Electronic Colloquium on Computational Complexity, Report No. 143 (2019)

Equivalence of Systematic Linear Data Structures and Matrix Rigidity

Sivaramakrishnan Natarajan Ramamoorthy*
University of Washington, Seattle
sivanr@cs.washington.edu

Cyrus Rashtchian
University of California, San Diego
crashtchian@eng.ucsd.edu

October 19, 2019

Abstract

Recently, Dvir, Golovnev, and Weinstein have shown that sufficiently strong lower bounds for linear data structures would imply new bounds for rigid matrices. However, their result utilizes an algorithm that requires an NP oracle, and hence, the rigid matrices are not explicit. In this work, we derive an equivalence between rigidity and the systematic linear model of data structures. For the n-dimensional inner product problem with m queries, we prove that lower bounds on the query time imply rigidity lower bounds for the query set itself. In particular, an explicit lower bound of $\omega\left(\frac{n}{r}\log m\right)$ for r redundant storage bits would yield better rigidity parameters than the best bounds due to Alon, Panigrahy, and Yekhanin. We also prove a converse result, showing that rigid matrices directly correspond to hard query sets for the systematic linear model. As an application, we prove that the set of vectors obtained from rank one binary matrices is rigid with parameters matching the known results for explicit sets. This implies that the vector-matrix-vector problem requires query time $\Omega(n^{3/2}/r)$ for redundancy $r \geqslant \sqrt{n}$ in the systematic linear model, improving a result of Chakraborty, Kamma, and Larsen. Finally, we prove a cell probe lower bound for the vector-matrix-vector problem in the high error regime, improving a result of Chattopadhyay, Koucký, Loff, and Mukhopadhyay.

*Supported by the National Science Foundation under agreement CCF-1420268.

ISSN 1433-8092

1 Introduction

A matrix is *rigid* if it is far in Hamming distance from low rank matrices; it is *explicit* if its entries are computable in polynomial time. A classic result of Valiant proves that explicit rigid matrices imply super-linear lower bounds for linear circuits [35], a major open problem in computational complexity [34, 37]. Implications of new lower bounds for communication complexity and other models are also known [24, 39]. Unfortunately, the current bounds for explicit matrices are very far from the required parameters [16, 33], and natural candidates (e.g., Fourier and Hadamard matrices) have been discovered to be less rigid than desired [3, 12, 14]. This motivates alternative avenues for constructing rigid matrices. Recently, multiple connections between data structures and circuits have arisen [7, 11, 13, 38]. The premise of these results is that hard problems for these models may shed new light on rigid matrices and circuits. We take a similar angle, studying a generic linear problem for a model that resembles a depth-two circuit with linear gates.

Valiant's result concerns arithmetic circuits computing the linear map $v \mapsto Mv$ for a matrix M. In other words, the circuit computes the inner products between v and the rows of M. We study a related data structure problem, the *inner product problem*. The task is to preprocess an n-bit vector v to compute inner products $\langle q, v \rangle$ over $\mathbb{F}_2$ for queries $q \in Q$, where $Q \subseteq \mathbb{F}_2^n$ is the *query set*. This problem generalizes the prefix-sum problem [17] and vector-matrix-vector problem [8, 23].

We consider solving this problem using a restricted data structure model, the *systematic linear model*. This model may only store v verbatim along with a small number $r \ll n$ of *redundant* bits, which are the evaluations of r linear functions of v. To compute $\langle q, v \rangle$ for $q \in Q$, the query algorithm must output a linear function of these r bits along with any t bits of v, where t is the *query time*. We motivate this model with a simple upper bound. Suppose that the query set Q happens to be close to an r-dimensional subspace U. More precisely, assume that $d_H(q, U) \leqslant t$ for any $q \in Q$, where $d_H(q, U) := \min_{u \in U} d_H(q, u)$ and $d_H(q, u)$ denotes the Hamming distance. The systematic linear model will store r bits that correspond to inner products between v and some r vectors that form a basis for U. The query algorithm computes $\langle q, v \rangle$ by invoking the identity $\langle q, v \rangle = \langle u, v \rangle + \langle q - u, v \rangle$, using any vector $u \in U$ with $d_H(q, u) \leqslant t$. Indeed, the r precomputed bits suffice to determine $\langle u, v \rangle$, and at most t bits of v are needed to calculate $\langle q - u, v \rangle$.

We observe that rigidity exactly captures the complexity of the inner product problem in the above model. This connection uses a notion of rigid sets, defined by Alon, Panigrahy and Yekhanin [5]. Our result shows that an efficient algorithm exists in the above model if and only if the query set is not rigid in their sense. Conversely, it is possible to derive new rigidity lower bounds by proving lower bounds for the systematic linear model. A parameter of interest is the size of the rigid set, which corresponds to the number of queries in the inner product problem.

Dvir, Golovnev, and Weinstein also demonstrate a connection between rigidity and a different linear model, which is a restriction of the cell probe model [13]. This model stores $s \geqslant n$ linear functions, and the query algorithm outputs a linear function of t of these s bits. For the inner product problem with query set Q, they show that a lower bound for linear data structures leads to a semi-explicit rigid set. When $|Q| = m$, their result uses a $\mathrm{poly}(m)$ time algorithm that requires access to an NP oracle. Compared to their work, our connection preserves explicitness and offers a

two-way equivalence via the systematic linear model. In particular, when $r = \Theta(n)$, a lower bound of $t = \omega(\log m)$ in the systematic linear model implies that Q is rigid with better parameters than known results. Their work requires a lower bound of $t = \omega(\log m \log n)$ against the linear model, and the resulting set is not explicit. Our results also extend to show that linear data structure lower bounds lead to explicit rigid matrices. However, compared to the work of Dvir, Golovnev, and Weinstein, we require stronger lower bounds to achieve new rigidity parameters.

As an application of our framework, we provide new results for the vector-matrix-vector problem. The task is to preprocess a 0-1 matrix M to compute $u^\intercal M v$ when given vectors u, v as the query. The boolean semiring version of this problem has received much recent attention due to connections to the online matrix-vector multiplication conjecture [18]. Moreover, this problem has motivated the study of data structures for a super polynomial number of queries, even when the output is binary [8, 9]. Other prior work has either studied binary output problems with $\mathsf{poly}(n)$ queries (see e.g. [28, 30]) or achieved better lower bounds by looking at multi-output problems (see e.g. [10, 20]). In general, the vector-matrix-vector problem is a good testbed for proving better data structure lower bounds, because linear algebraic tools could provide new insights.

The $\mathbb{F}_2$ variant of this problem specializes the inner product problem because $u^\intercal M v$ equals the inner product of $uv^\intercal$ and M (viewed as vectors). The query set consists of $\sqrt{n} \times \sqrt{n}$ matrices with rank one; its size m satisfies $\log m = \Theta(\sqrt{n})$. As another contribution, we lower bound the rigidity of this set, and consequently, we obtain a query time lower bound of $\Omega(\frac{n}{r} \log m) = \Omega(n^{3/2}/r)$ for the systematic linear model with redundancy $r \geqslant \sqrt{n}$. Any asymptotically better lower bounds for this problem (in the systematic linear model) would directly imply that this query set is rigid with better parameters than the currently known results for explicit matrices [4, 5].

As a final result, we prove a new cell probe lower bound for the vector-matrix-vector problem, without restrictions on the data structure. Our result improves the current best lower bound due to Chattopadhyay, Koucký, Loff, and Mukhopadhyay [9]. Our lower bound matches the limit of present techniques and achieves the current best time-space trade-off in terms of query set size.

1.1 Rigid sets, systematic linear model, and the inner product partial function

Throughout, let $m = m(n)$ and $t = t(n)$ and $r = r(n)$ denote positive integers, with $m \geqslant n \geqslant t, r$. Alon, Panigrahy and Yekhanin defined the following notion of a rigid set [5].

Definition (Rigid Set)**.** *A set $Q \subseteq \mathbb{F}_2^n$ is (r, t)-rigid if for every subspace $U \subseteq \mathbb{F}_2^n$ with dimension at most r, some vector $q \in Q$ has Hamming distance at least t from all vectors in U, that is, $d_H(q, U) \geqslant t$.*

We define (r', t')-rigid for non-integral r', t' to mean $(\lfloor r' \rfloor, \lceil t' \rceil)$-rigid. It will be convenient to equate a set Q with a matrix M_Q by arranging vectors in Q as rows in M_Q in any order. If Q is (r, t)-rigid and $|Q| = m$, then the corresponding matrix $M_Q \in \mathbb{F}_2^{m \times n}$ is rigid in the usual sense: for any rank r matrix A, some row in $(M_Q - A)$ contains at least t nonzero entries. Hence, we may refer to rigid sets and rigid rectangular matrices interchangeably. A matrix in $\mathbb{F}_2^{m \times n}$ (or a set of n-dimensional vectors) is *explicit* if every entry can be computed in $\mathsf{poly}(n)$ time.

A random $m \times n$ matrix with $m = \mathrm{poly}(n)$ will be $(\epsilon n, \delta n / \log n)$-rigid with high probability for some constants $\epsilon, \delta \in (0, 1)$. The key challenge here is to construct explicit rigid matrices,

because they provide circuit lower bounds for functions that can be described in polynomial time [35]. Alon, Panigrahy and Yekhanin [5] followed by Alon and Cohen [4] exhibit multiple examples of explicit $m \times n$ matrices that are (r, t)-rigid with

$$t \geqslant \min\left\{\frac{cn}{r}\log\frac{m}{r},\ n\right\} \tag{1}$$

where $m \geqslant n$ and c is a constant. Note that when $r = \epsilon n$, the current best bound is $t = \Omega\left(\log \frac{m}{n}\right)$. For $m = \mathrm{poly}(n)$, this amounts to $t = \Omega(\log n)$, exponentially far from the ideal bounds (i.e., matching random constructions). It is an important open problem to improve the dependence on m in Eq. (1) and to find other candidate sets that may be rigid with better parameters.

