Electronic Colloquium on Computational Complexity, Report No. 188 (2018)

Static Data Structure Lower Bounds Imply Rigidity

Zeev Dvir* Alexander Golovnev† Omri Weinstein‡

Abstract

We show that static data structure lower bounds in the group (linear) model imply semi-explicit lower bounds on matrix rigidity. In particular, we prove that an explicit lower bound of $t \geq \omega(\log^2 n)$ on the cell-probe complexity of linear data structures in the group model, even against arbitrarily small linear space ($s = (1+\varepsilon)n$), would already imply a semi-explicit ($\mathbf{P^{NP}}$) construction of rigid matrices with significantly better parameters than the current state of art (Alon, Panigrahy and Yekhanin, 2009). Our results further assert that polynomial ($t \geq n^{\delta}$) data structure lower bounds against near-optimal space, would imply super-linear circuit lower bounds for log-depth linear circuits (a four-decade open question). In the succinct space regime ($s = n + o(n)$), we show that any improvement on current cell-probe lower bounds in the linear model would also imply new rigidity bounds. Our results rely on a new connection between the "inner" and "outer" dimensions of a matrix (Paturi and Pudlák, 2006), and on a new reduction from worst-case to average-case rigidity, which is of independent interest.

*Department of Computer Science and Department of Mathematics, Princeton University. Email: zeev.dvir@gmail.com. Research supported by NSF CAREER award DMS-1451191 and NSF grant CCF-1523816.

†Harvard University. Email: alexgolovnev@gmail.com.

‡Columbia University. Email: omri@cs.columbia.edu.

ISSN 1433-8092

1 Introduction

Proving lower bounds on the operational time of data structures has been a long and active research endeavor for several decades. In the static setting, the goal is to *preprocess* a database of n elements into minimum space s ($\geq n$), so that queries $q \in \mathcal{Q}$ on the input database can be answered quickly, in query time t (where the typical and realistic setting is $|\mathcal{Q}| = \text{poly}(n)$). The two naïve solutions to any such problem is to either precompute and store the answers to all queries in advance, which has optimal query time but prohibitive space ($s = |\mathcal{Q}|$), or to store the raw database using optimal space ($s \sim n$) at the price of trivial query time ($t = n$). The obvious question is whether the underlying problem admits a better time-space trade-off. Static data structure lower bounds aim to answer this question by proving unconditional lower bounds on this trade-off.

The most compelling model for proving such lower bounds is the "cell-probe" model [Yao81], in which a data structure is simply viewed as a table of s memory cells (w-bit words) and query time is measured only by the *number t of memory accesses* (I/Os), whereas all computations on "probed" memory cells are free of charge. This nonuniform[1] model of computation renders time-space trade-offs as purely information-theoretic question and thereby extremely powerful. Unfortunately, the abstraction of this model also comes at a price: While a rather straight-forward counting argument [Mil93] shows that *most* static data structure problems with $m := |\mathcal{Q}|$ queries indeed require either $t \geq n^{0.99}$ time or $s \geq m^{0.99}$ space (i.e., the naïve solutions are essentially optimal), the highest *explicit* cell-probe lower bound known to date is

$$t \geq \Omega\left(\frac{\log(m/n)}{\log(s/n)}\right). \tag{1}$$

In the interesting and realistic regime of polynomially many queries ($m = n^{O(1)}$), this yields a $t \gtrsim \log n$ lower bound on the query time of *linear space* ($s = O(n)$) data structures for several natural problems, such as polynomial-evaluation, nearest-neighbor search and 2D range counting to mention a few [Sie04, Păt08, PTW10, Lar12]. Proving an $\omega(\log n)$ cell-probe lower bound on *any* explicit static problem in the linear space regime, is a major open problem, and the trade-off in (1) remains the highest static cell-probe lower bound known to date, even for *nonadaptive* data structures (This is in sharp contrast to *dynamic* data structures, where the restriction to nonadaptivity enables *polynomial* cell-probe lower bounds [BL15]).

In an effort to circumvent the difficulty of proving lower bounds in the cell-probe model, several restricted models of data structures have been studied over the years, e.g., the pointer-machine and word-RAMs [vEB90] as well as (stronger) algebraic models, most notably, the *group model* [Fre81, Cha90, Păt07]. Since many important data structure problems involve *linear queries* over the database (e.g., orthogonal range counting, partial sums, dictionaries, matrix-vector multiplication and polynomial evaluation to mention a few), it is natural to restrict the data structure to use only *linear* operations as well. More formally, a static *linear* data structure problem over a field $\mathbb{F}$ and input database $x \in \mathbb{F}^n$, is defined by an $m \times n$ matrix (i.e., a linear map) $M \in \mathbb{F}^{m \times n}$. The m queries are the rows M_i of M, and the answer to the ith query is $\langle M_i, x \rangle = (Mx)_i \in \mathbb{F}$. An (s,t)-*linear data structure* for M is allowed to store s arbitrary field elements in memory $P(x) \in \mathbb{F}^s$, and must compute each query $(Mx)_i$ as a *t-sparse linear combination* of its memory state (we assume the

[1] Indeed, a nonadaptive cell-probe data structure is essentially equivalent to an m-output depth-2 circuit with *arbitrary gates*, "width" s, and a *bounded* top fan-in t, see [JS11, BL15] and Section 2.5.

word-size satisfies $w \geq \log |\mathbb{F}|$, see Section 2.1 for the complete details). This model is a special case of the *static group model*, except here the group (field) is fixed in advance.[2]

While the restriction to the group model has been fruitful for proving strong lower bounds on *dynamic* data structures[3] ([Aga04, Pătˇ07, Lar14]), the static group model resisted this restriction as well, and (1) remains the highest static lower bound even against nonadaptive linear data structures.

This paper shows that this barrier is no coincidence. We study linear data structures and show that proving super-logarithmic static lower bounds, even against nonadaptive linear data structures with *arbitrarily* small linear space $s = (1+\varepsilon)n$, implies semi-explicit lower bounds on *matrix rigidity*. Before stating our main results, we take a moment to introduce the notion of rigidity.

Matrix rigidity The notion of matrix rigidity was introduced by Valiant [Val77] as a possible approach for proving circuit lower bounds. We say that a matrix $A \in \mathbb{F}^{m \times n}$ is (r, d)-row rigid, if decreasing the rank of A below r, requires modifying at least d entries in *some row* of A. In other words, for any r-dimensional subspace U of $\mathbb{F}^n$, there exists a row in A that is d-far (in Hamming distance) from U. We discuss a stronger notion of rigidity called *'global rigidity'* later in the paper (requiring many rows of A to be far from U) and prove a general reduction from one to the other (see Theorem 3 below) which may be of independent interest.

The best known bound on matrix rigidity of square matrices (for any rank parameter r) is $\Omega(\frac{n}{r} \log \frac{n}{r})$ [Fri93, PR94, SSS97, Lok09][4]. Although matrix rigidity has attracted a lot of attention, this bound remains the best known for more than two decades.

