lower bound in this trade-off.

The main tool for proving such a lower bound is the cell-probe model introduced by Yao [Yao81]. Here, a data structure is simply a table of s-memory cells each with w-words. Query time is only measured for probing memory cells. Computations are free-of-charge. Intuitively, this measures how much *information* one must need to read to answer the query. (or it is an information-theoretic lower bound)

Though the model is unrealistically powerful, this implies that a cell-probe lower bound must be a lower bound in any reasonable model. In fact, this is the only setting where we can prove an **unconditional** data structure lower bound – we do not have any other technique to give an unconditional computational lower bound. Yet mainly due to its unrealistic power, the state-of-the-art *explicit* cell-probe lower bound stands at

$$t \geq \Omega\left(\frac{\log(|\mathcal{Q}|/n)}{\log(s/n)}\right) \tag{1}$$

for any *explicit* data structure problem. In the regime of interest, $|\mathcal{Q}| = \text{poly}(n)$ and $s = O(n)$, this is only logarithmic. Proving any unconditional **super-logarithmic lower bound** for query time for static data structure, where the data structure does not change over time and only answers query, is a major open problem [Pat08] in data structure.

Dynamic Data Structure Lower Bound Unlike in static data structure, where we only consider the trade-off between the storage space s and query time t, in the dynamic data structure, we want the data structure to support a sequence of queries and **updates**. And depending on the sequence of updates, the answers to the queries are allowed to change.

For example, consider a data structure that answers s-t connectivity query over a graph. In a static data structure, the graph stays unchanged, therefore if the query asks for the same s and t in any sequence of operations, it will return the same answer. But in the dynamic data structure, it must support the addition and deletion of the edges in the graph. Therefore depending on which edges are added and removed, the answer to the query must change accordingly as well.

[1] With a single probe, you can read a single element of $\mathbb{F}$ therefore a single coordinate in x or h. Computations are free-of-charge

[2] This is static since input does not change after preprocessing

Since the data structure changes over time, in the dynamic data structure, we focus on the trade-off between update time t_u and query time t_q. Unlike in static data structure, a recent breakthrough shows that we can prove super-logarithmic ($\max\{t_u, t_q\} \geq \tilde{\Omega}(\log^{1.5} n)$) cell probe lower bound [Lar12a, LWY18]. Yet, a major open problem is if we can prove $\max\{t_u, t_q\} \geq n^{\varepsilon}$ for some constant $\varepsilon > 0$. That is, can we prove a **polynomial lower bound**?

Pătraşcu [Pat10, Tho13] introduced the so-called Multiphase Problem as a candidate dynamic problem with a polynomial lower bound. The problem proceeds in three stages.

1. (Pre-processing Phase) Pre-process a matrix $M \in \{0,1\}^{m \times n}$ where $m = \text{poly}(n)$
2. (Update Phase) A vector $X \in \{0,1\}^n$ is revealed and data structure updates its memory using $t_u \cdot n$ time.
3. (Query Phase) A query index $i \in [m]$ is revealed and the data structure must output $\mathsf{DISJ}(M_i, X)$ after t_q time.

Pătraşcu's Multiphase Conjecture then states that there exists some $c > 1$ where if $m = n^c$ and $t_u \cdot n < o(m)$, then it must be the case that $t_q > n^{\varepsilon}$ for some constant $\varepsilon > 0$.

Despite its importance, there has not been much progress in completely resolving the conjecture, even if one wants to compute $\langle M_i, X \rangle$ instead of $\mathsf{DISJ}(M_i, X)$ which is a seemingly harder problem. For general data structure, the only known lower bound is merely constant [GGL15]. To achieve any super-constant lower bound, only lower bounds on a restricted class of data structures are known. Polynomial lower bounds are known for some non-adaptive data structures [CEEP12, BL15] and "semi-adaptive" data structures [KW19, DL20] which is the current state-of-the-art.

Data Structure Lower Bounds for Restrictive Models There have also been works to bypass the strength of the cell-probe model via limiting the possible pre-processing or update function respectively for static and dynamic data structure lower bound. Unlike the cell probe model, these models restrict the class of functions that are computed by the data structure. For example, in the linear model that we consider, a cell is only allowed to store some linear function (or weighted sum) of the updated input. That is there exists some **linear** function $P_j(X)$ for each cell address j where X is the input of size n, or entry of j-th cell can be written as

$$P_j(X) = \alpha_{j,0} + \sum_{i=1}^{n} \alpha_{j,i} \cdot X_i$$

for some $\alpha_{j,0}, \alpha_{j,1}, \ldots, \alpha_{j,n} \in \mathbb{F}$. In a more restrictive **group** model, $\alpha_{j,0} = 0$ and $\alpha_{j,i}$'s are bounded to some precomputed elements from the field, and therefore are not allowed to be an arbitrary element from $\mathbb{F}$.

The main intuition with studying these models is that non-linear bits cannot help with computing a linear operator, as also observed in the context of circuit complexity. This was explicitly stated in [JS11, Juk12] in terms of circuits with arbitrary gates. It is worth noting that circuits with arbitrary gates are equivalent to data structures with non-adaptive probes [Vio18]. Therefore static data structures with adaptive probes are stronger than circuits with arbitrary gates.

Conjecture 1.2 ([JS11] informal)**.** *If there exists a (Boolean) circuit with non-linear gates computing $x \mapsto Mx$ with w wires (and depth d) then there exists a linear (Boolean) circuit that computes $x \mapsto Mx$ with $O(w)$ wires (and depth $O(d)$)*

Under this intuition, can we prove a polynomial lower bound for a more restricted model such as a group model or linear model, which encompasses almost all known upper bounds for computational geometry and spatial databases [Mat94, Aga17]?

For static data structure lower bound, it is known that even under the above restriction, if one insists on finding *an explicit* M, the problem remains hard due to its connection to circuit lower bound [DGW19]. While for a more restrictive group model, a recent result by Golovnev et al. [GPRW20] proves an explicit polynomial lower bound $t \geq n^{\Omega(1)}$. This suggests that proving a lower bound in the linear model is a more daunting task than in the group model.