Our connection between rigidity and data structures arises via the inner product problem. The task is to preprocess a vector $v \in \mathbb{F}_2^n$ to compute inner products. The queries are specified by $Q \subseteq \mathbb{F}_2^n$, which is called the *query set*. The data structure must compute the inner product of v and any $q \in Q$, that is, $\langle q, v\rangle := \sum_{i=1}^{n} q[i] \cdot v[i] \mod 2$, where $q[i]$ denotes the i^{th} coordinate of q.

Consider the following model for solving this problem, known as a systematic linear data structure. During preprocessing, the data structure stores v along with the evaluations of r linear functions $\langle a_1, v\rangle, \ldots, \langle a_r, v\rangle$, where these inner products are single bits, and $a_1, \ldots, a_r$ denote vectors in $\mathbb{F}_2^n$. To compute the answer on query q, the data structure accesses these r bits in addition to any t entries of v. That is, the r linear functions are fixed, and the t bits from v may depend on q and the linear functions. Finally, the query algorithm must output a linear function of these r bits and the t entries of v. In this fashion it must be able to correctly compute $\langle q, v\rangle$ for all queries $q \in Q$. We note that a result of Jukna and Schnitger [19] shows that the $\{a_1, \ldots, a_r\}$ vectors do not depend on v without loss of generality. Letting $T(Q, r)$ denote the minimum value t of the best data structure for this problem (over worst-case v), we formalize the model as follows.

Definition (Systematic Linear Model)**.** *Let $Q \subseteq \mathbb{F}_2^n$ be a set. Define $T(Q, r)$ to be the maximum over all $v \in \mathbb{F}_2^n$ of the minimum t sufficient to compute the inner product $\langle q, v\rangle$ for every $q \in Q$ when only allowed to output a linear function of r precomputed linear functions of v along with any t bits of v.*

Note that the model does not charge the query time for accessing the r precomputed bits, even if $t \ll r$. This coincides with the systematic model studied by Chakraborty, Kamma and Larsen [8].

1.2 Equivalence between rigidity and data structures

We prove that the rigidity of a set Q corresponds to the time complexity $T(Q, r)$ in the systematic linear data structure model. Some aspects of this result are implicit in prior work [19, 31], but no previous work seems to show this exact correspondence.

Theorem 1. *A set $Q \subseteq \mathbb{F}_2^n$ is (r, t)-rigid if and only if $T(Q, r) \geqslant t$.*

Proof. We first prove that $T(Q, r) \geqslant t$ implies that Q is (r, t)-rigid. Assume for contradiction that there is an r-dimensional subspace U such that $d_H(q, U) < t$ for all $q \in Q$. Let $v \in \mathbb{F}_2^n$ be the input data. Store v along with the r bits $\langle b_1, v\rangle, \ldots, \langle b_r, v\rangle$, where $b_1, \ldots, b_r$ form a basis for U. For every $q \in Q$, there exists $u_q \in U$ such that $q - u_q$ has Hamming weight less than t. Using

the r redundant bits, the algorithm on query q can compute $\langle u_q, v\rangle$ by writing u_q in terms of the stored basis vectors. Then, it computes $\langle q - u_q, v\rangle$ by accessing fewer than t coordinates of v. Since $\langle q, v\rangle = \langle u_q, v\rangle + \langle q - u_q, v\rangle$, we have that $T(Q, r) < t$, which is a contradiction.

We now prove that if Q is (r, t)-rigid, then $T(Q, r) \geqslant t$. Let $e_1, \ldots, e_n$ denote the standard basis, and let $k = T(Q, r)$ be the query time. We show that $k \geqslant t$. Consider a systematic linear data structure whose redundant bits are given by $\langle a_1, v\rangle, \ldots, \langle a_r, v\rangle$. Let U denote the span of $\{a_1, \ldots, a_r\}$. As Q is (r, t)-rigid, there exists $q^* \in Q$ with $d_H(q^*, U) \geqslant t$. When q^* is the query, assume that the query algorithm accesses the bits $v_{i_1}, \ldots, v_{i_k}$ for indices $i_1, \ldots, i_k$ to compute $\langle q^*, v\rangle$. Now, define U' to be the span of $\{a_1, \ldots, a_r, e_{i_1}, \ldots, e_{i_k}\}$. Observe that all points in U' are at distance at most k from U. Thus, $d_H(q^*, U) \leqslant d_H(q^*, U') + k$. We will show that $d_H(q^*, U') = 0$, which implies that $k \geqslant t$. We claim that if $d_H(q^*, U') \geqslant 1$, then the query algorithm makes an error. Since $d_H(q^*, U') \geqslant 1$, there exists a vector y with $\langle y, q^*\rangle = 1$. Moreover, this vector can be taken to be orthogonal to U' so that $\langle y, x\rangle = 0$ for every $x \in U'$. In other words, for every $x \in U'$ we have $\langle y + v, x\rangle = \langle y, x\rangle + \langle v, x\rangle = \langle v, x\rangle$. Hence, the query algorithm sees the same values on input data $y + v$ and v because it only accesses the input via vectors in U', and we have $x \in U'$. Thus, the algorithm on query q^* must err either on input $y + v$ or v because $\langle q^*, y + v\rangle \neq \langle q^*, v\rangle$. □

1.3 Relationship to the cell probe model and other models

The systematic linear model specializes the *systematic model* [8, 17]. The latter model still stores the input data $x \in \mathbb{F}_2^n$ verbatim, and it also stores $r < n$ bits that can be precomputed from x, where these need not be linear functions of the input data. The query time is t if the query algorithm reads at most t bits from x to compute a query. The output can also be an arbitrary function of these t bits along with the r precomputed bits. The systematic linear model only makes sense for linear queries, whereas the systematic model applies to arbitrary query functions.

Yao's *cell probe model* is the most general data structure model [40]. On input data $x \in \mathbb{F}_2^n$, the data structure stores s cells, containing w bits that are arbitrary functions of x. Here, w is the *word size* and s is the *space*. The *query time* is t if the algorithm accesses at most t cells to answer any query about x from a set of m possible query functions. There is a rich collection of lower bounds for this model (see e.g. [2, 15, 20, 26, 27, 28, 29]). The best lower bounds known are of the form

$$t \geqslant \min\left\{\frac{c \log \frac{m}{n}}{\log \frac{sw}{n}}, \frac{cn}{w}\right\}, \tag{2}$$

where $m \geqslant n$ is the number of queries and c is a constant. It is a long-standing problem to prove that $t = \omega(\log m)$ for any explicit problem, even in the linear space regime $s \cdot w = O(n)$.

A special case of the cell probe model is the *linear model* [1, 13]. The latter model stores $s \geqslant n$ linear functions of x (implicitly $w = 1$ is fixed). The query time is t if the query algorithm reads at most t of these s bits to compute a query. The output is restricted to be a linear function of these t bits. A distinguishing aspect between linear and systematic linear is that in the latter model, the query algorithm is not charged for accessing the r precomputed bits. In Section 2, we compare the linear and systematic linear models in the context of rigidity and previous work [13].

Equivalences all the way down. We note that the systematic data structure model is identical to the common bits model defined by Valiant [36]. Corrigan-Gibbs and Kogan [11] demonstrate a relationship between the common bits model and a variant of the systematic model defined by Gal and Miltersen [17]. The common bits model is nothing but a certain depth two circuit, and the systematic linear model is simply the common bits model with the restriction that the common bits and output gates are linear functions [31]. Hence, in language of data structures, the linearization conjecture of Jukna and Schnitger posits that the systematic linear model is asymptotically as powerful as the systematic model for answering linear queries [19].

1.4 The vector-matrix-vector problem

We now define the vector-matrix-vector problem, which we call "the $u^\intercal Mv$ problem" for short. Let n be a perfect square. After preprocessing a matrix $M \in \mathbb{F}_2^{\sqrt{n}\times\sqrt{n}}$, the goal is to output the binary value $u^\intercal Mv$ for vectors $u, v \in \mathbb{F}_2^{\sqrt{n}}$. It will be convenient to consider a $\sqrt{n} \times \sqrt{n}$ matrix as an n-bit vector $\mathsf{vec}(M)$ by concatenating consecutive rows. More formally, let $x = \mathsf{vec}(M)$, and for $i \in \{1, 2, \ldots, n\}$, set $x[i] = M[a, b]$, where a and b satisfy $i = (a-1)\sqrt{n} + b$ and $a, b \in \{1, 2, \ldots, \sqrt{n}\}$. Then, $u^\intercal Mv = \langle \mathsf{vec}(uv^\intercal), \mathsf{vec}(M)\rangle$. In this way we consider the $u^\intercal Mv$ problem a special case of the inner product problem. The query set is the collection of rank one binary matrices. Let $\Upsilon \subseteq \mathbb{F}_2^n$ denote the set of vectors obtained from rank one binary matrices via $M \mapsto \mathsf{vec}(M)$, that is,

$$\Upsilon := \left\{ \mathsf{vec}(uv^\intercal) \;\middle|\; u, v \in \mathbb{F}_2^{\sqrt{n}\times\sqrt{n}} \right\} \subseteq \mathbb{F}_2^n. \tag{3}$$

This set has size $|\Upsilon| = 2^{2\sqrt{n}} - 2^{\sqrt{n}+1} + 1$.

A classic result of Artazarov, Dinic, Kronrod and Faradzev [6] provides a data structure with space $s = \mathsf{poly}(n)$, word size $w = O(\log n)$, and time $t = O(n/\log n)$. In fact, this algorithm operates in the linear cell probe model. It is a central open question to determine whether $t = \Omega(n)$ is necessary in linear space regime, that is, when $s \cdot w = O(n)$.