Matrix rigidity is also studied for rectangular matrices (introduced in [APY09]), where we allow the number of rows m to be larger than the number of columns. One can thus fix the parameters r (typically $r \approx \varepsilon n$) and d (the sparsity) and try to minimize the number of rows in A. One can easily show that a random square matrix is highly rigid (say with r and d both close to n), but the best explicit constructions of an $m \times n$ matrix which is $(\varepsilon n, d)$-row-rigid requires $m = n \cdot 2^d$ [APY09, SY11]. This bound (and the related lower bound for square matrices which is even older) represent, to many experts in the field, a real barrier, and any improvement to it is likely to require substantially new ideas. Our results below show that this barrier can be surpassed if one can improve the currently known lower bounds on static data structures by a slightly super-logarithmic factor.

1.1 Our results

Our first main result is the following (see Theorem 7 for a formal statement):

[2] In the general (oblivious) group model, the input database consists of n elements from a black-box (commutative) group, the data structure can only store and manipulate group elements through black-box group operations, and query-time (t) is measured by the number of algebraic operations (see e.g. [Aga04, Pătˇ07] for further details).

[3] In the dynamic setting, the data structure needs to maintain an *online* sequence of operations while minimizing the number of memory accesses for update and query operations. In the group (linear) model, these constraints are essentially equivalent to a decomposition of a matrix $M = AB$ where *both* A and B are sparse. In contrast, static lower bounds only require A to be sparse, hence intuitively such decomposition is much harder to rule out.

[4] Goldreich and Tal [GT16] also give a "semi-explicit" construction of rigid matrices which uses $O(n)$ bits of randomness. This construction has (global) rigidity $\Omega(\frac{n^2}{r^2 \log n})$ for any $r \geq \sqrt{n}$, which improves on the classical bound for $r = o(\frac{n}{\log n \log \log n})$. In particular, since the required number of random bits is only linear, this construction is in $\mathbf{E}^{\mathbf{NP}}$.

Theorem 1 (Main Theorem, Informal)**.** *A data structure lower bound of $t \geq \log^c n$ in the group (linear) model for computing a linear map $M \in \mathbb{F}^{m \times n}$, even against data structures with arbitrarily small linear space $s = (1+\varepsilon)n$, yields an $(\varepsilon n', d)$-row-rigid matrix $M' \in \mathbb{F}^{m \times n'}$ with $\varepsilon n' \geq d \geq \Omega(\log^{c-1} n)$. Moreover, if M is explicit, then $M' \in \mathbf{P^{NP}}$.*

The premise of Theorem 1 would imply a (semi-explicit) construction of an $m \times n$ matrix which is $d \sim \log^{c-1}(m/n)$-rigid (i.e., requires modifying at least d entries in some row to decrease the rank below, say, $n/4$). In comparison, the aforementioned best known explicit constructions only yield an $\Omega(\log(m/n))$-rigid matrix [APY09, SY11], which is only $\Omega(\log n)$ when $m = \text{poly}(n)$. In particular, Theorem 1 asserts that proving a $t \geq \omega(\log^2 n)$ data structure lower bound against arbitrarily small linear space, would already yield an asymptotic improvement on (rectangular) rigid matrix construction.[5]

Theorem 1 indicates a "threshold" in data structure lower bounds, since for *succinct* data structures (which are constrained to use only $s = n + o(n)$ space), polynomial lower bounds ($t \geq n^{\varepsilon}$) are known on the query time (e.g., [GM07, BL13]), even in the general *cell-probe* model.

Our second main result concerns implications of data structure lower bounds on *square* matrix rigidity (see Theorem 7, item 3 for a formal statement):

Theorem 2 (Implications to Square Rigidity, Informal)**.** *For any $\delta > 0$, a data structure lower bound of $t \geq \log^{3+\delta} n$ in the group (linear) model, for computing a linear map $M \in \mathbb{F}^{m \times n}$, even against arbitrarily small linear space $s = (1+\varepsilon)n$, yields a* square *matrix $M' \in \mathbb{F}^{n' \times n'}$ which is $\left(r, \omega(\frac{n'}{r} \log \frac{n'}{r})\right)$-rigid, for some $r = o(n)$. Moreover, if M is explicit, then $M' \in \mathbf{P^{NP}}$.*

Since the highest rigidity bound known to date for square matrices (and any rank parameter r) is $\Omega(\frac{n'}{r} \log \frac{n'}{r})$ [Fri93], the premise of Theorem 2 would imply an asymptotic improvement over state-of-art lower bounds (the precise factor is given in the formal statement, see Theorem 7).

Our main result has further significant implications to other time-space regimes. In the succinct space regime, we show that any asymptotic improvement on the current best cell-probe lower bounds mentioned above, would yield improved rigidity bounds for near-square matrices, and vice versa. In particular, a corollary of this connection yields a logarithmic improvement on succinct lower bounds (in the group model): We exhibit an (explicit) data structure lower bound of $t \cdot r \geq \Omega(n \log(n/r))$ for linear data structures using space $s = n + r$, which is a logarithmic-factor improvement on the aforementioned bounds of [GM07, BL13] for a problem with linear number of queries $m = O(n)$.

Finally, we show that 'holy-grail' *polynomial* ($t \geq n^{\delta}$) data structure lower bounds against near-trivial space ($s \sim m/\log\log m$), would imply superlinear circuit lower bounds (see Theorem 8). We discuss all of these implications in Section 4.

1.2 Technical Overview

It is not hard to see that a (nonadaptive) (s,t)-*linear* data structure for a linear problem $M \in \mathbb{F}^{m \times n}$ is nothing but a factorization of M as a product of two matrices $M = AB$, where A is a t-sparse $(m \times s)$ matrix (with $\leq t$ non-zeros in each row), and B is an arbitrary matrix with only s rows (see Section 2.1). As such, proving a lower bound on (s,t) linear data structures is equivalent to finding an (explicit) matrix $M \in \mathbb{F}^{m \times n}$ which *does not* admit such factorization, or equivalently,

[5] Although here we state Theorem 1 for the lowest probe complexity that is interesting, it actually gives a smooth trade-off: a lower bound on the linear data structure query time t implies rigidity $\frac{t}{\log n}$.

showing that M is "*sumset-evasive*", in the sense that the m rows of M (viewed as points $M_i \in \mathbb{F}^n$) are not contained in the t-span[6] of *any fixed* set of s points in $\mathbb{F}^n$ (see Section 2.3 below for the formal definition of (s,t)-sumset evasive sets).

In contrast, *matrix rigidity* is the problem of finding an (explicit) matrix $M \in \mathbb{F}^{m\times n}$ which cannot be factorized as the *sum* (rather than product) $M = A + B$ of a t-row-sparse matrix A plus a *low rank* matrix, say, $rk_{\mathbb{F}}(B) \leq r$. For brevity, unless otherwise stated, we say below that a matrix is (r,d)-rigid to mean that is is d-row-rigid, and that it is t-sparse to mean t-row-sparse.

We establish a new relationship between these two (seemingly disparate) factorizations. A key step in showing this relationship is to re-interpret the two factorization problems above as two (respective) "geometric" measures on the *column space* of M, i.e., viewing the matrix $M \in \mathbb{F}^{m\times n}$ as an n-dimensional subspace $V_M \subset \mathbb{F}^m$ spanned by its columns. Informally, the *inner dimension* of V_M is the maximal dimension $d_M \leq n$ of the *intersection* of V_M with any *t-sparse subspace*[7] A of the same dimension n (in other words, V_M has small inner dimension $d_M(t)$ if it has low-dimensional intersection with any n-dimensional *t-sparse* subspace, see Definition 2 below). The *outer dimension* of V_M is the minimal dimension $D_M \geq n$ of a t-sparse subspace A that *contains* V_M (Definition 3). We first prove the following characterization (Lemmas 4 and 2):

- M is strongly[8] (r,t)-rigid if an only if V_M has *small inner dimension* ($d_M(t) < n - r$).
- M is (s,t) sumset-evasive if and only if V_M has *large outer dimension* ($D_M(t) > s$).