Restriction to group model led to many fruitful lower bounds for dynamic data structure as well [AE99, Pat07]. A breakthrough work by Larsen [Lar14] shows that we can prove a lower bound of $t_u t_q \geq n^{1-O(1)}$ for group model which gives a polynomial bound on update and query time simultaneously (that is $\max\{t_u, t_q\} \geq n^{1/2-O(1)}$)

Derandomizing Hard Instance The main challenge of proving cell-probe lower bound for static data structure comes from the fact that you can write anything in the pre-processing stage. From a counting perspective, there are simply doubly exponentially many possible choices for pre-processing. Even so, a straightforward counting argument by Miltersen [Mil93] shows that "most" data structure problems where each $q \in \mathcal{Q}$ is a Boolean valued function (each of which requires exponential bits of randomness to describe) must have either $s \geq |\mathcal{Q}|^{0.99}$ or $t \geq n^{0.99}$. Yet we have no explicit instance that achieves $t = \omega(\log n)$ if $s = O(n)$ with $|\mathcal{Q}| = \text{poly}(n)$, and any breakthrough will lead to a circuit lower bound [DGW19].

Now suppose we want to describe a hard instance using a few random bits (say $\text{poly}(n)$), which we call the semi-explicit regime. Then can we beat the bound of (1)? This is an open problem raised in [DGW19].

1.1 Our Result

We show that for any given field $\mathbb{F}$, there exists 0/1 matrices that are hard to compute with a small hint s. Specifically, we show the following **unconditional** trade-off.

- The linear operator $M \in \mathbb{F}^{m \times n}$ is publicly known.
- $X_1, \ldots, X_n \in \mathbb{F}$ is given as input.
- Preprocess M and $X_1, \ldots, X_n$ to hint $H \cdot X = h \in \mathbb{F}^s$ using s extra space using some linear operator $H \in \mathbb{F}^{s \times n}$.
- Given any $i \in [m]$ as query, the querier can output $\langle M_i, X \rangle$ using t adaptive probes into h and $X_1, \ldots, X_n$.

Problem 1: Computing $\langle M_i, X \rangle$ for any $i \in [m]$

We show the following trade-off between s and t for Problem 1. First, we show a random matrix is hard via straightforward counting.

Theorem 1.3 (Informal)**.** *There exists some $M \in \mathbb{F}_q^{m \times n}$ such that if the length of the linear hint $s < o(m)$ then $t \geq \tilde{\Omega}(n)$.*

Since M is random over $\mathbb{F}_q^{m \times n}$, total number of random bits used is $mn \log q$. This dependence on $|\mathbb{F}|$ can be removed at the small cost of s.

Theorem 1.4 (Informal)**.** *There exists some $M \in \{0,1\}^{m \times n}$ such that if the length of the hint $s < \tilde{o}(m)$ then $t \geq \tilde{\Omega}(n)$.*

Now suppose we make the following adjustment of Conjecture 1.2, which extends non-adaptive probe of the circuits to adaptive probes of the data structure.

Conjecture 1.5. *There exists $\mathbb{F}_q$ such that for every collection $\mathcal{M} \subset \mathbb{F}_q^{m \times n}$ of density $1 - o(1)$, there exists $M \in \mathcal{M}$ such that if there exists a static data structure using s space that outputs $\langle M_i, x \rangle$ with t (adaptive) probes, then there exists a linear data structure using $O(s)$ space that outputs $\langle M_i, x \rangle$ using $O(t)$ (adaptive) probes.*

The conjecture states that for every dense $\mathcal{M} \subset \mathbb{F}_q^{m \times n}$ we take, we will be able to find some M which is efficiently linearizable. Note that the conjecture is weaker than stating that all M's are linearizable (as in Conjecture 1.2). Similarly, we can also pose an analogous conjecture over $M \in \{0,1\}^{m \times n}$, which we denote as **Boolean variant** of Conjecture 1.5.

We remark that the only linear operator M that we know of with a non-trivial non-linear data structure is **Vandermonde matrix** [KU08, KU11]. We do not yet know if the data structure for the Vandermonde matrix is linearizable or not.

The theorems (along with the conjecture) have the following consequences in the data structure lower bound.

Static Data Structure Lower Bound If one is willing to believe in this conjecture, then this directly implies a hard semi-explicit static data structure problem. Notice that the only part where randomness is used to describe the problem is the description of M. From Theorem 1.4, we know that this is exactly mn bits. Setting $m = \text{poly}(n)$, and the space used as $m^{0.99}$, we can fit our result in the following table.

Random Bits Used	Query Time Lower Bound
$2^{\Omega(n)}$	$\Omega(n^{0.99})$ [Mil93]
$\text{poly}(n)$	Assuming Conjecture 1.5, $\tilde{\Omega}(n)$ (This work)
0	$\Omega(\log n)$ [Pat08, PTW10, Lar12b] Improvement implies Circuit LB [DGW19]

Table 1: Summary of Known Results for Number of Random Bits used

Dynamic Data Structure Lower Bound A more interesting consequence of Conjecture 1.5 is in dynamic data structure lower bound. Consider the following generalized Multiphase problem over any given field $\mathbb{F}$.

1. (Pre-processing Phase) The linear operator $M \in \mathbb{F}^{m \times n}$ is given and preprocessed.
2. (Update Phase) $X_1, \ldots, X_n \in \mathbb{F}$ is revealed and updated (in time $t_u \cdot n$)
3. (Query Phase) Given any $i \in [m]$ as query, the data structure must output $\langle M_i, X \rangle$ using t_q adaptive probes.

Problem 2: Generalized Multiphase Problem

Here, we remark that this is an explicit dynamic data structure problem, while its static counterpart was a semi-explicit problem. We show that the static data structure lower bound implies the lower bound for Problem 2.

Corollary 1.6. *Let $m = \text{poly}(n)$. For any finite field $\mathbb{F} = \mathbb{F}_q$, any linear data structure for Problem 2 must have* $\max\{t_u, t_q\} \geq \tilde{\Omega}(n)$.

If one does not assume Conjecture 1.5, this is a polynomial lower bound on both update and query time in the **linear model** over an explicit problem, improving over [Lar14] with the group model. We can fit our result in the following table.