The current best cell probe lower bound for the $u^\intercal Mv$ problem is due to Chattopadhyay, Koucký, Loff, and Mukhopadhyay [9]. Moreover, their lower bound holds for a randomized model with high error. For constants c and c', they prove that if for every matrix M and every query $uv^\intercal$, the query algorithm correctly computes $u^\intercal Mv$ with probability at least $\frac{1}{2} + \frac{1}{2^{c'\sqrt{n}}}$, then

$$t \geqslant \min\left\{ \frac{c\sqrt{n}}{\log \frac{sw}{\sqrt{n}}}, \frac{cn}{w} \right\} \tag{4}$$

Better lower bounds for the $u^\intercal Mv$ problem are known in the systematic model. Chakraborty, Kamma, and Larsen [8] prove that t and r must satisfy $t \cdot r = \Omega(n^{3/2}/\log n)$ as long as $r \geqslant \sqrt{n}$. In the case of $r \leqslant \sqrt{n}$, they prove that $t = \Omega(n/\log n)$. As the systematic model subsumes the linear version of this model, combining their result with Theorem 1 implies that Υ is (r, t)-rigid with

$$t = \Omega\left(\frac{n^{3/2}}{\max\{\sqrt{n}, r\} \cdot \log n} \right). \tag{5}$$

1.5 New results on the rigidity of Υ and the cell probe complexity of the $u^\intercal Mv$ problem

We lower bound the rigidity of Υ, defined in Eq. (3). This also implies a lower bound in the systematic linear model. The proof is inspired by a result of Alon, Panigrahy, and Yekahnin [5].

Theorem 2. *Let* $n \geqslant 1024$. *The set* $\Upsilon \subseteq \mathbb{F}_2^n$ *of rank one matrices is* (r, t)*-rigid with* $t \geqslant \frac{n^{3/2}}{128 \cdot \max\{\sqrt{n}, r\}}$.

We improve the prior bound in Eq. (5) by an $\Omega(\log n)$ factor. For example, when $r \leqslant \sqrt{n}$, then $t = \Omega(n)$, and when $r = n/2$, then $t = \Omega\left(n^{1/2}\right)$. Theorem 2 matches Eq. (1), the current best bound for explicit rigid sets. We do not know whether there is a subspace U of linear dimension such that all elements of Υ are at distance $o(n)$ from U (unlike for some set rigidity results, where the bounds are tight). As a corollary of Theorem 1, we immediately get that

$$T(\Upsilon, r) \geqslant \frac{n^{3/2}}{128 \cdot \max\{\sqrt{n}, r\}}.$$

In other words, we prove a lower bound for the $u^\intercal Mv$ problem in the systematic linear model that improves the prior bound by an $\Omega(\log n)$ factor. The proof of Theorem 2 appears in Section 3.

We also prove a general cell probe lower bound for the uMv problem in the high error regime. Our result improves the previous lower bound in Eq. (4). For example, in the linear space regime, when $s \cdot w = O(n)$, we show that $t = \Omega(\sqrt{n})$ while the prior result gives only $t = \Omega(\sqrt{n}/\log n)$.

Theorem 3. *Let* $M \in \mathbb{F}_2^{\sqrt{n} \times \sqrt{n}}$ *be a matrix. If a randomized data structure with space* s, *word size* w, *and time* t *correctly computes queries for the* $u^\intercal Mv$ *problem with probability at least* $\frac{1}{2} + \frac{1}{2^{\sqrt{n}/64}}$, *then*

$$t \geqslant \min\left\{\frac{c\sqrt{n}}{\log \frac{s\alpha}{n}}, \frac{cn}{\alpha}\right\}$$

where $0 < c \leqslant 1/36$ *is a universal constant and* $\alpha := 2(w + \log \frac{sw}{n})$.

The prior work utilizes a general lifting result for two-way communication complexity from parity decision trees [9]. To obtain the improved bound, we use a variant of the cell sampling technique [21, 27] combined with a reduction to a new lower bound on one-way communication (via discrepancy). The modifications over standard techniques are needed to handle the high error regime for a binary output problem. We note that a recent result of Larsen, Weinstein and Yu also uses one-way communication to prove lower bounds for binary output problems for dynamic data structures [22]. However, their method seems limited to only handling zero error query algorithms. The proof of Theorem 3 appears in Section 4. Specifically, see Lemma 14 in Section 4 for the variant of cell sampling and see Theorem 11 in Section 4.1 for the discrepancy argument.

2 Linear Data Structures and Rigidity

In this section, we relate linear data structures and rigidity. As linear data structures are a special case of the cell probe model, we may obtain rigidity lower bounds from strong enough static data

structure lower bounds (when the queries are linear). We also compare with Dvir, Golovnev, and Weinstein, who exhibit a similar connection [13]. We first provide some notation.

Definition. *Let $Q \subseteq \mathbb{F}_2^n$ be a set. Define $\mathsf{LT}(Q, s)$ to be the maximum over all $v \in \mathbb{F}_2^n$ of the minimum t sufficient to compute the inner product $\langle q, v\rangle$ for every $q \in Q$ when the query algorithm's output is a linear function of t bits chosen from the s precomputed linear functions of v.*

Table 1 provides a glimpse of our results on linear data structures along with a comparison to [13]. Recall that a set $Q \subseteq \mathbb{F}_2^n$ is explicit if each coordinate of an arbitrary element of the set can be computed in $\mathsf{poly}(n)$ time. The prior work shows that sufficiently strong lower bounds against linear data structures will imply *semi-explicit* rigid sets. A bit more formally, consider a data structure query set $Q \subseteq \mathbb{F}_2^n$ of size m for the inner product problem. They show the following: If $\mathsf{LT}(Q, c \cdot n) \geqslant t$ for some constant c, then there is a $(n'/2, t/\log n)$-rigid set Q' of size at most m contained in $\mathbb{F}_2^{n'}$, where $n' \geqslant t$. However, the set Q' is only semi-explicit in that it is in P^{NP} – every element can be computed by a $\mathsf{poly}(m)$ time algorithm with access to an NP oracle.

We now summarize a few differences between our work and [13]. Our result proves that polynomial lower bounds on the query time imply the existence of an explicit rigid set, which is in contrast to semi-explicit sets obtained by [13]. On the other hand, explicitness comes with a cost; when $m = \mathsf{poly}(n)$, we need much stronger data structure lower bounds to produce explicit rigid sets. When $m \gg \mathsf{poly}(n)$, the algorithm of [13] takes $\mathsf{poly}(m)$ time with access to an NP oracle to compute an element of the semi-explicit rigid set. For problems such as the $u^\intercal M v$ problem, this is super polynomial time. The rest of this section concerns proving the following theorem, which implies all of our results in Table 1.

Theorem 4. *Let $k = \mathsf{LT}(Q, 3n/2)$ and let $Q \subseteq \mathbb{F}_2^n$ of size m be an explicit query set. There exists a set $Q' \subseteq \mathbb{F}_2^k$ with size at most $m \cdot \lceil \frac{n}{k} \rceil$, whose elements can be computed in $\mathsf{poly}(n)$ time. Moreover, if $k \geqslant 2\sqrt{n}$, then Q' is explicit and $\left(\frac{k}{2}, \frac{k^2}{4n}\right)$-rigid.*

Note that for every $s \geqslant 3n/2$, we have that $\mathsf{LT}(Q, 3n/2) \geqslant \mathsf{LT}(Q, s)$. Hence, a sufficiently strong lower bound on $\mathsf{LT}(Q, s)$ for any $s \geqslant 3n/2$ will imply a rigidity lower bound. The following corollary shows the consequence of Theorem 4 for specific values of k.

Corollary 5. *Let $k = \mathsf{LT}(Q, 3n/2)$ and let $Q \subseteq \mathbb{F}_2^n$ of size m be an explicit query set. There exists a set $Q' \subseteq \mathbb{F}_2^k$ with size at most $m \cdot \lceil \frac{n}{k} \rceil$, whose elements can be computed in $\mathsf{poly}(n)$ time. Moreover,*

(a) If $k = \omega\left(\sqrt{n \log m}\right)$, then Q' is explicit and $(k/2, \omega(\log m))$-rigid.

(b) If $k = \Omega\left(n^{(1+\delta)/2}\right)$ for some $\delta > 0$, then Q' is explicit and $\left(k/2, \Omega\left(n^{\delta}\right)\right)$-rigid.

Corollary 5(a) explains the first and last rows in Table 1, and Corollary 5(b) explains the middle row. Using Corollary 5(a) applied to Υ with $m = 2^{2\sqrt{n}} - 2^{\sqrt{n}+1} + 1$, we obtain that a lower bound of $\mathsf{LT}(\Upsilon, 3n/2) \geqslant \omega(n^{3/4})$ would imply the existence of an explicit set $Q' \subseteq \mathbb{F}_2^k$ of size $2^{O(\sqrt{n})}$ that is $(k/2, \omega(\sqrt{n}))$-rigid. We note that it is an open question to prove $\mathsf{LT}(\Upsilon, 3n/2) \geqslant \omega(\sqrt{n})$.

m vs n	$k = \mathsf{LT}(Q, 3n/2)$	Rigidity Bounds	Explicitness	Reference
$m = n^c$	$k = \omega\left(\sqrt{n \log n}\right)$	$(k/2, \omega(\log n))$-rigid	$\mathsf{poly}(n)$ time	This work
	$k = \omega\left(\log^2 n\right)$	$(k/2, \omega(\log n))$-rigid	$\mathsf{poly}(n)$ time + NP oracle calls	[13]
$m = n^c$	$k = \Omega\left(n^{(1+\delta)/2}\right)$	$\left(k/2, \Omega\left(n^{\delta}\right)\right)$-rigid	$\mathsf{poly}(n)$ time	This work
	$k = \Omega\left(n^{\delta} \log n\right)$	$\left(k/2, \Omega\left(n^{\delta}\right)\right)$-rigid	$\mathsf{poly}(n)$ time + NP oracle calls	[13]
$m = 2^{c\sqrt{n}}$	$k = \omega\left(n^{3/4}\right)$	$\left(k/2, \omega\left(\sqrt{n}\right)\right)$-rigid	$\mathsf{poly}(n)$ time	This work
	$k = \omega\left(\sqrt{n} \cdot \log n\right)$	$\left(k/2, \omega\left(\sqrt{n}\right)\right)$-rigid	$\mathsf{poly}\left(2^{\sqrt{n}}\right)$ time + NP oracle calls	[13]

Table 1: Comparison with [13, Theorem 7.1]: Let $Q \subseteq \mathbb{F}_2^n$ of size m be a query set, $c \geqslant 1$ and $\delta > 0$ be constants, and let $k = \mathsf{LT}(Q, 3n/2)$. The second column states the lower bound on $\mathsf{LT}(Q, 3n/2)$ that implies existence of rigid sets whose parameters are given in the third column. All rigid sets have size at most $\mathsf{poly}(m)$ and are contained in $\mathbb{F}_2^k$.