(We note that the nontrivial direction of the first statement was already shown by [PP06] for a subtly different notion of inner/outer dimensions, we provide an alternative proof in Appendix C). In this terminology, proving that lower bounds on linear data structures imply lower bounds on (row) rigidity, is essentially equivalent to showing that *large outer dimension implies small inner dimension* (perhaps of a related explicit matrix). Indeed, our first main technical contribution is establishing the following relationship between these two measures on *submatrices* of M, which is the heart of the proof of Theorem 1.

Lemma 1 (Large outer dimension implies small inner dimension, Theorem 5, Informal)**.** *If $D_M(t) \geq (1+\varepsilon)n$, there exists an $m \times n'$ submatrix $M' \subseteq M$ for which $d_{M'}(t/\log n) \leq (1-\varepsilon)n'$.*

Indeed, by the characterization above, the last inequality implies that M' is $(\varepsilon n', t/\log n)$-rigid. The high level idea of the proof is a simple recursive procedure that, given a matrix M with high outer dimension $D_M(t)$, 'finds' a submatrix with low inner dimension. The heart of each iteration is as follows: If our current matrix (which is initially M itself) is rigid (i.e., has low inner dimension, which can be checked with an **NP** oracle), then we are done. Otherwise, the **NP** oracle together with the characterization above, gives us a sparse subspace V (of only n dimensions) that has large intersection with the column space of M. After a change of basis (of the column space) we can essentially, partition the columns of M into the part covered by V and the remaining columns. We then apply the same argument on the remaining columns. At each iteration we 'accumulate' an additional sparse V (whose dimension is small – merely the dimension of the residual space) and so, at the end, we must show that these can be pasted together to give a low-dimensional 'cover'

[6] I.e., the union of all t-dimensional subspaces generated by any *fixed* set $S \subset \mathbb{F}^n$ of size s. We borrow the term "sumset evasive" by analogy from additive combinatorics, but caution that this definition allows arbitrary *linear combinations* and not just sums.

[7] We say that a subspace $U \subseteq \mathbb{F}^m$ is *t-sparse* if it is the column-space of a t-row-sparse matrix

[8] A matrix M is strongly-rigid if it remains (row) rigid in *any* basis of its column-space V_M, see Definition 6.

of the column-space V of the original matrix M (i.e., a small space data structure for M). Thus, the final sparsity grows by a factor proportional to the number of iterations, which is logarithmic. This implies that the process must end prematurely, resulting in a (semi-) explicit rigid submatrix.

Square Matrix Rigidity (Theorem 2). One caveat of Theorem 1 is that it may produce a highly skewed (rectangular) $m \times n'$ matrix, where $m \gg n'$ (this is because we only recurse on the column-space of M but never on the row-space). While this already yields a significant improvement over current-best rectangular rigidity results (e.g., when $n' = \text{poly}\log(n)$), this argument does not seem to imply anything about rigidity for *square* matrices.

A trivial idea to turn the rectangular $(m \times n')$-row-rigid matrix M' produced by Lemma 1 into a square matrix while preserving rigidity (at the expense of decreasing the relative rank parameter), is to "stack" m/n' identical copies of M' side by side. Clearly, the rank of the resulting $m \times m$ matrix M'' remains unchanged (bounded by n', so the relative rank parameter may decrease significantly relative to m) [9], but on the other hand, the hope is that M'' remains $\Omega(m)$-row-rigid. Indeed, Lemma 1 guarantees that M' is (say) $(n'/10, n'/2)$-row-rigid, and therefore in order to decrease the rank of M'' below $(n'/10)$, one would need to decrease the rank of *each* of the m/n' blocks below $n'/10$, which requires modifying $\sim (m/n') \cdot (n'/2) = m/2$ row entries in total. The problem, of course, is that these rows may be *different* in each block, which completely dooms the argument. Note that this is a direct consequence of working with *row-rigidity*: If M' were *globally* rigid (i.e., at least 10% of its rows need to be modified in at least $\sim t$ entries in order to decrease the rank below $n'/10$), this simple trick would have gone through flawlessly.

In order to bypass this obstacle, we prove the following *black-box* reduction from row-rigidity to global-rigidity. Our reduction uses Locally Decodable Codes (LDCs) in a way reminiscent of the way LDCs are used in worst-case to average-case hardness amplification results [IW01].

Theorem 3 (Row to Global Rigidity)**.** *Let $E : \mathbb{F}^m \mapsto \mathbb{F}^{m'}$ be a* linear *q-query locally decodable code (LDC) against constant noise δ, and let $E(A) \in \mathbb{F}^{m' \times n}$ be the application of E to each column of $A \in \mathbb{F}^{m \times n}$. Then if A is (r,t)-row-rigid, then $E(A)$ is $(r, \delta t m'/q)$-globally rigid.*

To prove the theorem, we use the following crucial property of *linear* LDCs, originally observed by Goldreich et. al [GKST02]: If $E : \mathbb{F}^m \mapsto \mathbb{F}^{m'}$ is a q-query linear LDC (applied on the columns of $A \in \mathbb{F}^{m \times n}$), then for *any* subset S of at most $\delta m'$ rows of $E(A)$, and any row A_j of A, there exist q rows of $E(A)$ that lie *outside* S and span A_j. (See Section 3.2 for a formal argument). Now, suppose towards contradiction, that $E(A)$ is not (r,d)-globally rigid, for $d := t\delta m'/2q$. This means that there is some r-dimensional subspace $L \subset \mathbb{F}^n$ which is at most $(t\delta/2q)$-far (in Hamming distance) from an average row of $E(A)$, hence by a Markov argument, at most $\delta m'$ of the rows of $E(A)$ are $> (t/2q)$-far from L. Let $\mathcal{B}$ denote this set of rows. Since $|\mathcal{B}| \leq \delta m'$, the LDC property above asserts that *every row of A* is a linear combination of at most q rows in $E(A) \setminus \mathcal{B}$, each of which is $(t/2q)$-close to L by definition of $\mathcal{B}$. But this means that *every* row of A is at most $(t/2q) \cdot q = t/2$ far from L, which is a contradiction since A was assumed to be t-row-rigid (hence there must be at least *one* row that is t-far from L). The complete proof can be found in Theorem 6.

Since there are explicit linear $q = \log^{1+\varepsilon}(n)$-query LDCs with polynomial rate ($m' \approx m^{1/\varepsilon}$), Theorem 3 now completes the proof of Theorem 2 using the aforementioned program (stacking copies of M' next to each other), at the price of an extra logarithmic loss in sparsity. To the best of our knowledge, Theorem 3 establishes the first nontrivial relationship between rectangular

[9] The ratio between n' and m depends on the the postulated data structure lower bound on M, determining n'.

and square (row) rigidity, hence it may be of independent and general interest to rigidity theory (as the notion of row-rigidity is potentially much weaker than global rigidity). We also remark that Theorem 3 applies in the same way to reduce worst-case to average-case *data-structure* lower bounds for linear problems.