	Static	Dynamic
Group Model	$\Omega(n^{\varepsilon})$ [GPRW20]	$\Omega(n^{1/2})$ [Lar14]
Linear Model	$\Omega(\log n)$ [Pat08, PTW10, Lar12b] Improvement implies Circuit LB [DGW19]	$\tilde{\Omega}(n)$ (This work)
General Model	Same as above	$\tilde{\Omega}(\log^2 n)$ [LWY18]

Table 2: Summary of Known Results for Restrictive Models

We also remark that as observed by [DGW19] and [GPRW20], there is a fundamental difference between proving a lower bound in the linear model as opposed to the group model.

Furthermore, if we assume Conjecture 1.5, then this immediately implies that Problem 2 must have $\max\{t_u, t_q\} \geq \tilde{\Omega}(n)$, thereby showing a polynomial lower bound for dynamic data structure.

Remark 1.7. *Here we remark that Brody and Larsen [BL15] also consider the "linear" update model where $\mathbb{F} = \mathbb{F}_2$ and the query must be non-adaptive. We consider any general finite field $\mathbb{F}_q$ and we do not impose any restriction on the query. Furthermore, in our model, linear coefficients of each cell $\alpha_{j,0}, \ldots, \alpha_{j,n}$ are allowed to depend arbitrarily on the pre-processed matrix M, which is not the case for the "linear" model considered by Brody and Larsen.*

Intermediate Conjecture between NOF-Multiphase and Multiphase Another view of our result is formalizing and understanding why Pătraşcu's original approach towards the Multiphase Conjecture remained elusive. Pătraşcu [Pat10] proposed the NOF-Multiphase Conjecture as an avenue of attack on the Multiphase Conjecture. In particular, he proposed the following communication problem.

1. Alice has access to M and $i \in [m]$, Bob has access to X and $i \in [m]$, Charlie has access to M and X.
2. Charlie sends message h to Bob.
3. Alice and Bob communicate t-bits to output $\langle M_i, X \rangle$.

Problem 3: NOF Multiphase Problem

NOF-Multiphase Conjecture states that (say on average over some random M and X) if $h = o(m)$ then $t \geq \Omega(n)$, intuitively saying that since h contains very little information per copy of $i \in [m]$, it cannot help much in computing $\langle M_i, X \rangle$. Pătraşcu then showed that NOF-Multiphase Conjecture implies the Multiphase Conjecture.

One can show that NOF-Multiphase implies Conjecture 1.5 using the reduction given in [KW19]. The NOF communication model in Problem 3 can simulate the static data structure computing M. We attach the proof in Section B. And Conjecture 1.5 implies the Multiphase Conjecture from the connection to the dynamic data structure. This establishes the Conjecture 1.5 as the intermediate conjecture between the NOF-Multiphase and the Multiphase Conjecture.

This gives the following consequence: If we are to attack the Multiphase Conjecture, without having consequences in linear vs non-linear circuits/data structures, we need to fine-tune the communication model, extracting further properties of the underlying dynamic data structure.

1.2 Technical Overview

First, we reduce dynamic data structure lower bound to static data structure lower bound over any given field. Then we reduce static data structure lower bound to some linear combinatorial problem over a given field regardless of the query being adaptive or non-adaptive. Finally, we resolve the combinatorial problem over the desired field.

Reducing to Static Data Structure Lower Bound First, we show that the dynamic data structure lower bound for Problem 2 reduces to showing a static data structure lower bound for Problem 5, which corresponds to Section 2 of the proof.

More specifically, we show that if Multiphase Problem over $\mathbb{F}$ can be computed by a data structure with t_u update time and t_q query time for some collection of matrices $\mathcal{M} \subseteq \mathbb{F}^{m \times n}$, then any $M \in \mathcal{M}$ can be computed by a static data structure using $t_u \cdot n$ space and t_q probes.

t-span Lower Bound Next to derive a contradiction, suppose there exists an efficient data structure that can compute any $\langle M_i, X \rangle$ using s-space and t-probes. (or in abbreviation, there exists a (s,t)-data structure)

From the linearity assumption, we know that all pre-processing functions for the data structure $P_1, \ldots, P_s : \mathbb{F}^n \to \mathbb{F}$ are all some linear functions. Then regardless of the probe being adaptive or not, if P_i's are restricted to be linear, and setting $\mathcal{P} := \text{span}\{\nabla P_1, \ldots, \nabla P_s\}$ where ∇P_i refers to the vector of coefficients of the linear function, M_i can be formed by using at most t terms in $\mathcal{P}$. More formally speaking, for an integer t and set of points $S \subseteq \mathbb{F}^n$, let tS denote t-span of S that is

$$tS = \left\{ \sum_{\ell=1}^{t} w_\ell s_\ell : w_\ell \in \mathbb{F}, s_\ell \in S \right\}.$$

Using this notation, we know that $M_1, \ldots, M_m \in t\mathcal{P}$.

Next, we introduce the following definition.

Definition 1.8 (Sumset evasive)**.** *For integer s and t, a set M is (s,t)-sumset evasive if for any set $S \subseteq \mathbb{F}^n$ of size $|S| = s$, it holds that*

$$M \nsubseteq tS.$$

The question is if there exists a $m \times n$ matrix $M \in \mathbb{F}^{m \times n}$ that is (s,t)-sumset evasive? We answer this combinatorial question affirmatively, first from a naive counting argument. Then we remove the dependence on $|\mathbb{F}|$ by showing that actually there exists $M \in \{0,1\}^{m \times n}$ using a counting argument due to a generalization of classic Theorem of Warren [War68, RBG01] with $s = \tilde{o}(m)$ and $t = \tilde{\Omega}(n)$.

Overall, combined with previous reduction, we know that such M cannot be computed by $(\tilde{o}(m), \tilde{\Omega}(n))$-data structure. Setting $m = n^2$, we obtain that there must exists some $M \in \{0,1\}^{m \times n}$ such that if $t_u < \tilde{o}(n)$ then $t_q > \tilde{\Omega}(n)$.

2 Static Lower Bound implies Dynamic Lower Bound

First, we show that static data structure lower bound for Problem 1 implies a lower bound for dynamic data structure, which might be of independent interest related to the Multiphase Conjecture. Recall that we want to give a lower bound on the trade-off between the update time t_u and the query time t_q for the following **dynamic** data structure problem.