2.1 Proof of Theorem 4

We already know the equivalence between systematic linear data structures and rigidity (from Theorem 1). Therefore, it is sufficient to design a linear data structure from a systematic linear data structure to relate the former with rigidity.

Proposition 6. *Let $Q \subseteq \mathbb{F}_2^n$ be a query set. If $\mathsf{T}(Q, r) \leqslant t$, then $\mathsf{LT}(Q, n + r) \leqslant t + r$.*

Proof. Let $v \in \mathbb{F}_2^n$ be the input data, and let $\langle a_1, v\rangle, \ldots, \langle a_r, v\rangle$ be the r redundant bits stored by the systematic linear data structure. We now describe a linear data structure for Q with space $n + r$ and query time $t + r$. The data structure stores $\langle a_1, v\rangle, \ldots, \langle a_r, v\rangle, \langle e_1, v\rangle, \ldots, \langle e_n, v\rangle$, where $e_1, \ldots, e_n$ are the standard basis vectors. The query algorithm on $q \in Q$ first accesses $\langle a_1, v\rangle, \ldots, \langle a_r, v\rangle$ and then simulates the query algorithm of the systematic linear data structure on q. Since the systematic linear data structure accesses at most t bits from $\langle e_1, v\rangle, \ldots, \langle e_n, v\rangle$, we can conclude that the query time is at most $t + r$. □

We prove that if a set contained in a n-dimensional space is (r, t)-rigid, then there is another $(r, tr/n)$-rigid set which is contained in a $2r$-dimensional space.

Lemma 7. *Let r, n be positive integers. If $S \subseteq \mathbb{F}_2^n$ is (r, t)-rigid of size m, then there is a set $S' \subseteq \mathbb{F}_2^{2r}$ of size at most $m \cdot \left\lceil \frac{n}{2r} \right\rceil$ that is $(r, tr/n)$-rigid. Moreover, if S is explicit, then each element of S' can be computed in $\mathsf{poly}(n)$ time.*

Proof. Let $k = \left\lfloor \frac{n}{2r} \right\rfloor$ and define $S_1, \ldots, S_k \subseteq \mathbb{F}_2^{2r}$ by

$$S_i = \{(s[2r \cdot (i-1) + 1], \ldots, s[2r \cdot i]) \mid s \in S\}$$

for each $i \in \{1, 2, \ldots, k\}$. Additionally, if $n/2r$ is not an integer, then define

$$S_{k+1} = \{(s[2r \cdot k + 1], \ldots, s[n], 0, \ldots, 0) \mid s \in S\} \subseteq \mathbb{F}_2^{2r};$$

otherwise set $S_{k+1} = \emptyset$. Define $S' = \bigcup_{i=1}^{k+1} S_i$. We claim that S' is $(r, tr/n)$-rigid. Indeed, for the sake of contradiction assume that there is a subspace V in $\mathbb{F}_2^{2r}$ of dimension r such that all points in S' are at a distance less than tr/n from V. Consider the subspace $\{(v, v, \ldots, v) \mid v \in V\} \subseteq \mathbb{F}_2^{2r\cdot(k+1)}$ and project it to the first n coordinates. Call this subspace V', which has dimension r. Now, the distance of each point in S from V' is less than $\frac{tr}{n} \cdot \left\lceil \frac{n}{2r} \right\rceil < t$, which is a contradiction.

Regarding the explicitness of S', it is clear that all coordinates of an element of S' correspond to some coordinate of a specific element of S. Since S is explicit, we can infer that each element of S' can be computed in $\mathsf{poly}(n)$. □

Proof of Theorem 4. Since $\mathsf{LT}(Q, 3n/2) = k$ and $k \leqslant n$, Proposition 6 implies that $T(Q, k/2) \geqslant k/2$. Therefore by Theorem 1, we can conclude that Q is $(k/2, k/2)$-rigid. Lemma 7 implies that there exists a set Q' that is $\left(\frac{k}{2}, \frac{k^2}{4n}\right)$-rigid and the size of Q' is at most $m \cdot \left\lceil \frac{n}{k} \right\rceil$. Moreover, every element of Q' can be computed in time $\mathsf{poly}(n)$. Since $k/2 \geqslant \sqrt{n}$, we can conclude that Q' is explicit. □

3 Rigidity Lower Bounds for the Set of Rank One Matrices

Before proving Theorem 2, we present preliminaries. Recall two standard binomial estimates:

Proposition 8. *For integers $0 \leqslant k \leqslant \ell$,*

1. $\log \binom{\ell}{k} \leqslant k \cdot \log \frac{e\ell}{k}$.
2. *if $k \leqslant \ell/16$, then $\sum_{i=0}^{k} \binom{\ell}{i} \leqslant 2^{\ell/4}$.*

We will need a useful property about the distance of a point from a subspace.

Lemma 9. *Let $V \subseteq \mathbb{F}_2^\ell$ be a subspace. For $u_1, u_2 \in \mathbb{F}_2^\ell$, $d_H(u_1 + u_2, V) \leqslant d_H(u_1, V) + d_H(u_2, V)$.*

Proof. Let $u_1', u_2' \in V$ be the points in V closest to u_1 and u_2 respectively. Since $u_1' + u_2' \in V$, we have

$$d_H(u_1 + u_2, V) \leqslant d_H(u_1 + u_2, u_1' + u_2') = d_H(u_1 + u_2, u_1' + u_2').$$

Note that $d_H(u_1 + u_2, u_1' + u_2')$ is the number of ones in $u_1 + u_2 + u_1' + u_2'$, which is at most the sum of the number of ones in $u_1 + u_1'$ and $u_2 + u_2'$. Therefore,

$$d_H(u_1 + u_2, u_1' + u_2') \leqslant d_H(u_1, u_1') + d_H(u_2, u_2') = d_H(u_1, V) + d_H(u_2, V). \qquad \square$$

A simple counting argument establishes the existence of a point that is far away in Hamming distance from a collection of large sized sets.

Lemma 10. *Let $V_1, \ldots, V_k$ be subsets of $\mathbb{F}_2^\ell$, each of size at most $2^{\ell/2}$. If $k < 2^{\ell/4}$, then there is a vector $v \in \mathbb{F}_2^\ell$ such that the Hamming distance of v from each V_i is at least $\ell/16$.*

Proof. For every $i \in [k]$, define $B(V_i, \ell/16) = \left|\left\{v \in \mathbb{F}_2^\ell \;\middle|\; d_H(v, V_i) < \ell/16\right\}\right|$. For any $u \in V_i$, the number of vectors in $\mathbb{F}_2^\ell$ at a distance less than $\ell/16$ from u is at most $\sum_{j=0}^{\ell/16} \binom{\ell}{j} \leqslant 2^{\ell/4}$, where the

inequality follows from Proposition 8. Hence $B(V_i, \ell/16) \leqslant |V_i| \cdot 2^{\ell/4} = 2^{3\ell/4}$. Since

$$\sum_{i=1}^{k} B(V_i, \ell/16) \leqslant k \cdot 2^{3\ell/4} < 2^{\ell},$$

there is a $v \in \mathbb{F}_2^{\ell}$ such that $d_H(v, V_i) \geqslant \ell/16$ for every $i \in [k]$. □

3.1 Proof of Theorem 2

Let V be any r'-dimensional subspace of $\mathbb{F}_2^n$, where $r' \geqslant r$ is the smallest positive integer divisible by $\sqrt{n}$. We first define the inverse of $\mathsf{vec}(\cdot)$. For every $v \in \mathbb{F}_2^n$, define $\mathsf{mat}(v)$ to be the matrix obtained by splitting v into $\sqrt{n}$ length consecutive blocks and stacking each of these blocks to form a $\sqrt{n} \times \sqrt{n}$ matrix. Formally, $\mathsf{mat}(v) \in \mathbb{F}_2^{\sqrt{n} \times \sqrt{n}}$ is such that $\mathsf{mat}(v)[a, b] = v[(a-1)\sqrt{n} + j]$ for every $a, b \in [\sqrt{n}]$. Note that $\mathsf{vec}(\mathsf{mat}(v)) = v$.

We provide a brief outline of the proof of Theorem 2. The first step of the proof is to produce a vector in v that is at a distance of $\Omega(n)$ from V and $\mathsf{mat}(v)$ is low rank. The rank being low is helpful as we can express $\mathsf{mat}(v)$ as the sum of a small number of rank one matrices. Lemma 9 will then imply the existence of a rank one matrix that is far away from V. If we only cared about the existence of a vector that is far away from V, Lemma 10 would suffice. To ensure that simultaneously the rank is small, we first project V on to $n/2r'$ coordinates indexed by consecutive blocks each of length $2r'$. Then we find a vector $v' \in \mathbb{F}_2^{2r'}$ that is far away from all the projections, which is still guaranteed by Lemma 10. Concatenating v' with itself $2r'$ times has the property that its corresponding matrix is low rank.