2 Setup and Preliminaries

2.1 Linear Data Structures (Static Group Model)

A *linear* data structure problem with $|\mathcal{Q}| = m$ queries over a field $\mathbb{F}$ and an input database of n elements is defined by an $m \times n$ matrix (i.e., a linear map) $V \in \mathbb{F}^{m \times n}$. The queries are the rows V_i of V, and for any input database $x \in \mathbb{F}^n$, the answer to the ith query is given by $\langle V_i, x \rangle = (Vx)_i \in \mathbb{F}$.

An (s, t) nonadaptive data structure $\mathcal{D}$ for the problem V in the cell-probe model is a pair $\mathcal{D} = (P, Q)$, where P is a *preprocessing* function $P : \mathbb{F}^n \mapsto \mathbb{F}^s$ that encodes the database $x \in \mathbb{F}^n$ into s memory *cells*, and a query algorithm $Q : \mathbb{F}^s \mapsto \mathbb{F}^m$ that correctly answers every query of V by *probing* at most t memory cells[10], i.e., such that $Q(P(x)) = (Vx)_i$, for every $x \in \mathbb{F}^n$ and every query $i \in [m]$.

$\mathcal{D}$ is a *linear data structure* for the (linear) problem V if both P and Q only compute linear functions over $\mathbb{F}$ .We observe that it suffices to require Q to be linear: if a linear problem V is solved by a data structure $\mathcal{D}$ with a linear *query* function Q, then $\mathcal{D}$ can be transformed into an equivalent data structure with the same parameters s and t where *both* P and Q are linear.

Proposition 1 (Lemma 2.5 [JS11], Ex. 13.7 [Juk12])**.** *Given an (s, t)-data structure $\mathcal{D}$ computing a linear transformation Vx for $V \in \mathbb{F}^{m \times n}, x \in \mathbb{F}^n$ with linear query function Q, one can efficiently construct an equivalent (s, t)-data structure where* both *the query function Q and the preprocessing function P are linear.*

2.2 Inner and Outer Dimensions

We state our main technical results in terms of Paturi-Pudlák dimensions [PP06, Lok09], and then show that they imply new connections between data structure lower bounds and matrix rigidity. While Paturi-Pudlák dimensions are defined w.r.t. column sparsity, for our applications we need to consider an analogous definition w.r.t. row sparsity (this difference is important in this context).

Definition 1 (Sparse subspaces)**.** *A matrix $M \in \mathbb{F}^{m \times n}$ is* t-globally sparse *if it has t non-zero elements, and M is* t-row sparse *if each of its* rows *has at most t non-zero entries. A subspace $V \subseteq \mathbb{F}^m$ is* t-sparse *if it is the* column space *of a t-row sparse matrix.*

Definition 2 (Inner dimension [PP06])**.** *Let $V \subseteq \mathbb{F}^m$ be a subspace, and t be a sparsity parameter. Then the* inner dimension $d_V(t)$ *of V is*

$$d_V(t) = \max_U \{\dim(V \cap U) \colon \dim(U) \leq \dim(V),\ U \text{ is } t\text{-sparse}\}.$$

Definition 3 (Outer dimension [PP06])**.** *Let $V \subseteq \mathbb{F}^m$ be a subspace, and t be a sparsity parameter. Then the* outer dimension $D_V(t)$ *of V is*

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

[10] the indices of memory cells are only function of the query index

By abuse of notation, for a matrix $M \in \mathbb{F}^{m \times n}$ we denote by $d_M(s)$ and $D_M(s)$ the inner and outer dimensions of the column space of M.

2.3 Sumset Evasive Sets

For an integer t and a set of points $S \subseteq \mathbb{F}^n$, tS denotes the t-span of S, i.e., the union of all t-sparse linear combinations of S :

$$tS := \{w_1 \cdot s_1 + \ldots + w_t \cdot s_t : \forall i, w_i \in \mathbb{F}, s_i \in S\}.^{11}$$

Definition 4 (Sumset evasive sets)**.** *For integers s and t we say that a set $M \subseteq \mathbb{F}^n$ of size $|M| = m$ is (s,t)-sumset evasive if for any set $S \subseteq \mathbb{F}^n$ of size $|S| = s$, it holds that*[12]

$$|tS \cap M| < m\,.$$

The next lemma asserts that linear data structure lower bounds, sumset evasive sets and subspaces of high outer dimension are all equivalent.

Lemma 2. *Let $M \subseteq \mathbb{F}^n$ be a set of size $|M| = m$, let $A \in \mathbb{F}^{m \times n}$ be a matrix composed of the vectors of M, and let $V \subseteq \mathbb{F}^m$ be the column space of A. The following are equivalent:*

(1) There is an (s,t) linear data structure computing A.

(2) $D_V(t) \leq s$.

(3) M is not *(s,t)-sumset evasive.*

Proof. (2) $\implies$ (1): Since $D_V(t) \leq s$, there exists a t-sparse subspace $U \subseteq \mathbb{F}^m$ of $\dim(U) \leq s$ such that $V \subseteq U$. Then let $Q \in \mathbb{F}^{m \times s}$ be a t-row sparse matrix whose columns generate U. Since $V \subseteq U$, each column of A is a linear combination of columns from Q. Therefore, there exists a matrix $P \in \mathbb{F}^{s \times n}$ such that $A = Q \cdot P$. We show that there exists an (s,t) linear data structure $\mathcal{D}$ which computes A. Indeed, let the preprocessing function of $\mathcal{D}$ be the linear transformation defined by P, and let the query algorithm be the linear function defined by Q. Since Q is t-row sparse, and $P \in \mathbb{F}^{s \times n}$, $\mathcal{D}$ is an (s,t) linear data structure.

(3) $\implies$ (2): Since M is not (s,t)-sumset evasive, there exists a set $S \subseteq \mathbb{F}^n$ of size $|S| = s$ such that $M \subseteq tS$. Let $P \in \mathbb{F}^{s \times n}$ be a matrix composed of the vectors of S. Since $M \subseteq tS$, there exists a t-row sparse matrix $Q \in \mathbb{F}^{m \times s}$ such that $A = Q \cdot P$. Let $U \subseteq \mathbb{F}^m$ be the column space of Q. We have that $V \subseteq U$ and $\dim(U) \leq s$.

(1) $\implies$ (3): Let $\mathcal{D}$ be an (s,t) linear data structure which computes A. Let $P \in \mathbb{F}^{s \times n}$ be the linear transformation computed by its preprocessing function, and $Q \in \mathbb{F}^{m \times s}$ be the linear transformation computed by its query function. Let $S \subseteq \mathbb{F}^n$ be the set of s rows of P. Since Q is t-row sparse, the set tS contains the set M which contradicts sumset evasiveness of M.

□

[11] We note that the t-sum of S is often defined as the set $\{s_1 + \ldots + s_t : \forall i, s_i \in S\}$. We abuse the notation by using the term t-sum for all linear combinations of length t of the vectors from S.

[12] We shall see that the definition of sumset evasive sets exactly captures the hardness of linear data structure problems. One can extend the definition of sumset evasive sets to capture the hardness of approximating a linear problem by a linear data structure. Since the main focus of this work is exact data structures, we omit this extended definition.