1. (Pre-processing Phase) Pre-process a matrix $M \in \mathbb{F}^{m \times n}$ where $m = \text{poly}(n)$
2. (Update Phase) A vector $X \in \mathbb{F}^n$ is revealed and data structure updates its memory using $t_u \cdot n$ time.
3. (Query Phase) A query index $i \in [m]$ is revealed and the data structure must output $\langle M_i, X \rangle$ after t_q probes.

Problem 4: Multiphase Problem over $\mathbb{F}$

Here, both M and X are part of the input to the data structure, therefore this is an *explicit* dynamic data structure problem. While M is revealed in stage 1, X is revealed in stage 2, the update stage. Therefore the data structure is dynamic in the sense that it must be able to update its contents depending on the second input X.

Now we show that exhibiting a lower bound for Problem 1 is a harder task than for Problem 4. Formally, we show that a lower bound for Problem 5 implies a lower bound for Problem 4.

Lemma 2.1 (Translation between static vs. dynamic)**.** *Suppose there exists data structure for Problem 4 with update time t_u and query time t_q. Then there exists a a static data structure for Problem 1 using $t_u \cdot n$ space and t_q probe for any fixed M.*

Proof. Fix the data structure for Problem 4. Now denote the set of updated cells (and its contents) as U. From our assumption on update time, we know that the number of updated cells $|U| \leq t_u \cdot n$.

Then consider the following data structure for Problem 1. Take U as the pre-processed cells. Then in the query stage, we can simulate the query algorithm from the data structure for Problem 4. For the query to pre-processed cells (i.e. cells processed in stage 1), these are given for free in Problem 1 since M is publicly given. For the query to updated cells, these cells are written as $t_u \cdot n$ extra cells. The number of probes to those cells is at most t_q. □

Now the contrapositive of Lemma 2.1 shows that a lower bound for Problem 1 implies a lower bound for Problem 4. Here also note that the reduction is **not dependent** on the underlying field $\mathbb{F}$. Furthermore, if the data structure for Problem 4 were of the linear model, that is content of each updated cell j is a linear function $P_j(X)$, then the corresponding data structure is of the linear model as well. Therefore formally we get the following lemma for the linear model.

Lemma 2.2 (Linear Translation)**.** *Suppose there exists a linear data structure for Problem 4 with update time t_u and query time t_q. Then there exists a static linear data structure for Problem 1 using $t_u \cdot n$ space and t_q probe for any fixed M.*

3 Static Lower Bound

Recall that we want to prove a lower bound for the following static data structure problem for any field $\mathbb{F}$.

- The linear operator $M \in \mathbb{F}^{m \times n}$ is publicly known.
- $X_1, \ldots, X_n \in \mathbb{F}$ is given as input.
- Preprocess M and $X_1, \ldots, X_n$ to hint $P \cdot X = h \in \mathbb{F}^s$ using s extra space using some linear operator $P \in \mathbb{F}^{s \times n}$.
- Given any $i \in [m]$ as query, the querier can output $\langle M_i, X \rangle$ using t adaptive probes into h and $X_1, \ldots, X_n$.

Problem 5: Computing $\langle M_i, X \rangle$ for any $i \in [m]$

We prove a weaker but straightforward theorem with a dependence on $|\mathbb{F}|$,

Theorem 3.1 (Formal version of Theorem 1.3)**.** *There exists $M \in \mathbb{F}^{m \times n}$ such that any linear data structure for Problem 1 with M must have $t = \Omega(n)$ if $s = O(m)$.*

and the stronger theorem with no dependence.

Theorem 3.2 (Formal version of Theorem 1.4)**.** *There exists $M \in \{0,1\}^{m \times n}$ such that any linear data structure for Problem 1 with M must have $t = \Omega(n/\log^2 n)$ if $s = m/\log n$.*

3.1 Reduction to Outer Dimension

We first show that even if decoding functions are not linear functions, as long as the preprocessing functions $P_1, \ldots, P_s$ are linear, M must be in $t\mathcal{P}$. We remark that an analogous lemma for the circuit was proved in [JS11]. The main difference here is that t-probes are allowed to be adaptive in contrast to the circuit where probes are non-adaptive. Formally, we show the following lemma, the proof of which we append in the appendix.

Lemma 3.3. *Suppose we make t adaptive probes to linear data structure which computes $\langle M_i, X \rangle$ for any given $i \in [m]$. Let the answer to the t adaptive probes be $\langle \alpha_1, X \rangle, \ldots, \langle \alpha_t, X \rangle$. Then it must be the case $M_i \in \mathrm{span}\{\alpha_1, \ldots, \alpha_t\}$*

From the correctness assumption of the data structure, we know that Lemma 3.3 must hold for all i. Therefore, as a corollary we get

Corollary 3.4. *Suppose there exists a static linear data structure for Problem 5 for M with parameters s and t. Then there exists some $\mathcal{P} \subset \mathbb{F}^n$ with $|\mathcal{P}| = s$ such that for all $i \in [m]$, $M_i \in t\mathcal{P}$.*

So if there is a good static data structure for M, M must be a t-span of some s-sized subset.

3.2 Outer Dimension Lower Bounds

In the previous section, we have shown the connection between a good possibly adaptive data structure for M and t-span of $\mathcal{P}$. In other words, if M is computed by a data structure, then M cannot be (s,t)-sumset evasive, that is all the rows in M can be represented as t-sum of some elements in $\mathcal{P}$ of size s. This has a connection with the notion considered in the outer dimension introduced in [PP06].

Definition 3.5 (Outer Dimension of a matrix [PP06])**.** *Let $V \subseteq \mathbb{F}^m$ be a subspace, and t a sparsity parameter. Then the outer dimension of V, denoted as $D_V(t)$ is*

$$D_V(t) := \min_U \{\dim(U) : V \subseteq U, U \text{ is } t\text{-sparse}\}$$

Without loss of generality, we denote the outer dimension of a matrix $M \in \mathbb{F}^{m\times n}$ as $D_M(t)$ for the **column space** of M. Then we have the following lemma from [DGW19].

Lemma 3.6 ([DGW19])**.** *Let $M \subseteq \mathbb{F}^n$ be a subset of size m. Without loss of generality let M be the matrix formed by setting the vectors in M as rows. The following are equivalent.*

- *$D_M(t) \leq s$;*
- *M is not (s,t)-sumset evasive.*

Now we have reduced the problem to finding a matrix M that has a large outer dimension. If we do not care about dependence on $|\mathbb{F}|$, a straightforward counting gives the following lemma.