Let $k = \max\left\{\left\lfloor \frac{n}{2r'} \right\rfloor, 1\right\}$. The goal is to find a $v \in \mathbb{F}_2^n$ such that $d_H(v, V) \geqslant k \cdot r'/8$ and the rank of $\mathsf{mat}(v)$ is at most $2r'/\sqrt{n}$. If $\left\lfloor \frac{n}{2r'} \right\rfloor \geqslant 1$, then define $S_1, \ldots, S_k$ such that

$$S_i = \left\{(i-1) \cdot 2r' + 1, \ldots, i \cdot 2r'\right\}$$

for $i \in [k]$; otherwise, define $S_1 = [n]$. By definition, the dimension of V_{S_i} is at most $r' = |S_i|/2$, for every $i \in [k]$. Since $r' \geqslant \sqrt{n}$ and $n \geqslant 1024$, we can infer that $k \leqslant 2r'$ and $2r' < 2^{r'/2}$. Lemma 10 implies the existence of a $v' \in \mathbb{F}_2^{2r'}$ with the property that $d_H(v', V_{S_i}) \geqslant r'/8$ for every $i \in [k]$. Now define $v \in \mathbb{F}_2^n$ by

$$v[i] = \begin{cases} v'[i \bmod 2r'] & \text{if } i \leqslant k \cdot 2r' \text{ and } i \bmod 2r' \neq 0, \\ v'[2r'] & \text{if } i \leqslant k \cdot 2r' \text{ and } i \bmod 2r' = 0, \\ 0 & \text{if } i > 2kr', \end{cases}$$

for all $i \in [n]$. In words, v is the length n vector that is the concatenation of k copies of v' along with the vector of zeros of length $n - 2kr'$. By the choice of v, we get that,

$$d_H(v, V) \geqslant \sum_{i=1}^{k} d_H(v, V_{S_i}) \geqslant k \cdot r'/8.$$

Moreover, the rank of $\mathsf{mat}(v)$ is at most $\frac{2r'}{\sqrt{n}}$. Therefore we can express

$$\mathsf{mat}(v) = \sum_{i=1}^{2r'/\sqrt{n}} a_i b_i^\intercal,$$

for some $a_1, b_1, \ldots, a_{\frac{2r'}{\sqrt{n}}}, b_{\frac{2r'}{\sqrt{n}}} \in \mathbb{F}_2^{\sqrt{n}}$. By Lemma 9, we know that

$$d_H(v, V) \leqslant \sum_{i=1}^{2r'/\sqrt{n}} d_H\left(\mathsf{vec}(a_i b_i^\intercal), V\right).$$

Hence there exists an $i \in \left[\frac{2r'}{\sqrt{n}}\right]$ such that $d_H\left(\mathsf{vec}(a_i b_i^\intercal), V\right) \geqslant \frac{\sqrt{n}\cdot k}{16} \geqslant \frac{n^{3/2}}{64r'}$. The observation that $r' \leqslant 2\max\{\sqrt{n}, r\}$ completes the proof of the theorem.

Remark (Extension to strong rigidity)**.** Alon and Cohen [4] defined the notion of *strong* rigidity; a set $Q \subseteq \mathbb{F}_2^n$ is (r,t)-strongly rigid if for every subspace of $\mathbb{F}_2^n$ of dimension at most r, the average distance of all the points to the subspace is at least t. For strong rigidity, the best lower bounds known for explicit sets are also of the form given in Eq. (1). We can show that Υ is (r,t)-strongly rigid with $t \geqslant \Omega\left(\frac{n^{3/2}}{\max\{\sqrt{n},r\}}\right)$, matching the best strong rigidity bounds known for explicit sets. We sketch the proof here. We know that

$$u^\intercal M v + (u+e_i)^\intercal M v + u^\intercal M(v+e_j) + (u+e_i)^\intercal M(v+e_j) = b_i^\intercal M b_j,$$

where $u, v \in \mathbb{F}_2^{\sqrt{n}}$ and $e_1, \ldots, e_{\sqrt{n}}$ are standard basis vectors in $\mathbb{F}_2^{\sqrt{n}}$. This fact can be used to prove that the matrix M_Υ corresponding to the set Υ is a generator matrix of a 4-query locally decodable code that tolerates a constant fraction of errors. A result of [13, Theorem 6] shows that Theorem 2 and the locally decodable code property of M_Υ imply the strong rigidity of Υ.

4 Cell Probe Lower Bounds for the $u^\intercal M v$ Problem

We know of two techniques for proving cell probe lower bounds matching Eq. (2). One is a technique of Pătraşcu and Thorup [30] who combined the communication complexity simulation of Miltersen [25] with multiple queries on the same input data. The other is the technique we use, which is based on cell sampling. Cell sampling typically requires one to work with large sized fields in order to handle errors. This large field size is needed to encode a large subset of the correctly computed queries using a small subset of cells. Here, we avoid encoding the subset of queries by a reduction to one-way communication complexity.

Proof outline for Theorem 3. By Yao's min-max principle, it suffices to prove a lower bound on deterministic data structures. The hard distribution on the input data M and query $uv^\intercal$ we use is given by sampling $M, (u,v)$ uniformly and independently at random. We prove the theorem by

contradiction, and we start by assuming that the query time is small. The proof is carried out in three steps. First, modify the data structure so that for every M, the fraction of queries correctly computed is at least $1/2$. This modification only increases the query time and space by 1, and it can only increase the overall probability of the query algorithm being correct. Second, for a given M, we use a variant of cell sampling (see Lemma 14) to obtain a small subset of cells S and a large subset of queries Q' such that all queries in Q' can be computed by only accessing cells in S. Moreover,

$$\begin{aligned}&\Pr\left[\text{query algorithm correctly computes } u^\intercal M v \;\middle|\; uv^\intercal \in Q'\right]\\ &\approx \Pr\left[\text{query algorithm correctly computes } u^\intercal M v\right].\end{aligned}$$

Third, we show that S can be used to design an efficient protocol for the following communication game: Alice's input is M and Bob's input is $uv^\intercal$, and the goal is for Bob to correctly compute $u^\intercal M v$ on a sufficiently good fraction of the inputs after receiving a message from Alice.

We now describe the protocol (see Figure 1). Alice sends the locations and contents of S. This ensures that Bob correctly computes $u^\intercal M v$ on a large fraction of queries in Q'. Alice also communicates the majority value of $u^\intercal M v$ for $uv^\intercal \notin Q'$ so that Bob is correct on half of his possible inputs that are not in Q'. Overall, Bob's output is correct on a sufficiently good fraction of all $M, (u, v)$. Since we have assumed that the query time is small, we are able to show that Alice's communication is small. This contradicts a lower bound on the communication complexity of this game. More precisely, we prove the following lower bound.

Theorem 11. *Suppose that Alice gets a uniformly random matrix $M \in \mathbb{F}_2^{\sqrt{n}\times\sqrt{n}}$ as input and Bob receives a uniform pair $(u, v) \in \mathbb{F}_2^{\sqrt{n}} \times \mathbb{F}_2^{\sqrt{n}}$ as input. If Alice sends a deterministic message to Bob and Bob computes $u^\intercal M v$ such that*

$$\Pr_{M,u,v}[\textit{Bob computes } u^\intercal M v \textit{ correctly}] \geqslant \frac{1}{2} + \frac{1}{2^{\sqrt{n}/8}},$$

then Alice must communicate at least $n/10$ bits.

Previously, in the *randomized* two-way communication setting, Chattopadhyay, Koucký, Loff, and Mukhopadhyay [9] proved a lower bound for the game given in Theorem 11. Their lower bound implies the lower bound in Theorem 11 against randomized protocols. We need a lower bound against *deterministic* protocols under the *uniform distribution* on the inputs, and we cannot use their theorem as a black-box. We provide a straightforward proof of Theorem 11 in Section 4.1 by using the *discrepancy method* on a related communication game (resembling a direct sum, where Bob receives multiple inputs).

Preliminaries. Before presenting the proof of Theorem 3, we define some notation. For a real valued function f with a finite domain $X \times Y$, $\mathbb{E}_{x,y}[f(x, y)] = \frac{1}{|X|\cdot|Y|} \cdot \sum_{x\in X, y\in Y} f(x, y)$. Similarly, for $X' \subseteq X$, $\mathbb{E}_{x,y}[f(x, y) \mid x \in X'] = \frac{1}{|X'|\cdot|Y|} \cdot \sum_{x\in X', y\in Y} f(x, y)$. An argument in the proof of

Theorem 3 requires an upper bound on the number of bits to encode the contents and locations of a subset of the cells, which is given by the following proposition.

Proposition 12. *Let S be a subset of the cells of a data structure with word length w and size s. Then, the contents and locations of S can be encoded in $|S| \cdot w + |S| \cdot \log \frac{es}{|S|}$ bits.*

Proof. Since each cell stores w bits, the number of bits to encode the contents is $|S| \cdot w$. Since the total number of cells is s, the locations can be encoded in $\log \binom{s}{|S|} \leqslant |S| \cdot \log \frac{es}{|S|}$ bits, where the inequality followed from Proposition 8. □

4.1 Proof of Theorem 11

We start by discussing a slightly related problem, whose solution will lead to the proof strategy used here. Let $M \in \mathbb{F}_2^{\sqrt{n} \times \sqrt{n}}$, $v \in \mathbb{F}_2^{\sqrt{n}}$, and $e_1, \ldots, e_{\sqrt{n}}$ be the standard basis vectors in $\mathbb{F}_2^{\sqrt{n}}$. Consider the communication game in which Alice gets as input a uniform random M and Bob gets as input a uniform random pair (e_i, v). Bob's goal is to compute $e_i^\intercal M v$ after receiving a message from Alice. To understand how much Alice has to communicate, it is natural to look at the problem where Bob computes $\sum_{i=1}^{\sqrt{n}} e_i^\intercal M v_i$, where $v_1, \ldots, v_{\sqrt{n}} \in \mathbb{F}_2^{\sqrt{n}}$. Now observe that this sum is the same as the trace of $\left(\sum_{i=1}^{\sqrt{n}} e_i v^\intercal\right) M$, which in turn is the inner product between two n-bit vectors. The communication complexity of the inner product between two n-bit vectors is very well understood. Therefore, the lower bound on the amount of communication to compute the inner product between two n-bit vectors translates to a lower bound to the problem of computing $e_i^\intercal M v$. This strategy applied to our setting gives us the following lower bound, which will be used to prove Theorem 11. Our presentation closely follows [32, Chapter 5].

Lemma 13. *Let $0 < \epsilon \leqslant 1/2$ and let k be an integer. Alice gets a uniformly random $M \in \mathbb{F}_2^{\sqrt{n} \times \sqrt{n}}$ as input and Bob receives k uniform pairs $(u_1, v_1), \ldots, (u_k, v_k) \in \mathbb{F}_2^{\sqrt{n}} \times \mathbb{F}_2^{\sqrt{n}}$ as input. Assume that Alice communicates a deterministic message to Bob, and Bob computes $\sum_{i=1}^{k} u_i^\intercal M v_i$ with*

$$\Pr_{M, u_1, v_1, \ldots, u_k, v_k} \left[\textit{Bob computes } \sum_{i=1}^{k} u_i^\intercal M v_i \textit{ correctly} \right] \geqslant \frac{1}{2} + \epsilon.$$

If $k \leqslant \sqrt{n}$, then Alice must communicate at least $9k\sqrt{n}/40 - \log(1/\epsilon)$ bits.