2.4 Rigidity

Definition 5 (Rigidity)**.** *A matrix $M \in \mathbb{F}^{m\times n}$ is* (m,n,r,t)-row rigid *if any matrix which differs from M in at most t elements in each row, has rank at least r. M is* (m,n,r,t)-globally rigid *if any matrix which differs from M in at most t elements has rank at least r.*

In other words, M is rigid if it *cannot* be written as a sum $M = A + B$ of a sparse matrix A and a low rank matrix B.

Now we define a stronger notion of rigidity which is invariant under basis changes.

Definition 6 (Strong rigidity)**.** *A subspace $U \subseteq \mathbb{F}^m$ of dimension $n = \dim(U)$ is* (m,n,r,t)-strongly row rigid *if U cannot be written as a sum $U = A + B$ of a t-sparse subspace $A \subseteq \mathbb{F}^m$ of $\dim(A) \leq n$ and a subspace $B \subseteq \mathbb{F}^m$ of dimension $< r$. We abuse the notation by saying that a matrix $M \in \mathbb{F}^{m\times n}$ is (m,n,r,t)-strongly row rigid if its column space is (m,n,r,t)-strongly row rigid. Similarly, we say that U is* (m,n,r,t)-strongly globally rigid *if U cannot be written as a sum $U = A + B$ of a subspace $A \subseteq \mathbb{F}^m$ generated by a t-globally sparse matrix, and a subspace $B \subseteq \mathbb{F}^m$ of dimension $< r$.*[13]

Lemma 3 (Strong rigidity implies rigidity)**.** *If $M \in \mathbb{F}^{m\times n}$ is (m,n,r,t)-strongly row rigid, then M is (m,n,r,t)-row rigid.*

Proof. Assume that M is not (m,n,r,t)-row rigid. Then, by Definition 5, there exist matrices $A, B \in \mathbb{F}^{m\times n}$, where A is t-sparse, and $\mathsf{rk}(B) < r$, such that $M = A + B$. Let V_M, V_A, and V_B be the column spaces of M, A, and B, respectfully. We have that $V_M \subseteq V_A + V_B$. This implies thay $V_M = V'_A + V'_B$ for some $V'_A \subseteq V_A$ and $V'_B \subseteq V_B$. Since V_A is t-sparse, so is V'_A, and $\dim(V'_B) \leq \dim(V_B) < r$, which contradicts the definition of strong rigidity of M. □

Friedman [Fri93] defines strong rigidity in the same way as inner dimension. Now we show that this definition is equivalent to the definition above. The following simple lemma (which is a modified version of Proposition 3 from [PP06]) will play a key role in our proof of Theorem 1. It asserts that if a matrix M is non-rigid, then there must be some sparse subspace with a significant intersection with the column space of M.

Lemma 4 (Inner dimension is equivalent to strong rigidity)**.** *Let $V \subseteq \mathbb{F}^m$ be a subspace of $\dim(V) = n$. V is (m,n,r,t)-strongly row rigid if and only if $d_V(t) \leq n - r$.*

Proof. Assume that V is not (m,n,r,t)-strongly row rigid. Then, by Definition 6, there exist subspaces $A, B \subseteq \mathbb{F}^m$, where A is t-sparse, $\dim(A) \leq n$ and $\dim(B) < r$, such that $V = A + B$. From $V = A + B$ we have that $A + V = A + B$ which gives us that

$$\dim(A) + \dim(V) - \dim(A \cap V) = \dim(A + V) = \dim(A + B) \leq \dim(A) + \dim(B)$$

and

$$\dim(A \cap V) \geq \dim(V) - \dim(B) > n - r\,,$$

[13] We remark that while strong rigidity is interesting for rectangular matrices, and many of the known constructions of rigid matrices are actually strongly rigid (see, e.g., Proposition 4 in Appendix A), this definition is meaningless for square matrices. Indeed, any subspace $U \subseteq \mathbb{F}^n$ of $\dim(U) = n$ equals $I + 0$ where I is generated by the 1-sparse identity matrix, and $\mathsf{rk}(0) = 0$.

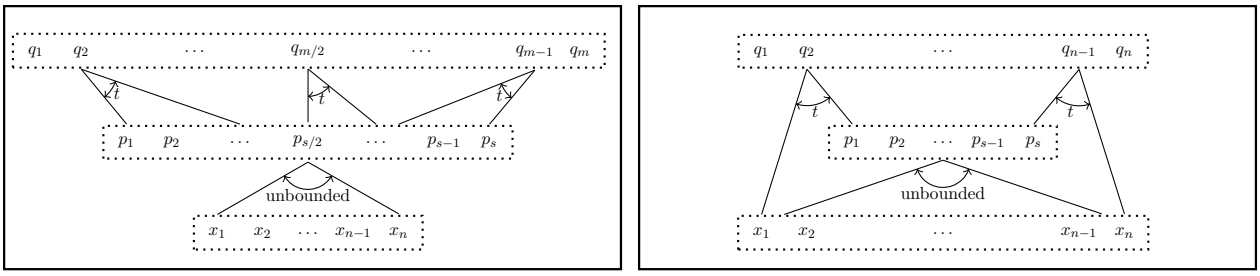

Figure 1: (a) A (nonadaptive) static data structure as a depth-2 circuit. The n input nodes feed $s \geq n$ memory cells, and we do not pose any restrictions on linear functions computed in memory cells. Each query (or output gate) depends only on t memory cells. In a typical scenario, t is as low as $t = (\log n)^{O(1)}$ or $t = n^{\varepsilon}$. (b) Depth-2 circuit resulting from Valiant's reduction. The n inputs feed only $s = O(n/\log\log n)$ middle layer gates. Again, we do not pose any restrictions on the linear functions computed in the middle layer gates. Each of the n output gates depends only on $t < n^{\varepsilon}$ inputs and middle layer gates.

which implies that $d_V(t) \geq \dim(A \cap V) > n - r$.

In the other direction: Assume that $d_V(t) > n - r$. Then, by Definition 2, there exists a t-sparse subspace $U \subseteq \mathbb{F}^m$ of $\dim(U) \leq n$, such that $\dim(U \cap V) > n - r$. Thus, there exists a subspace $W \subseteq \mathbb{F}^m$ of $\dim(W) < r$ such that $V = U + W$. □

2.5 Circuit Lower Bounds

A long-standing open problem in circuit complexity is to prove a super-linear lower bound on the size of circuits of depth $O(\log n)$ computing an explicit function [Val77, AB09, Frontier 3]. The same question remains open for linear circuits (i.e., circuits where each gate computes a linear combination of two of its inputs) computing an explicit linear map $f\colon \{0,1\}^n \to \{0,1\}^n$. Using a classical graph-theoretic result [EGS75], Valiant [Val77] reduced this problem to a problem about depth-2 circuits of a special kind: there are only $O(n/\log\log n)$ gates in the middle layer which depend on the n inputs, and each output gate depends on n^{ε} input and middle layer gates (for an arbitrary constant ε). Note that a static data structure can be thought of as a depth-2 circuit with n inputs, m outputs, s middle layer gates which depend on inputs, where each output depends on t gates in the middle layer. Figures 1 (a) and (b) illustrate the depth-2 circuits corresponding to static data structures and Valiant's reduction.