Lemma 3.7 ([PP06, Lok09])**.** *A random n-dimensional subspace V of $\mathbb{F}^m$ where $|\mathbb{F}| = q$ has*

$$D_V(t) \geq m \cdot \left(1 - \frac{t \log_q m}{n}\right)$$

with high probability. Therefore there exists some matrix $M \in \mathbb{F}_q^{m\times n}$ such that $D_M(t) \geq m \cdot \left(1 - \frac{t \log_q m}{n}\right)$.

Lemma 3.7 combined with Lemma 3.6 and Corollary 3.4 will imply Theorem 3.1 which we will show in Section 3.3. Here, note that the number of random bits required to express a hard M has a dependence on $|\mathbb{F}|$. We can remove this dependency with better counting.

Lemma 3.8. *There exists a matrix $M \in \{0,1\}^{m\times n}$ with $D_M(t) \geq s$ where $t = \Omega(\frac{n}{(\log s)\cdot(\log n)})$ and $s = \frac{m}{\log n}$.*

At a high level, we show that the total number of 0/1-matrix M with $D_M(t) \leq s$ is less than 2^{mn} which is less than the total number of Boolean $m \times n$ matrix. Therefore there must exists M with $D_M(t) \geq s$. A key ingredient here is bound on the number of zero-patterns of a sequence of polynomials whose proof can be found in [RBG01]. Consider the following definition of zero patterns of a sequence of functions.

Definition 3.9 (Zero Pattern)**.** *A set of zero patterns of sequence of functions $(f_1, \ldots, f_m)$ over some field $\mathbb{F}$ with n variables is defined as*

$$Z_{\mathbb{F}}((f_1, \ldots, f_m)) = \{(\delta(f_1(X)), \ldots, \delta(f_m(X))) : X \in \mathbb{F}^n\}$$

where the function δ is defined as

$$\delta(a) = \begin{cases} 0 & \text{if } a = 0 \\ * & \text{otherwise} \end{cases}$$

Now if we view each M_{ij} as a function, and view M as a sequence of M_{ij}, the size of zero patterns of all possible 0/1 matrices M would be exactly 2^{mn}. We will use the following bound on the size of zero patterns.

Theorem 3.10 ([RBG01])**.** *Consider $(f_1, \ldots, f_m)$, a sequence of m polynomials in n variables $x_1, \ldots, x_n$ over $\mathbb{F}$ of degree at most d. Then the number of zero patterns of f is less than*

$$|Z_{\mathbb{F}}((f_1, \ldots, f_m))| \leq \left(\frac{emd}{n}\right)^n.$$

Proof of Lemma 3.8. We count the number of $M \in \{0,1\}^{m\times n}$ with $D_M(t) \le s$ using Theorem 3.10. Now if M has $D_M(t) \le s$, there must exist a t-sparse subspace U which contains M. In other words, there exist vectors $u_1, \ldots, u_s \in \mathbb{F}^m$ which are the basis of the subspace U and when written as a column of a $m \times s$ matrix U,

$$U = \begin{bmatrix} u_1 \mid & \cdots & \mid u_s \end{bmatrix}$$

each row of U is t-sparse.

This induces the following parameterization of any entry M_{ij} as a polynomial over the following set of variables.

Parameterizing U First, we use the sparsity of U to parameterize U. We create $mt \log s$ variables $\zeta_{11}, \ldots, \zeta_{mt} \in [s] = \{0,1\}^{\log s}$ where $\zeta_{k\ell}$ denotes ℓ-th non-zero **index** of k-th row in U. Then create mt variables $\alpha_{11}, \ldots, \alpha_{mt}$ where $\alpha_{k\ell}$ denotes ℓ-th non-zero **entry** of k-th row in U. Then we show that each entry U_{ij} can be written as a degree $\log s + 1$ polynomial over these variables. For any fixed row k and $\ell \in [t]$, we consider the following (degree $\log s$) polynomial

$$q_{k,s}(\zeta_{k\ell}) = \begin{cases} 1 & \text{if } s = \zeta_{k\ell} \\ 0 & \text{otherwise but } \zeta_{k\ell} \in \{0,1\}^{\log s} \end{cases}$$

Then for each entry U_{ij}, given that all $\zeta_{k\ell} \in \{0,1\}^{\log s}$, we can write it as

$$U_{ij} = \sum_{\ell=1}^{t} \alpha_{i\ell} \cdot q_{i,j}(\zeta_{i\ell})$$

which is then a degree $\log s + 1$ polynomial. Therefore, we can first parameterize U by $mt \log s + mt$ variables using degree $\log s + 1$ polynomials, where variables $\zeta_{k\ell}$'s are restricted to be of binary form (binary string representation of $[s]$).

Parameterizing M Next, we express each entry M_{ij} using the parameterization of U and additional ns variables $\gamma_{11}, \ldots, \gamma_{ns} \in \mathbb{F}$. From the definition of the outer dimension of the matrix, $M \subseteq U$. Thus, we know that each column of M, say $m_1, \ldots, m_n$ must be inside U. Therefore using the additional variables, we can write each m_j as

$$m_j = \sum_{t=1}^{s} \gamma_{jt} \cdot u_t$$

Therefore we can write each entry M_{ij} as

$$M_{ij} = p_{ij}(\vec{\alpha}, \vec{\gamma}, \vec{\zeta}) = \sum_{\tau=1}^{s} \gamma_{j\tau} \cdot U_{i\tau} = \sum_{\tau=1}^{s} \sum_{\ell=1}^{t} \gamma_{j\tau} \cdot \alpha_{i\ell} \cdot q_{i,\tau}(\zeta_{i\ell})$$

where p_{ij} is then a degree $\log s + 2$ polynomial over $mt \log s + mt + ns$ variables.

Now since each $M_{ij} \in \{0,1\}$, the zero pattern of p_{ij} uniquely determines the entry of M_{ij}. Note that ζ variables are restricted to be of binary form. But the restriction can only lower the number of possible zero patterns. Thus from Theorem 3.10, the number of possible zero patterns for such p_{ij}'s (therefore the number of matrices M with $D_M(t) \le s$) is at most

$$\left(\frac{emn(t\log s + 2)}{mt\log s + mt + ns}\right)^{mt\log s + mt + ns} \le (en)^{mt\log s + mt + ns} \le 2^{(mt\log s + ns)\cdot \log n}.$$

given that $t \geq 2$.