Proof. We use the discrepancy method to prove the communication lower bound. This requires upper bounding the discrepancy of the *communication matrix* under a given distribution. Let R be a rectangle of the communication matrix, which is defined by indicator functions A_R and B_R such that $(M, ((u_1, v_1), \ldots, (u_k, v_k)))$ is in the rectangle R if and only if $A_R(M) = 1$ and $B_R((u_1, v_1), \ldots, (u_k, v_k)) = 1$.

Consider the distribution in which $M, (u_1, v_1), \ldots, (u_k, v_k)$ are chosen at random uniformly and independently. We upper bound the discrepancy under this distribution. In other words, we

claim that for every rectangle R,

$$\mathop{\mathbb{E}}_{M,(u_1,v_1),\ldots,(u_k,v_k)} \left[A_R(M) B_R\left((u_1,v_1),\ldots,(u_k,v_k)\right) (-1)^{\sum_{i=1}^k u_i^\intercal M v_i} \right] \leqslant 2 \cdot 2^{-9k\sqrt{n}/40}. \tag{6}$$

By a standard relation in communication complexity between the number of bits communicated and discrepancy of rectangles (see [32, Chapter 5, Theorem 5.2]), Eq. (6) implies that Alice must communicate at least $9k\sqrt{n}/40 - \log(1/\epsilon)$ bits. We are left with the proof of Eq. (6).

$$\begin{aligned}
&\left(\mathop{\mathbb{E}}_{(u_1,v_1),\ldots,(u_k,v_k)} \left[B_R\left((u_1,v_1),\ldots,(u_k,v_k)\right) \mathop{\mathbb{E}}_{M} \left[A_R(M)(-1)^{\sum_{i=1}^k u_i^\intercal M v_i} \right] \right] \right)^2 \\
&\leqslant \mathop{\mathbb{E}}_{(u_1,v_1),\ldots,(u_k,v_k)} \left[B_R\left((u_1,v_1),\ldots,(u_k,v_k)\right)^2 \left(\mathop{\mathbb{E}}_{M} \left[A_R(M)(-1)^{\sum_{i=1}^k u_i^\intercal M v_i} \right] \right)^2 \right] \\
&\leqslant \mathop{\mathbb{E}}_{(u_1,v_1),\ldots,(u_k,v_k)} \left[\left(\mathop{\mathbb{E}}_{M} \left[A_R(M)(-1)^{\sum_{i=1}^k u_i^\intercal M v_i} \right] \right)^2 \right].
\end{aligned}$$

where the first inequality follows from convexity and the second one follows from the fact that $B_R\left((u_1,v_1),\ldots,(u_k,v_k)\right) \leqslant 1$. Now

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(u_1,v_1),\ldots,(u_k,v_k)} \left[\left(\mathop{\mathbb{E}}_{M} \left[A_R(M)(-1)^{\sum_{i=1}^k u_i^\intercal M v_i} \right] \right)^2 \right] \\
&\leqslant \mathop{\mathbb{E}}_{(u_1,v_1),\ldots,(u_k,v_k),M,M'} \left[A_R(M) A_R(M')(-1)^{\sum_{i=1}^k u_i^\intercal M v_i + \sum_{i=1}^k u_i^\intercal M' v_i} \right] \\
&\leqslant \mathop{\mathbb{E}}_{M,M'} \left[\left| \mathop{\mathbb{E}}_{(u_1,v_1),\ldots,(u_k,v_k)} \left[(-1)^{\sum_{i=1}^k u_i^\intercal (M+M') v_i} \right] \right| \right] \\
&= \mathop{\mathbb{E}}_{M} \left[\left| \mathop{\mathbb{E}}_{(u,v)} \left[(-1)^{u^\intercal M v} \right] \right|^k \right],
\end{aligned}$$

where the last equality follows from the fact that $(u_1,v_1),\ldots,(u_k,v_k)$ are chosen independent of each other and $M + M'$ is uniformly distributed as M and M' are chosen uniformly and independently at random. We are left with upper bounding $\mathbb{E}_M \left[\left| \mathbb{E}_{(u,v)} \left[(-1)^{u^\intercal M v} \right] \right|^k \right]$. First note that if M has rank r, then $\mathbb{E}_{u,v} \left[(-1)^{u^\intercal M v} \right] = 2^{-r}$. This is because,

$$\mathop{\mathbb{E}}_{u,v} \left[(-1)^{u^\intercal M v} \right] = \frac{1}{2\sqrt{n}} \cdot \left(\sum_{v : Mv = 0} 1 \right) + \frac{1}{2\sqrt{n}} \cdot \left(\sum_{v : Mv \neq 0} \mathop{\mathbb{E}}_{u} \left[(-1)^{u^\intercal M v} \right] \right) = \frac{2\sqrt{n} - r}{2\sqrt{n}} + 0 = 2^{-r}.$$

In addition, $\Pr_M \left[\text{rank of } M \leqslant 9\sqrt{n}/20 \right] \leqslant 2^{-9n/10}$. Indeed, the number of matrices in $\mathbb{F}_2^{\sqrt{n} \times \sqrt{n}}$ of rank at most k is at most

$$\binom{2\sqrt{n}}{k} \cdot \left(2^k\right)^{\sqrt{n}} \leqslant 2^{2k\sqrt{n}}.$$

Therefore, using the law of total expectation, we have that

$$\mathop{\mathbb{E}}_{M}\left[\left|\mathop{\mathbb{E}}_{(u,v)}\left[(-1)^{u^\intercal Mv}\right]\right|^k\right] \leqslant \Pr_{M}\left[\text{rank of } M \leqslant 9\sqrt{n}/20\right] + 2^{-9k\sqrt{n}/20} \leqslant 2\cdot 2^{-9k\sqrt{n}/20},$$

where the last inequality followed from the fact that $k \leqslant \sqrt{n}$. □

Proof of Theorem 11. Let c be the number of bits communicated by Alice. We show that $c > n/10$. Define $Z_M(u,v) = 1$ if Bob correctly computes $u^\intercal Mv$ and $Z_M(u,v) = -1$ otherwise. By the definition of $Z_M(u,v)$ and the lower bound on the probability of Bob's computation being correct, we have that $\mathbb{E}_{M,u,v}\left[Z_M(u,v)\right] \geqslant 2\cdot 2^{-\sqrt{n}/8}$.

We note that it is without loss of generality that $\mathbb{E}_{u,v}\left[Z_M(u,v)\right] \geqslant 0$ for every $M \in \mathbb{F}_2^{\sqrt{n}\times\sqrt{n}}$. This is because Alice on input M can send an extra bit indicating whether $\mathbb{E}_{u,v}\left[Z_M(u,v)\right] < 0$ and Bob will flip his output accordingly.

We now use the given protocol to design a protocol for a new communication game: Suppose that Alice gets a uniformly random $M \in \mathbb{F}_2^{\sqrt{n}\times\sqrt{n}}$ as input and Bob receives $\sqrt{n}$ uniform pairs $(u_1,v_1),\ldots,(u_{\sqrt{n}},v_{\sqrt{n}}) \in \mathbb{F}_2^{\sqrt{n}} \times \mathbb{F}_2^{\sqrt{n}}$ as input. We will use Lemma 13 with $k = \sqrt{n}$ to obtain the desired lower bound on c.

We claim that there is a communication protocol in which Alice communicates c bits and Bob computes $\sum_{i=1}^{\sqrt{n}} u_i^\intercal Mv_i$ such that

$$\Pr_{M,u_1,v_1,\ldots,u_{\sqrt{n}},v_{\sqrt{n}}}\left[\text{Bob computes } \sum_{i=1}^{\sqrt{n}} u_i^\intercal Mv_i \text{ correctly}\right] \geqslant \frac{1}{2} + \frac{2^{\sqrt{n}-1}}{2^{n/8}}. \tag{7}$$

Alice's message is same as before, and Bob computes each of $u_i^\intercal Mv_i$ separately and outputs the sum modulo 2. We now prove Eq. (7). For a fixed M, the probability that Bob correctly computes $\sum_{i=1}^{\sqrt{n}} u_i^\intercal Mv_i$ is $\frac{1}{2}\left(1 + \left(\mathbb{E}_{u,v}\left[Z_M(u,v)\right]\right)^{\sqrt{n}}\right)$. Therefore the overall probability that Bob correctly computes $\sum_{i=1}^{\sqrt{n}} u_i^\intercal Mv_i$ is at least

$$\frac{1}{2}\left(1 + \frac{\sum_M \left(\mathbb{E}_{u,v}\left[Z_M(u,v)\right]\right)^{\sqrt{n}}}{2^n}\right) \geqslant \frac{1}{2}\left(1 + \left(\mathop{\mathbb{E}}_{M,u,v}\left[Z_M(u,v)\right]\right)^{\sqrt{n}}\right) \geqslant \frac{1}{2} + \frac{2^{\sqrt{n}-1}}{2^{n/8}},$$

where the first inequality follows from convexity of the function $f(x) = x^k$ with $k = \sqrt{n}$. Applying Lemma 13 with $k = \sqrt{n}$ implies that $c > n/10$, which completes the proof of the theorem. □

4.2 Proof of Theorem 3

If $n < 36$, the theorem is vacuously true as $c \leqslant 1/36$. For the rest of the argument we will assume that $n \geqslant 36$. We prove a lower bound on the query time t against deterministic data structures with space s and word size w. Suppose that the input data M and query $uv^\intercal$ is given by choosing

M, u, v uniformly and independently at random, and the query algorithm is guaranteed to satisfy

$$\Pr_{M,u,v}[\text{query algorithm computes } u^\intercal M v \text{ correctly}] \geqslant \frac{1}{2} + 2^{-\sqrt{n}/16}.$$

By Yao's minmax principle, this will imply a lower bound on randomized data structures.