From Valiant's reduction (see Figure 1 (b)) one can conclude that if a linear-size log-depth linear circuit computes a linear map $M \in \mathbb{F}^{n \times n}$, then M can be written as $M = A + C \cdot D$, where A, C, and D encode the dependence of outputs on inputs, the dependence of outputs on the middle layer, and the dependence of the middle layer on inputs, respectfully. Note that since every output has fan-in t, we can conclude that the matrices A and C are t-sparse. Formally, Valiant gave the following decomposition:

Theorem 4 ([Val77])**.** *Let $m \geq n$. For every $c, \varepsilon > 0$, there exists $\delta > 0$ such that any linear map $M \in \mathbb{F}^{m \times n}$ computable by a circuit of size cm and depth $c \log m$, can be decomposed as*

$$M = A + C \cdot D\,,$$

where $A \in \mathbb{F}^{m\times n}, C \in \mathbb{F}^{m\times s}, D \in \mathbb{F}^{s\times n}$, A and C are t-sparse. There are two decompositions:

- $s = \varepsilon m, t = 2^{(\log m)^{1-\delta}}$;
- $s = \frac{\delta m}{\log\log m}, t = m^{\varepsilon}$.

In particular, from the dimensions of C and D, $C \cdot D$ has rank at most s. Thus, $M = A + B$ for a t-sparse A and $\mathsf{rk}(B) \leq s$.

Corollary 1. *An $(n, n, \varepsilon n, n^{\delta})$-row rigid matrix (for arbitrary constants ε, δ) does not have linear-size log-depth circuits.*

The best known (row) rigidity lower bound for the regime of $r = \varepsilon n$ is only $t \geq \Omega(1)$. If we relax the requirement of matrices to be (almost) square, then for $m = \mathrm{poly}(n)$ we know examples of $m \times n$ matrices with $t \geq \Omega(\log n)$ [APY09, SY11].

The problem of finding non-trivial circuit lower bounds and rigid matrices is open not only in $\mathbf{P}$, but also in larger uniform classes like $\mathbf{P}^{\mathbf{NP}}$ or even $\mathbf{E}^{\mathbf{NP}}$.

3 Main Building Blocks

This section contains our two main tools for converting data structure lower bounds into rigidity lower bounds. In Section 3.1, we show that a rectangular matrix $M \in \mathbb{F}^{m\times n}$ which is hard for linear data structures contains a rectangular submatrix of high row-rigidity. In Section 3.2, we show that a rectangular matrix of high row-rigidity can be transformed in a rigid *square* matrix (with some loss in the relative rank parameter but almost no loss in the relative sparsity parameter).

3.1 Connection Between Outer and Inner Dimensions

In this section we shall prove that every matrix either has small outer dimension or contains a matrix of small inner dimension. From Lemmas 2 and 4, this implies that every matrix which cannot be computed by efficient data structures (has large outer dimension) contains a rigid submatrix (submatrix of low inner dimension). We start with the following auxiliary lemma.

Lemma 5. *Let m, n, k be positive integers. If $M \in \mathbb{F}^{m\times n}$ has $d_M(t) \geq \mathsf{rk}(M) - k$, then M can be decomposed as*

$$M = A \cdot B + M' \cdot C\,,$$

where $M' \in \mathbb{F}^{m\times k}$ is a submatrix of M, $A \in \mathbb{F}^{m\times n}$ is t-row sparse, $B \in \mathbb{F}^{n\times n}$, $C \in \mathbb{F}^{k\times n}$.

Moreover, if $\mathbb{F}$ is a finite field of size $2^{m^{O(1)}}$, such a decomposition can be found in time $\mathrm{poly}(n, m)$ *with an* **NP** *oracle.*

Proof. Let V be the column space of M. By Definition 2, there exists a t-sparse subspace $U \subseteq \mathbb{F}^m, \dim(U) \leq \dim(V) \leq n$, s.t. $\dim(V \cap U) \geq \mathsf{rk}(M) - k$. Let $A \in \mathbb{F}^{m\times n}$ be a *t-row sparse* matrix generating U.

Let us now extend A with at most k column vectors from M to generate the column space of M, and let $M' \in \mathbb{F}^{m\times k}$ be a matrix formed by these columns. Since the columns of $A \in \mathbb{F}^{m\times n}$ and

$M' \in \mathbb{F}^{m\times k}$ together generate the column space of M, there exist matrices $B \in \mathbb{F}^{n\times n}, C \in \mathbb{F}^{k\times n}$ such that

$$M = A \cdot B + M' \cdot C\,.$$

Let INNER-DIM be the language of triples (M, t, d) such that the matrix M has inner dimension (with the sparsity parameter t) $d_M(t) \geq d$. Since there is a polynomial-size witness (a t-row sparse matrix whose column space intersects with M in at least d dimensions) which can be verified in polynomial time, INNER-DIM $\in$ **NP**. Now we apply the standard search-to-decision reduction. Namely, we define **NP** languages so that we could use binary search to find each coordinate of a matrix A witnessing high inner dimension of M. For a field of size $2^{m^{O(1)}}$, this can be done with $\mathrm{poly}(m, n)$ queries to the **NP** oracle. Now, we can just use Gaussian elimination (running in time $\mathrm{poly}(n, m)$) to find a matrix M', and then the matrices B and C. □

We are now ready to present the main result of this section.

Algorithm 1 Find a Submatrix with Low Inner Dimension in a Matrix with High Outer Dimension

Input: Parameters ε, k, t, and a matrix $M \in \mathbb{F}^{m\times n}$ with $D_M(tk + n\varepsilon^k) \geq \frac{n}{1-\varepsilon}$.

Output: A submatrix $M' \in \mathbb{F}^{m\times n'}$ of M with $d_{M'}(t) < \mathsf{rk}(M') - \varepsilon n'$ and $n' \geq n\varepsilon^k$.

Let $n_i := n\varepsilon^i$ for every $0 \leq i \leq k$

Let $M_0 = M$

1: **for** $i = 0$ to $k - 1$ **do**

2: **if** $M_i \in \mathbb{F}^{m\times n_i}$ has $d_{M_i}(t) < \mathsf{rk}(M_i) - \varepsilon n_i$ **then**

3: **return** M_i

4: Let $k = \varepsilon n_i = n_{i+1}$, $d_{M_i}(t) \geq \mathsf{rk}(M_i) - k$

5: By Lemma 5, there exist t-row sparse $A_i \in \mathbb{F}^{m\times n_i}$, $M_{i+1} \in \mathbb{F}^{m\times n_{i+1}}$, $B_i \in \mathbb{F}^{n_i\times n_i}$, and $C_i \in \mathbb{F}^{n_{i+1}\times n_i}$, where M_{i+1} is a submatrix of M_i, such that:

$$A_iB_i + M_{i+1}C_i = M_i \tag{2}$$

Theorem 5. *Let t and k be positive integers, and let $0 < \varepsilon < 1$. If $M \in \mathbb{F}^{m\times n}$ is a matrix of outer dimension*

$$D_M\left(tk + n\varepsilon^k\right) \geq \frac{n}{1-\varepsilon}\,,$$

then for some $n' \geq n\varepsilon^k$, M contains a submatrix $M' \in \mathbb{F}^{m\times n'}$ of inner dimension

$$d_{M'}(t) \leq \mathsf{rk}(M') - \varepsilon n'\,.$$

Moreover, if $\mathbb{F}$ is a finite field of size $2^{m^{O(1)}}$, such a submatrix M' can be found in time $\mathrm{poly}(n, m)$ *with an* **NP** *oracle.*