Setting $t \log s \leq 0.1n/\log n$ and $s \leq 0.1m/\log n$, then we get that the number of zero patterns for p_{ij}'s or the number of matrices M with $D_M(t) \leq s$ is at most $2^{0.5mn}$. But we know that the number of possible Boolean M's is 2^{mn}. Therefore there must exist some M (with high probability) with $D_M(t) \geq s$ under such parameter. □

3.3 Combining Everything Together

Now, we are ready to combine all the lemmas for the proof of the main theorems. First, we combine Lemma 3.7, Lemma 3.6 and Corollary 3.4 to imply Theorem 3.1.

Proof of Theorem 3.1. Suppose we set $s = m/2$ and $t < \frac{n \log q}{2 \log m}$. Under such parameters, from Lemma 3.7 we know that there are M's with $D_M(t) > s$.

But from Lemma 3.6 and Corollary 3.4, we know that any M that can be computed by s-space t-probe linear data structure must have $D_M(t) \leq s$. Thus such M cannot be computed by s-space t-probe linear data structure. □

Similarly, we can combine Lemma 3.8, Lemma 3.6 and Corollary 3.4 to imply Theorem 3.2

Proof of Theorem 3.2. Suppose we set $s = m/\log n$ and $t < o(\frac{n}{\log s \cdot \log n})$. Under such parameters, from Lemma 3.8 we know that there are 0/1 M's with $D_M(t) > s$. But again from Lemma 3.6 and Corollary 3.4, such M cannot be computed by s-space t-probe linear data structure. □

If one believes that non-linear advice cannot help in computing a linear operator (i.e. Conjecture 1.5), these theorems hold for all possible data structures. Number of random bits required for M is equal to $mn = \text{poly}(n)$, with near-optimal (up to logarithmic factors) for s and t, implying derandomization of hard-instances of static data structure.

In addition, from reductions in Section 2, these imply explicit dynamic linear data structure lower bound, which we explicitly state here.

Corollary 3.11. *For any field $\mathbb{F}$, any linear data structure for Problem 2 must have* $\max\{t_u, t_q\} \geq \tilde{\Omega}(n)$. *Furthermore, if we assume Conjecture 1.5, any data structure for Problem 2 must have* $\max\{t_u, t_q\} \geq \tilde{\Omega}(n)$

Proof. Set $m > n^2$. Lemma 2.2 with Theorem 3.2 shows a lower bound of $t_u = s/n = n/\log n$ and $t_q = \Omega(n/\log^2 n)$ for linear data structure.

Suppose we assume Conjecture 1.5, then Lemma 2.1 shows an analogous lower bound for all possible dynamic data structure. □

References

[AE99] Pankaj K. Agarwal and Jeff Erickson. Geometric range searching and its relatives. *Contemp. Math.*, 223:1–56, 1999.

[Aga17] Pankaj K. Agarwal. Simplex Range Searching and Its Variants: A Review. In Martin Loebl, Jaroslav Nešetřil, and Robin Thomas, editors, *A Journey Through Discrete Mathematics*, pages 1–30. Springer International Publishing, Cham, 2017.

[BL15] Joshua Brody and Kasper Gren Larsen. Adapt or Die: Polynomial Lower Bounds for Non-Adaptive Dynamic Data Structures. *THEORY OF COMPUTING*, 11:19, 2015.

[BYBJK11] Ziv Bar-Yossef, Yitzhak Birk, T. S. Jayram, and Tomer Kol. Index Coding With Side Information. *IEEE Transactions on Information Theory*, 57(3):1479–1494, March 2011.

[CEEP12] Arkadev Chattopadhyay, Jeff Edmonds, Faith Ellen, and Toniann Pitassi. A Little Advice Can Be Very Helpful. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 615–625. Society for Industrial and Applied Mathematics, January 2012.

[CGK19] Henry Corrigan-Gibbs and Dmitry Kogan. The Function-Inversion Problem: Barriers and Opportunities. Technical Report 1046, 2019.

[DGW19] Zeev Dvir, Alexander Golovnev, and Omri Weinstein. Static data structure lower bounds imply rigidity. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 967–978, Phoenix, AZ, USA, June 2019. Association for Computing Machinery.

[DL20] Pavel Dvorák and Bruno Loff. Lower Bounds for Semi-adaptive Data Structures via Corruption. *CoRR*, abs/2005.02238, 2020. _eprint: 2005.02238.

[GGL15] R. C. A. Gronlund, A. Grønlund, and K. G. Larsen. New Unconditional Hardness Results for Dynamic and Online Problems. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1089–1107, October 2015.

[GPRW20] Alexander Golovnev, Gleb Posobin, Oded Regev, and Omri Weinstein. Polynomial Data Structure Lower Bounds in the Group Model. Technical Report 057, 2020.

[JS11] Stasys Jukna and Georg Schnitger. Min-rank conjecture for log-depth circuits. *J. Comput. Syst. Sci.*, 77(6):1023–1038, 2011.

[Juk12] Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer Science & Business Media, 2012.

[KU08] Kiran S. Kedlaya and Christopher Umans. Fast Modular Composition in any Characteristic. In *Proc. 49th*, pages 146–155, 2008.

[KU11] Kiran S. Kedlaya and Christopher Umans. Fast Polynomial Factorization and Modular Composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011.

[KW19] Young Kun Ko and Omri Weinstein. An Adaptive Step Toward the Multiphase Conjecture. *FOCS 2020 (to appear)*, October 2019. arXiv: 1910.13543.

[Lar12a] Kasper Green Larsen. The Cell Probe Complexity of Dynamic Range Counting. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 85–94, New York, NY, USA, 2012. ACM. event-place: New York, New York, USA.

[Lar12b] Kasper Green Larsen. Higher Cell Probe Lower Bounds for Evaluating Polynomials. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012*, pages 293–301, 2012.

[Lar14] Kasper Green Larsen. On Range Searching in the Group Model and Combinatorial Discrepancy. *SIAM J. Comput.*, 43(2):673–686, 2014.

[Lok09] Satyanarayana V. Lokam. Complexity Lower Bounds using Linear Algebra. *Found. Trends Theor. Comput. Sci.*, 4(1-2):1–155, 2009.