We first modify the given data structure to ensure that for every $M \in \mathbb{F}_2^{\sqrt{n}\times\sqrt{n}}$, the probability that $u^\intercal M v$ is correctly computed is at least $1/2$. Assume that we have a data structure with query time t', space s' and word size w. The modified data structure stores an extra bit indicating whether the $\Pr_{u,v}[\text{query algorithm computes } u^\intercal M v \text{ correctly}]$ is less than $1/2$ or not for a given M. The query algorithm is the same as before, but accesses this extra bit to flip the output if it is set to 1. Clearly, the new data structure has query time $t = t' + 1$, space $s = s' + 1$ and word size w. Moreover, under this modification, we have

- $\Pr_{M,u,v}[\text{query algorithm computes } u^\intercal M v \text{ correctly}] \geqslant 1/2 + 2^{-\sqrt{n}/16}$.
- $\Pr_{u,v}[\text{query algorithm computes } u^\intercal M v \text{ correctly}] \geqslant 1/2$ for every M.

In the rest of the proof, we work with this modification and show that $t \geqslant \Omega\left(\min\left\{\frac{n}{\beta}, \frac{\sqrt{n}}{\log \frac{s\beta}{n}}\right\}\right)$, where $\beta = 2(w + \log sw/n)$. Observe that $\beta \leqslant n/256$; otherwise the lower bound is vacuous.

Assume by contradiction that $t \leqslant \min\left\{\frac{n}{256\beta}, \frac{\sqrt{n}}{256\log \frac{s\beta}{n}}\right\}$. Define $Z_M(u,v) = 1$ if the query algorithm correctly computes $u^\intercal M v$, and -1 otherwise. We have

$$\mathop{\mathbb{E}}_{M,u,v}[Z_M(u,v)] = 2 \cdot \Pr_{M,u,v}[\text{query algorithm computes } u^\intercal M v \text{ correctly}] - 1 \geqslant 2 \cdot 2^{-\sqrt{n}/16}. \quad (8)$$

Note that $\mathbb{E}_{M,u,v}[Z_M(u,v)]$ captures the *advantage* or *bias* of the data structure - it is much more convenient to work with the advantage than the probability of the query algorithm being correct.

The following lemma, a variant of cell sampling, guarantees the existence of a small subset S of cells such that a large number of queries Q' can be computed by only accessing S, and $\mathbb{E}_{u,v}[Z_M(u,v) \mid uv^\intercal \in Q'] \approx \mathbb{E}_{u,v}[Z_M(u,v)]$.

Lemma 14. *Let $M \in \mathbb{F}_2^{\sqrt{n}\times\sqrt{n}}$. Define $Q_1 = \{uv^\intercal \mid Z_M(u,v) = 1\}$ and $Q_2 = \{uv^\intercal \mid Z_M(u,v) = -1\}$. If $t \leqslant \min\left\{\frac{n}{256\beta}, \frac{\sqrt{n}}{256\log \frac{s\beta}{n}}\right\}$, then there exits a subset of cells S, and subsets $Q_1' \subseteq Q_1$ and $Q_2' \subseteq Q_2$ such that*

1. $|S| = \left\lceil \frac{n}{128\beta} \right\rceil$,
2. $\Pr_{u,v}\left[uv^\intercal \in Q_1'\right] - \Pr_{u,v}\left[uv^\intercal \in Q_2'\right] \geqslant \mathbb{E}_{u,v}[Z_M(u,v)] \cdot 2^{-\sqrt{n}/16}$,
3. $Q_1' \cup Q_2'$ *is the set of all queries computed by accessing cells only in* S.

We move on to the final step of the proof of Theorem 3. What is left is to design a one-way protocol using the sets guaranteed by Lemma 14. The protocol is described in Figure 1. We will show the validity of this protocol by showing that both Alice and Bob know the subset Q' of

Input: Alice's input is M and Bob's input is (u, v)
Output: Alice communicates a deterministic message and Bob computes $u^\intercal M v$.

1 Let $Q_1 = \{uv^\intercal \mid Z_M(u,v) = 1\}$ and $Q_2 = \{uv^\intercal \mid Z_M(u,v) = -1\}$;
2 Apply Lemma 14 with Q_1, Q_2 to obtain a subset of cells S and subsets $Q_1' \subseteq Q_1$ and $Q_2' \subseteq Q_2$;
3 Let $b \in \{0,1\}$ be such that $\Pr_{u,v}[u^\intercal M v = b \mid uv^\intercal \notin Q'] \geqslant \Pr_{u,v}[u^\intercal M v = 1-b \mid uv^\intercal \notin Q']$, where $Q' = Q_1' \cup Q_2'$;
4 Alice communicates b followed by locations and contents of S;
5 **if** $uv^\intercal \in Q'$ **then** Bob uses the query algorithm to compute $u^\intercal M v$;
6 **else** Bob outputs b;

Figure 1: One-way protocol on inputs M, (u, v) computing $u^\intercal M v$.

queries. Since Alice's input is M, she knows the contents of all the cells, which gives S. With regard to knowing Q', the locations and contents of cells in S suffice. This is because the query algorithm can be simulated on all queries to check if any cell outside of S is being accessed. We are proving Theorem 3 by contradicting Theorem 11, which is achieved by the following.

Lemma 15. *The protocol in Figure 1 has the following guarantees (a) Alice communicates fewer than* $n/10$ *bits, and (b)* $\Pr_{M,u,v}[$*Bob computes* $u^\intercal M v$ *correctly*$] \geqslant 1/2 + 1/2^{\sqrt{n}/8}$.

Now, we need to prove Lemmas 14 and 15 to complete the proof of Theorem 3.

Proof of Lemma 14. Let S be a uniformly random subset of the cells of size $|S| = \left\lceil \frac{n}{128\beta} \right\rceil$. Define $D(u,v,S) = Z_M(u,v)$ if the query algorithm only accesses cells in S to compute $u^\intercal M v$; otherwise $D(u,v,S) = 0$. By linearity of expectation,

$$\mathop{\mathbb{E}}_{u,v,S}[D(u,v,S)] = \mathop{\mathbb{E}}_{u,v}[Z_M(u,v)] \cdot \frac{\binom{s-t}{|S|-t}}{\binom{s}{|S|}} = \mathop{\mathbb{E}}_{u,v}[Z_M(u,v)] \cdot \frac{|S| \cdot (|S|-1) \cdots (|S|-t+1)}{s \cdot (s-1) \cdots (s-t+1)}$$
$$\geqslant \mathop{\mathbb{E}}_{u,v}[Z_M(u,v)] \cdot \left(\frac{|S|-t}{s}\right)^t.$$

Recall that $|S| \geqslant \frac{n}{128\beta}$ and $t \leqslant \frac{n}{256\beta}$. Moreover, $\beta = 2(w + \log sw/n) \geqslant 2$. This implies that

$$\left(\frac{|S|-t}{s}\right)^t \geqslant 2^{-t \cdot \log \frac{256s\beta}{n}} \geqslant 2^{-16t \cdot \log \frac{s\beta}{n}}.$$

So we get $\mathbb{E}_{u,v,S}[D(u,v,S)] \geqslant \mathbb{E}_{u,v}[Z_M(u,v)] \cdot 2^{-16 \cdot t \cdot \log \frac{s\beta}{n}}$. Therefore, there exists an S such that

$$\mathop{\mathbb{E}}_{u,v}[D(u,v,S)] \geqslant \mathop{\mathbb{E}}_{u,v}[Z_M(u,v)] \cdot 2^{-16 \cdot t \cdot \log \frac{s\beta}{n}} \geqslant \mathop{\mathbb{E}}_{u,v}[Z_M(u,v)] \cdot 2^{-\sqrt{n}/16},$$

where the last inequality follows from the fact that $16 \cdot t \cdot \log \frac{s\beta}{n} \leqslant \sqrt{n}/16$. Setting

$$Q_1' = \{uv^\intercal \in Q_1 \mid D(u, v, S) = 1\} \text{ and } Q_2' = \{uv^\intercal \in Q_2 \mid D(u, v, S) = -1\}$$

completes the proof of the lemma. □

Proof of Lemma 15. We first prove part (a). Recall that $\beta = 2\left(w + \log \frac{sw}{n}\right)$. Let c be the number of bits communicated by Alice. By Proposition 12 and the definition of β,

$$\begin{aligned} c &\leqslant 1 + \left\lceil \frac{n}{128\beta} \right\rceil \cdot w + \left\lceil \frac{n}{128\beta} \right\rceil \cdot \log \frac{128e \cdot s\beta}{n} \\ &= 1 + \left\lceil \frac{n}{128\beta} \right\rceil \cdot \left(w + \log \frac{s\beta}{n} \right) + \left\lceil \frac{n}{128\beta} \right\rceil \cdot \log 128e. \end{aligned}$$

Since $\beta \geqslant 2w$, $\beta \geqslant 2\log \frac{s}{n}$ and $\beta \geqslant \log \beta$, we get that $w + \log \frac{s\beta}{n} \leqslant 2\beta$. Moreover, using the fact that $\left\lceil \frac{n}{128\beta} \right\rceil \leqslant \frac{n}{128\beta} + 1$, $\beta \geqslant 2$ and $\beta \leqslant n/256$, we can say that

$$\begin{aligned} c &\leqslant 1 + \frac{2n}{128} + 2\beta + \frac{n \log 128e}{128\beta} + \log 128e \\ &\leqslant 1 + \frac{2n}{128} + \frac{4.5n}{128(\beta/2)} + \frac{n}{128} + \log 128e \leqslant 10 + \frac{7.5n}{128} < \frac{n}{10}, \end{aligned}$$

where the last inequality follows from $n \geqslant 36$.

We now prove part (b) of the claim. Define $Z'_M(u, v) = 1$ if the Bob correctly computes $u^\intercal M v$ and $Z'_M(u, v) = -1$ otherwise. The probability with which Bob correctly computes $u^\intercal M v$ is given by $\left(1 + \mathbb{E}_{M,u,v}\left[Z'_M(u, v)\right]\right)/2$. We will show that $\mathbb{E}_{M,u,v}\left[Z'_M(u, v)\right] \geqslant 2 \cdot 2^{-\sqrt{n}/8}$, which will imply that the probability of being correct is at least $1/2 + 2^{-\sqrt{n}/8}$.