Proof. We prove that Algorithm 1 must return a submatrix of M of the claimed size with low inner dimension. Assume towards a contradiction that the algorithm did not return a matrix M_i

in Step 3 in any of the iterations $0 \le i \le k-1$. Then, from Equation (2), we have

$$\begin{aligned} M = M_0 &= A_0B_0 + M_1C_0 \\ &= A_0B_0 + (A_1B_1 + M_2C_1)C_0 \\ &= A_0B_0 + A_1B_1C_0 + M_2C_1C_0 \\ &= A_0B_0 + A_1B_1C_0 + A_2B_2C_1C_0 + M_3C_2C_1C_0 \\ &\dots \\ &= A_0B_0 + A_1B_1C_0 + \ldots + M_kC_{k-1}C_{k-2}\ldots C_0 \\ &= \sum_{i=0}^{k-1} A_iB_i \prod_{j=i-1}^{0} C_j + M_k \prod_{j=k-1}^{0} C_j \\ &= \begin{bmatrix} A_0 & A_1 & \dots & A_{k-1} & M_k \end{bmatrix} \cdot \begin{bmatrix} D_0 \\ D_1 \\ \dots \\ D_{k-1} \\ D_k \end{bmatrix}, \end{aligned}$$

where

$$D_i := \begin{cases} B_i \cdot \prod_{j=i-1}^{0} C_j\,, & \text{for } 1 \le i < k \\ \prod_{j=k-1}^{0} C_j, & \text{for } i = k\,. \end{cases}$$

Now, recall that each A_i is t-sparse, and that $M_k \in \mathbb{F}^{m \times n_k}$ where $n_k = n\varepsilon^k$. Thus, we have that $M = AB$ where A has at most $tk + n_k = tk + n\varepsilon^k$ non-zero entries per row, and $B \in \mathbb{F}^{s \times n}$ for $s = \sum_{i=0}^{k} n_i = \sum_{i=0}^{k} n\varepsilon^i < \frac{n}{1-\varepsilon}$. This implies that the columns of M can be generated by the columns of a $\left(tk + n\varepsilon^k\right)$-row sparse matrix $A \in \mathbb{F}^{m \times n'}$, which contradicts the assumption about the outer dimension of M.

Now we show that one can implement Algorithm 1 in time polynomial in n and m with an **NP** oracle. Since the language INNER-DIM (the language of triples (M, t, d) such that $d_M(t) \ge d$) is in **NP**, Step 2 can be done with an **NP** oracle. Step 5 can be performed in polynomial time (with an **NP** oracle) by Lemma 5. □

We conclude this section with an application of Theorem 5 to data structures.

Lemma 6. *Let $\varepsilon > 0$ be a constant. If the linear map given by a matrix $M \in \mathbb{F}^{m \times n}$ cannot be solved by an $\left(\frac{n}{1-\varepsilon}, (t+1) \cdot \frac{\log(n/t)}{\log(1/\varepsilon)}\right)$ linear data structure, then M contains an $(m, n', \varepsilon n', t)$-row rigid submatrix $M' \in \mathbb{F}^{m \times n'}$ for some $n' \ge t$.*

Proof. Since M cannot be solved by the claimed data structure, by Lemma 2,

$$D_M\left((t+1) \cdot \frac{\log(n/t)}{\log(1/\varepsilon)}\right) > \frac{n}{1-\varepsilon}\,.$$

Let us set $k = \frac{\log(n/t)}{\log(1/\varepsilon)}$. Then, by Theorem 5, M contains a submatrix $M' \in \mathbb{F}^{m \times n'}$ with $d_{M'}(t) \le \mathsf{rk}(M') - \varepsilon n'$ for

$$n' \ge n\varepsilon^k = n\varepsilon^{\frac{\log(n/t)}{\log(1/\varepsilon)}} = n \cdot \frac{t}{n} = t\,.$$

By Lemma 4, the column space of M' is $(m, n', \varepsilon n', t)$-strongly row rigid. Now, by Lemma 3, M' is $(m, n', \varepsilon n', t)$-row rigid. □

3.2 Row Rigidity to Global Rigidity

One drawback of Theorem 5 is that the recursive algorithm produces skewed matrices (as we only recurse on the column space). To remedy this limitation, in this subsection we exhibit a reduction from worst case to average case rigidity, which will allow us to translate our results to square matrix rigidity with some loss in the rank parameter (thereby proving Theorem 2). The main ingredient of our reduction is the use of locally decodable codes:

Definition 7 (Locally Decodable Codes)**.** *A mapping $E : \mathbb{F}^n \mapsto \mathbb{F}^m$ is a (q, δ, ε) locally decodable code (LDC) if there exists a probabilistic procedure $D\colon [n] \times \mathbb{F}^m \to \mathbb{F}$ such that*

- *For every $i \in [n]$ and $y \in \mathbb{F}^m$, $D(i, y)$ reads at most q positions of y;*
- *For every $i \in [n]$, $x \in \mathbb{F}^n$ and $v \in \mathbb{F}^m$ such that $|v| \leq \delta m$,*

$$\Pr[D(i, E(x) + v) = x_i] \geq 1 - \varepsilon\,.$$

An LDC is called linear *if the corresponding map E is linear. In this case we can identify the code E with its generating matrix $E \in \mathbb{F}^{m \times n}$.*

There are constructions of LDCs over all fields with $m = \text{poly}(n), q = (\log n)^{1+\alpha}$ for arbitrarily small $\alpha > 0$, and constant δ and ε (based on Reed-Muller codes).

Lemma 7 ([Dvi11], Corollary 3.14)**.** *Let $\mathbb{F}$ be a finite field. For every $\alpha, \varepsilon > 0$ there exists $\delta = \delta(\varepsilon) > 0$ and an explicit family of $((\log n)^{1+\alpha}, \delta, \varepsilon)$-linear LDCs $M \in \mathbb{F}^{m \times n}$ for $m = n^{O(1/\alpha)}$.*

We will use the following property of linear LDCs.

Lemma 8 (Implicit in [GKST02, DS07])**.** *Let $E \in \mathbb{F}^{m \times n}$ be a $(q, \delta, 3/4)$ linear LDC, and let R be a set of at least $(1 - \delta)m$ rows of E. For any $i \in [n]$, there exists a set of q rows in R which spans the ith standard basis vector e_i.*

We are now ready to present the main result of this section.

Theorem 6. *Let $M \in \mathbb{F}^{m \times n}$ be a matrix, $E \in \mathbb{F}^{m' \times m}$ be a $(q, \delta, 3/4)$-linear LDC, and let $A = EM$ (i.e., the matrix obtained by applying E to each column of M).*

- *If M is $(m, n, r, t + 1)$-row rigid, then A is $(m', n, r, \frac{\delta t m'}{q})$-globally rigid.*
- *If M is $(m, n, r, t + 1)$-strongly row rigid, then A is $(m', n, r, \frac{\delta t m'}{q})$-strongly globally rigid.*

Proof.