[LWY18] Kasper Green Larsen, Omri Weinstein, and Huacheng Yu. Crossing the Logarithmic Barrier for Dynamic Boolean Data Structure Lower Bounds. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 978–989, New York, NY, USA, 2018. ACM. event-place: Los Angeles, CA, USA.

[Mat94] Jiří Matoušek. Geometric range searching. *ACM Computing Surveys (CSUR)*, 26(4):422–461, December 1994.

[Mil93] Peter Bro Miltersen. The Bit Probe Complexity Measure Revisited. In *STACS 1993*, pages 662–671, 1993.

[Pat07] Mihai Patrascu. Lower Bounds for 2-dimensional Range Counting. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 40–46, New York, NY, USA, 2007. ACM. event-place: San Diego, California, USA.

[Pat08] Mihai Patrascu. *Lower Bound Techniques for Data Structures*. 2008.

[Pat10] Mihai Patrascu. Towards polynomial lower bounds for dynamic problems. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 603–610. ACM, 2010.

[PP06] Ramamohan Paturi and Pavel Pudlák. Circuit lower bounds and linear codes. *J. Math. Sci.*, 134(5):2425–2434, 2006.

[PTW10] Rina Panigrahy, Kunal Talwar, and Udi Wieder. Lower Bounds on Near Neighbor Search via Metric Expansion. In *FOCS 2010*, pages 805–814, 2010.

[RBG01] Lajos Rónyai, László Babai, and Murali K. Ganapathy. On The Number Of Zero-Patterns Of A Sequence Of Polynomials. *Journal of the American Mathematical Society*, 14(03):717–736, July 2001.

[Tho13] Mikkel Thorup. Mihai PăTraşCu: Obituary and Open Problems. *SIGACT News*, 44(1):110–114, March 2013.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In G. Goos, J. Hartmanis, P. Brinch Hansen, D. Gries, C. Moler, G. Seegmüller, J. Stoer, N. Wirth, and Jozef Gruska, editors, *Mathematical Foundations of Computer Science 1977*, volume 53, pages 162–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 1977.

[Val92] Leslie G. Valiant. Why is Boolean Complexity Theory Difficult? In *Poceedings of the London Mathematical Society Symposium on Boolean Function Complexity*, pages 84–94, New York, NY, USA, 1992. Cambridge University Press. event-place: London, United Kingdom.

[Vio18] Emanuele Viola. Lower bounds for data structures with space close to maximum imply circuit lower bounds. In *ECCC*, volume 25, 2018.

[War68] Hugh E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1):167–167, January 1968.

[Yao81] Andrew Chi-Chih Yao. Should Tables Be Sorted? *J. ACM*, 28(3):615–628, July 1981.

A Missing Proof from Section 3.1

Towards showing Lemma 3.3, first, we show the following simple combinatorial lemma on an affine linear subspace of $\mathbb{F}_q^n$.

Lemma A.1. *Let $Q_1, \ldots, Q_d \in \mathbb{F}_q^n$ be linearly independent vectors. Then consider uniform distribution $\mathcal{U}$ over $X \in \mathbb{F}_q^n$. For any $a_1, \ldots, a_d \in \mathbb{F}_q$,*

$$\Pr_{X \sim \mathcal{U}} \left[\langle Q_d, X \rangle = a_d | \, \langle Q_1, X \rangle = a_1, \ldots, \langle Q_{d-1}, X \rangle = a_{d-1} \right] = \frac{1}{q}.$$

Towards the proof of Lemma A.1, we show the following claim about the property of random inner product with any vector $\alpha \in \mathbb{F}_q^d$ when $q = p^k$ for some prime number p and a natural number k.

Claim A.2. *For any $d \in \mathbb{N}$ and any $\alpha \in \mathbb{F}_q^d$ with $\alpha \neq 0$*

$$\Pr_{X \sim \mathcal{U}} \left[\langle \alpha, X \rangle = \beta \right] = \frac{1}{q}$$

for any $\beta \in \mathbb{F}_q$.

Proof. We prove by induction on d. Suppose $d = 1$ and $\alpha \neq 0$. Since $\mathbb{F}_q$ is a field, the inverse of α, α^{-1} is well-defined and unique. Furthermore for any $\alpha \in \mathbb{F}_q$, $x \mapsto \alpha^{-1} \cdot x$ as a mapping from $\mathbb{F}_q$ to $\mathbb{F}_q$ is a bijection. Therefore we have

$$\Pr_{X \sim \mathcal{U}} \left[\alpha X = \beta \right] = \Pr_{X \sim \mathcal{U}} \left[X = \alpha^{-1} \cdot \beta \right] = \frac{1}{q}.$$

For inductive step, suppose it were true when $d = \ell$. We want to show this is true for $\ell + 1$ as well. We divide into two cases: (i) $\alpha_{\ell+1} = 0$; and (ii) $\alpha_{\ell+1} \neq 0$.

If $\alpha_{\ell+1} = 0$, for any $\beta \in \mathbb{F}_p$ we have that

$$\Pr_{X \sim \mathcal{U}} \left[\langle \alpha, X \rangle = \beta \right] = \Pr_{X \sim \mathcal{U}} \left[\sum_{i=1}^{\ell} \alpha_i X_i = \beta \right] = \frac{1}{q} \tag{2}$$

where the first equality holds since $\alpha_{\ell+1} = 0$ and the second equality holds from induction hypothesis.

Otherwise, if $\alpha_{\ell+1} \neq 0$, we have that for any $\beta', \beta \in \mathbb{F}_q$,

$$\Pr_{X\sim\mathcal{U}}\left[\langle\alpha, X\rangle = \beta \;\middle|\; \sum_{i=1}^{\ell}\alpha_i X_i = \beta'\right] = \Pr_{X\sim\mathcal{U}}\left[\alpha_{\ell+1}\cdot X_{\ell+1} = \beta - \beta' \;\middle|\; \sum_{i=1}^{\ell}\alpha_i X_i = \beta'\right]$$
$$= \Pr_{X\sim\mathcal{U}}\left[\alpha_{\ell+1}\cdot X_{\ell+1} = \beta - \beta'\right] = \Pr_{X\sim\mathcal{U}}\left[X_{\ell+1} = \alpha_{\ell+1}^{-1}(\beta-\beta')\right] = \frac{1}{q}.$$

Then we can write the probability as

$$\Pr_{X\sim\mathcal{U}}\left[\langle\alpha, X\rangle = \beta\right] = \sum_{\beta'}\Pr\left[\langle\alpha, X\rangle = \beta \;\middle|\; \sum_{i=1}^{\ell}\alpha_i X_i = \beta'\right]\cdot\Pr\left[\sum_{i=1}^{\ell}\alpha_i X_i = \beta'\right]$$
$$= \frac{1}{q}\sum_{\beta'}\Pr\left[\sum_{i=1}^{\ell}\alpha_i X_i = \beta'\right] = \frac{1}{q}.$$

□

Now we are ready to prove Lemma A.1 using Claim A.2.