Let Q_1, Q_2, Q_1', Q_2', and Q' be as defined in the protocol in Figure 1. We first establish some properties about these sets. We know that $\Pr_{u,v}[uv^\intercal \in Q_1] - \Pr_{u,v}[uv^\intercal \in Q_2] = \mathbb{E}_{u,v}\left[Z_M(u, v)\right]$. Moreover, the application of Lemma 14 in the protocol is valid since $t \leqslant \frac{n}{256\alpha}$, and hence

$$\Pr_{u,v}\left[uv^\intercal \in Q_1'\right] - \Pr_{u,v}\left[uv^\intercal \in Q_2'\right] \geqslant \mathop{\mathbb{E}}_{u,v}\left[Z_M(u, v)\right] \cdot 2^{-\sqrt{n}/16}. \tag{9}$$

Since Bob can simulate the query algorithm on Q' by accessing only S, which is guaranteed by Lemma 14, we have

$$\begin{aligned} \mathop{\mathbb{E}}_{u,v}\left[Z'_M(u, v)\right] &= \Pr_{u,v}\left[uv^\intercal \in Q'\right] \cdot \left(\Pr_{u,v}\left[uv^\intercal \in Q_1' \mid uv^\intercal \in Q'\right] - \Pr_{u,v}\left[uv^\intercal \in Q_2' \mid uv^\intercal \in Q'\right] \right) \\ &\quad + \Pr_{u,v}\left[uv^\intercal \notin Q'\right] \left(\Pr_{u,v}\left[u^\intercal M v = b \mid uv^\intercal \notin Q'\right] - \Pr_{u,v}\left[u^\intercal M v = 1 - b \mid uv^\intercal \notin Q'\right] \right) \\ &\geqslant \left(\Pr_{u,v}\left[uv^\intercal \in Q_1'\right] - \Pr_{u,v}\left[uv^\intercal \in Q_2'\right] \right) \geqslant \mathop{\mathbb{E}}_{u,v}\left[Z_M(u, v)\right] \cdot 2^{-\sqrt{n}/16}, \end{aligned}$$

where the first inequality follows from the choice of b and the second inequality used Eq. (9).

To conclude,

$$\begin{aligned}\mathop{\mathbb{E}}_{M,u,v}\left[Z'_M(u,v)\right] = \mathop{\mathbb{E}}_{M}\left[\mathop{\mathbb{E}}_{u,v}\left[Z'_M(u,v)\right]\right] &\geqslant \mathop{\mathbb{E}}_{M}\left[\mathop{\mathbb{E}}_{u,v}\left[Z_M(u,v)\right]\cdot 2^{-\sqrt{n}/16}\right] \\ &= \mathop{\mathbb{E}}_{M}\left[\mathop{\mathbb{E}}_{u,v}\left[Z_M(u,v)\right]\right]\cdot 2^{-\sqrt{n}/16} \\ &= \mathop{\mathbb{E}}_{M,u,v}\left[Z_M(u,v)\right]\cdot 2^{-\sqrt{n}/16} \geqslant 2\cdot 2^{-\sqrt{n}/8},\end{aligned}$$

where the last inequality follows from Eq. (8). □

Acknowledgments

We thank Paul Beame, Sajin Koroth, Pavel Hrubeš, Pavel Pudlák, Anup Rao, Makrand Sinha, Amir Yehudayoff and Sergey Yekhanin for useful discussions. Special thanks to Paul, Anup, Makrand and Amir for the encouragement to write up these results.

References

[1] Pankaj K. Agarwal. Geometric Range searching. In *CRC Handbook of Computational Geometry*. CRC, 1997.

[2] Miklós Ajtai. A Lower Bound for Finding Predecessors in Yao's Cell Probe Model. *Combinatorica*, 8(3), 1988.

[3] Josh Alman and Ryan Williams. Probabilistic Rank and Matrix Rigidity. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 641–652. ACM, 2017.

[4] Noga Alon and Gil Cohen. On Rigid Matrices and U-Polynomials. *Computational Complexity*, 24(4):851–879, Dec 2015.

[5] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic Approximation Algorithms for the Nearest Codeword Problem. In *APPROX '09 / RANDOM '09*, pages 339–351, 2009.

[6] V. L. Artazarov, E. A. Dinic, D. A. Kronrod, and I. A. Faradzev. On Economical Construction of the Transitive Closure of a Directed Graph. *Soviet Math. Dokl.*, 11:1209–1210, 1970.

[7] Joshua Brody and Kasper Green Larsen. Adapt or Die: Polynomial Lower Bounds for Non-Adaptive Dynamic Data Structures. *Theory of Computing*, 11(19):471–489, 2015.

[8] Diptarka Chakraborty, Lior Kamma, and Kasper Green Larsen. Tight Cell Probe Bounds for Succinct Boolean Matrix-Vector Multiplication. In *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1297–1306, 2018.

[9] Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. Simulation Beats Richness: New Data-Structure Lower Bounds. In *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1013–1020, 2018.

[10] Raphaël Clifford, Allan Grønlund, and Kasper Green Larsen. New Unconditional Hardness Results for Dynamic and Online Problems. In *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1089–1107, 2015.

[11] Henry Corrigan-Gibbs and Dmitry Kogan. The Function-Inversion Problem: Barriers and Opportunities. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:182, 2018.

[12] Zeev Dvir and Benjamin Edelman. Matrix Rigidity and the Croot-Lev-Pach Lemma. *arXiv preprint arXiv:1708.01646*, 2017.

[13] Zeev Dvir, Alexander Golovnev, and Omri Weinstein. Static Data Structure Lower Bounds Imply Rigidity. In *Proc. 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 967–978, 2019. see `https://arxiv.org/abs/1811.02725v3`.

[14] Zeev Dvir and Allen Liu. Fourier and Circulant Matrices Are Not Rigid. In *34th Computational Complexity Conference (CCC 2019)*, volume 137, pages 17:1–17:23, 2019.

[15] Michael Fredman and Michael Saks. The Cell Probe Complexity of Dynamic Data Structures. In *ACM Symposium on Theory of Computing (STOC)*, 1989.

[16] Joel Friedman. A Note on Matrix Rigidity. *Combinatorica*, 13(2):235–239, Jun 1993.

[17] Anna Gál and Peter Bro Miltersen. The Cell Probe Complexity of Succinct Data Structures. *Theoretical Computer Science*, 379(3):405–417, 2007.

[18] Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and Strengthening Hardness for Dynamic Problems via the Online Matrix-Vector Multiplication Conjecture. In *Proc. 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 21–30, 2015.

[19] Stasys Jukna and Georg Schnitger. Min-Rank Conjecture for Log-Depth Circuits. *Journal of Computer and System Sciences*, 77(6):1023–1038, 2011.

[20] Kasper Green Larsen. Higher Cell Probe Lower Bounds for Evaluating Polynomials. In *FOCS*, pages 293–301. IEEE Computer Society, 2012.

[21] Kasper Green Larsen. The Cell Probe Complexity of Dynamic Range Counting. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 85–94, 2012.

[22] Kasper Green Larsen, Omri Weinstein, and Huacheng Yu. Crossing the Logarithmic Barrier for Dynamic Boolean Data Structure Lower Bounds. In *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 978–989, 2018.

[23] Kasper Green Larsen and Ryan Williams. Faster Online Matrix-Vector Multiplication. In *Proc. 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2182–2189, 2017.

[24] Satyanarayana V. Lokam. Complexity Lower Bounds Using Linear Algebra. *Foundations and Trends® in Theoretical Computer Science*, 4(1–2):1–155, 2009.

[25] Peter Bro Miltersen. Lower Bounds for Union-Split-Find Related Problems on Random Access Machines. In *Proceedings of the 26th Annual Symposium on the Theory of Computing*, pages 625–634, New York, May 1994. ACM Press.

[26] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On Data Structures and Asymmetric Communication Complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

[27] Rina Panigrahy, Kunal Talwar, and Udi Wieder. Lower Bounds on Near Neighbor Search via Metric Expansion. In *Proc. 51st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 805–814, 2010.

[28] Mihai Pătraşcu. Unifying the Landscape of Cell-Probe Lower Bounds. *SIAM Journal on Computing*, 40(3):827–847, 2011. See also FOCS'08, arXiv:1010.3783.

[29] Mihai Pătraşcu and Erik D. Demaine. Logarithmic Lower Bounds in the Cell-Probe Model. *SIAM Journal on Computing*, 35(4):932–963, 2006. See also STOC'04, SODA'04.

[30] Mihai Pătraşcu and Mikkel Thorup. Higher Lower Bounds for Near-Neighbor and Further Rich Problems. *SIAM Journal on Computing*, 39(2):730–741, 2010. See also FOCS'06.

[31] Pavel Pudlák, Vojtech Rödl, and Jirí Sgall. Boolean Circuits, Tensor Ranks, and Communication Complexity. *SIAM Journal on Computing*, 26(3):605–633, 1997.

[32] Anup Rao and Amir Yehudayoff. *Communication Complexity*. preprint at `https://homes.cs.washington.edu/~anuprao/pubs/book.pdf`.

[33] M.A. Shokrollahi, D.A. Spielman, and V. Stemann. A Remark on Matrix Rigidity. *Information Processing Letters*, 64(6):283 – 285, 1997.

[34] Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A Survey of Recent Results and Open Questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010.

[35] Leslie G. Valiant. Graph-Theoretic Arguments in Low-Level Complexity. In Jozef Gruska, editor, *Mathematical Foundations of Computer Science*, pages 162–176, 1977.

[36] 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.

[37] Emanuele Viola. On the Power of Small-Depth Computation. *Foundations and Trends® in Theoretical Computer Science*, 5(1):1–72, 2009.

[38] Emanuele Viola. Lower Bounds for Data Structures with Space Close to Maximum Imply Circuit Lower Bounds. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:186, 2018.

[39] Henning Wunderlich. On a Theorem of Razborov. *Computational Complexity*, 21(3):431–477, 2012.

[40] Andrew Yao. Should Tables be Sorted? *JACM: Journal of the ACM*, 28, 1981.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il