Proof of Lemma A.1.

Let V be the affine subspace with the constraints $\langle Q_1, X\rangle = a_1, \ldots, \langle Q_{d-1}, X\rangle = a_{d-1}$. Since it is an affine subspace, there exists basis $v_0, v_1, \ldots, v_{n-(d-1)} \in V$ such that V can be written as

$$V = \left\{v_0 + \sum \beta_i v_i \;\middle|\; \beta_1, \ldots, \beta_{n-(d-1)} \in \mathbb{F}_q\right\}.$$

Now a uniform distribution over such V is equivalent to a uniform distribution over β with the basis, since different β corresponds to different point. Also by the property of inner product, for any $w \in V$, note that we have

$$\langle Q_d, w\rangle = \left\langle Q_d, \left(v_0 + \sum \beta_i v_i\right)\right\rangle = \langle Q_d, v_0\rangle + \sum \beta_i \langle Q_d, v_i\rangle \tag{3}$$

If we write each $\alpha_i := \langle Q_d, v_i\rangle$, we can rewrite the probability term as

$$\Pr\left[\langle Q_d, X\rangle = a_d \middle| \langle Q_1, X\rangle = a_1, \ldots, \langle Q_{d-1}, X\rangle = a_{d-1}\right]$$
$$= \Pr_{\beta\sim\mathcal{U}}\left[\alpha_0 + \sum \beta_i\alpha_i = a_d\right] = \Pr_{\beta\sim\mathcal{U}}\left[\sum \beta_i\alpha_i = a_d - \alpha_0\right] = \frac{1}{q}$$

where the last equality holds from Claim A.2. □

We are now ready to prove the main lemma of this section using an information-theoretic argument.

Proof of Lemma 3.3. Suppose otherwise. Suppose $M_i \notin \text{span}\{\alpha_1, \ldots, \alpha_t\}$. For concise notation, we introduce the following notation for the answers to the probes. Π_{M_i} denote the final answer, that is $\langle M_i, X\rangle$. Each Π_i denotes answer to the i-th probe, that is $\langle\alpha_i, X\rangle$.

Without loss of generality, assume that $\alpha_1, \ldots, \alpha_t$ are linearly independent. If not, then the data structure could have removed linearly dependent query α_t, and terminate the probes after $t-1$ steps. Now using the property of entropy, we can write the total entropy of the probes as

$$H(\Pi_1, \ldots, \Pi_t) = \sum_{\tau=1}^{t} H(\Pi_\tau | \Pi_{<\tau}) \tag{4}$$

Now consider each $H(\Pi_\tau|\Pi_{<\tau})$. Under any fixed $\Pi_{<\tau}$, we know that the distribution of Π_τ is uniform due to Lemma A.1. Therefore we get that $H(\Pi_\tau|\Pi_{<\tau}) = \log q$. Then we can rewrite (4) as

$$H(\Pi_1, \ldots, \Pi_t) = t \log q. \tag{5}$$

Now since we assumed $\alpha_1, \ldots, \alpha_t$ are linearly independent and $M_i \notin \text{span}\{\alpha_1, \ldots, \alpha_t\}$, $M_i, \alpha_1, \ldots, \alpha_t$ are linearly independent. Therefore Lemma A.1 also implies that $H(\Pi_{M_i}|\Pi_1, \ldots, \Pi_t) = \log q$. We then get from (5)

$$H(\Pi_{M_i}, \Pi_1, \ldots, \Pi_t) = H(\Pi_{M_i}|\Pi_1, \ldots, \Pi_t) + H(\Pi_1, \ldots, \Pi_t) = (t+1) \log q > t \log q \tag{6}$$

But recall that one must know $\langle M_i, X \rangle$ from $\Pi_1, \ldots, \Pi_t$. Therefore the correctness of the protocol implies that $H(\Pi_{M_i}|\Pi_1, \ldots, \Pi_t) = 0$.

$$H(\Pi_{M_i}, \Pi_1, \ldots, \Pi_t) = H(\Pi_1, \ldots, \Pi_t) + H(\Pi_{M_i}|\Pi_1, \ldots, \Pi_t) = t \log q$$

which is a contradiction.

□

B NOF-Multiphase Conjecture implies Conjecture 1.5

In this section, we show that NOF-Multiphase Conjecture implies Conjecture 1.5 using the reduction given in [KW19].

Lemma B.1. *NOF-Multiphase Conjecture implies Conjecture 1.5.*

Proof. Suppose Conjecture 1.5 does not hold. Then we know that there exists some $\mathcal{M} \subset \mathbb{F}^{m \times n}$ of density $1 - o(1)$, that has much better non-linear data structure than linear data structure. We also know that for $1 - o(1)$-fraction of M any linear data structure using $\tilde{o}(m)$ space must have $\tilde{\Omega}(n)$ probes from Theorem 3.1 and Theorem 3.2. Taking the intersection of those two, we get $\mathcal{M}'$ of density $1 - o(1)$ where any $M \in \mathcal{M}'$ has no linear data structure but non-linear data structure using $\tilde{o}(m)$-space and $\tilde{o}(n)$-probes.

But this refutes NOF-Multiphase Conjecture due to the following reduction in [KW19]. Charlie sends h as what is written in data structure of size s to Bob; Alice simulates the query algorithm, Bob answers based on s received from Charlie. Therefore, for any $M \in \mathcal{M}'$, we get an efficient communication protocol. But NOF-Multiphase Conjecture states that this is impossible for such parameters of s and t.

□

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il