- Assume towards a contradiction that A is not $(m', n, r, \frac{\delta t m'}{q})$-globally rigid. Then $A = L + S$, where $\mathsf{rk}(L) \leq r$ and S is $\frac{\delta t m'}{q}$-globally sparse. Let S' be the set of rows of S with at most $\frac{t}{q}$ non-zero elements. By Markov's inequality there are at least $(1 - \delta)m'$ rows in S', let L' be the corresponding rows of L. By row-rigidity of M, some row i of M is $(t + 1)$-far from

the space generated by the rows of L (that it, the Hamming distance between this row and any vector in the span of the rows of L is at least $t+1$). By Lemma 8, there are q rows in $L'+S'$ which span the ith row of M. In particular, the ith row of M has distance at most $q \cdot \frac{t}{q}$ from the rowspan of L', which contradicts the assumption that this row is $(t+1)$-far from the space generated by the rows of L.

- In order to show that the resulting matrix A is strongly rigid, it suffices to show that the application of linear LDC commutes with basis changes. Assume towards a contradiction that the resulting matrix A is not strongly globally rigid. This implies that there exists an invertible matrix $U \in \mathbb{F}^{n \times n}$ such that

$$AU = (EM)U = E(MU)$$

is not $(m', n, r, \frac{\delta t m'}{q})$-globally rigid. Notice that from strong rigidity of M, we have that MU is $(m, n, r, t+1)$-row rigid. Thus, by the first item of this theorem, AU is also $(m', n, r, \frac{\delta t m'}{q})$-globally rigid.

□

We remark that the same argument as in the proof of Theorem 6 can be also used to give a worst-case to average-case reduction for linear *data-structures* (with a similar loss of $(\log n)^{1+\alpha}$ factor in the number of probes).

We next show that given a rectangular $m \times n$ matrix and *row rigidity* t, one can efficiently produce a square matrix of size $m' \times m'$ for $m' = n^{O(1/\alpha)}$ with row rigidity $\frac{m'}{n} \cdot \frac{t}{(\log n)^{1+\alpha}}$ (for the same rank parameter). That is, one can increase the rigidity proportionally to the increase in size with a loss of only $(\log n)^{1+\alpha}$ factor.

Corollary 2. *For every constant $\alpha > 0$, there is a polynomial-time algorithm which given an $(m, n, r, t+1)$-row rigid matrix $M \in \mathbb{F}^{m \times n}$, outputs a* square matrix $A \in \mathbb{F}^{m' \times m'}$ *which is $\left(m', m', r, \frac{m'^2}{n} \cdot \frac{t}{(\log m)^{1+\alpha}}\right)$-globally rigid for $m' = m^{O(1/\alpha)}$. In particular, A is $\left(m', m', r, \frac{m'}{n} \cdot \frac{t}{(\log m)^{1+\alpha}}\right)$-row rigid.*

Proof. Let $E \in \mathbb{F}^{m' \times m}$ be a $((\log m)^{1+\alpha/2}, \delta, \frac{3}{4})$-linear LDC (whose efficient construction is guaranteed by Lemma 7) for constant δ, and let $m' = m^{O(1/\alpha)}$ be a multiple of n. Then we construct $A \in \mathbb{F}^{m' \times m'}$ by stacking side by side (m'/n) copies of EM.

By Theorem 6, EM is $(m', n, r, \frac{tm'}{(\log m)^{1+\alpha}})$-globally rigid. In order to reduce the rank of A to r, one needs to reduce the rank of each copy of EM to at most r. Therefore, one needs to change at least $\frac{tm'}{(\log m)^{1+\alpha}} \cdot \frac{m'}{n} = \frac{m'^2}{n} \cdot \frac{t}{(\log m)^{1+\alpha}}$ entries in A in order to get rank at most r. This implies that A has global rigidity $\frac{m'^2}{n} \cdot \frac{t}{(\log m)^{1+\alpha}}$ and row rigidity $\frac{m'}{n} \cdot \frac{t}{(\log m)^{1+\alpha}}$ for the rank parameter r. □

4 Data Structures and Rigidity

In Section 2 we showed that a strong upper bound on the *inner dimension* of a matrix implies that the matrix has non-trivial rigidity, and that a strong lower bound on the *outer dimension* implies that the corresponding linear transformation cannot be computed by an efficient linear

data structure. In this section we use the relations between inner and outer dimensions, to show that any improvement on rigidity lower bounds will lead to higher data structures lower bounds (against linear space), while improvements (on (1)) in data structure lower bounds would yield new rigidity lower bounds. We state these (different) implications in various space regimes.

4.1 Linear Space

We will make use of the following known relation between inner and outer dimensions.

Proposition 2 ([PP06])**.** $d_V(t) + D_V(t) \geq 2 \dim V$.

This proposition directly yields the following connection between rigidity and data structure lower bounds.

Corollary 3. *If a matrix $M \in \mathbb{F}^{m \times n}$ is (m, n, r, t)-strongly row rigid, then the corresponding linear map cannot be computed by an $(n + r - 1, t)$ linear data structure.*

We remark that this corollary works for any function $r = r(n)$, including the regimes where $r = O(n)$ and $r = o(n)$.

Proof. From Lemma 4, we have that $d_M(t) \leq n - r$. By Proposition 2, this gives us that $D_M(t) \geq n + r$. Now Lemma 2 gives us that no $(n + r - 1, t)$ linear data structure can compute M. □

In particular, an $(m, n, (1 + \varepsilon)n, t)$-strongly row rigid matrix implies a lower bound of t on the query time of linear data structures with linear space $s = (1+\varepsilon)n$. We remark that the best known rigidity bound in this regime for $m = \text{poly}(n)$ is $t = \Omega(\log n)$ which matches the best known lower bound for linear space data structures. Any improvement to $t = \omega(\log n)$ on the known rigidity construction would lead to a new data structure lower bound (against data structures with small linear space).

Now we use Lemma 6 to show that the opposite direction (with a slight change of parameters) also holds for a submatrix of M.

Theorem 7.

1. *(Poly-logarithmic Lower Bounds) Let $\varepsilon > 0$ and $c \geq 1$ be constants. If the linear map given by a matrix $M \in \mathbb{F}^{m \times n}$ cannot be solved by an $\left(\frac{n}{1-\varepsilon}, (\log n)^c\right)$ linear data structure, then M contains an $(m, n', \varepsilon n', \alpha \cdot (\log n)^{c-1})$-row rigid submatrix $M' \in \mathbb{F}^{m \times n'}$ for some constant $\alpha > 0$ and $n' \geq \alpha \cdot (\log n)^{c-1}$.*

2. *(Polynomial Lower Bounds) Let $\varepsilon, \delta > 0$ be constants. If the linear map given by a matrix $M \in \mathbb{F}^{m \times n}$ cannot be solved by an $\left(\frac{n}{1-\varepsilon}, n^\delta\right)$ linear data structure, then M contains an $(m, n', \varepsilon n', n^\alpha)$-row rigid submatrix $M' \in \mathbb{F}^{m \times n'}$ for any $\alpha < \delta$ and some $n' \geq n^\alpha$.*

3. *(Square Rigidity) Let $\varepsilon > 0, \gamma > 0$ and $c > 2$ be constants. If the linear map given by a matrix $M \in \mathbb{F}^{m \times n}$ cannot be solved by an $\left(\frac{n}{1-\varepsilon}, (\log n)^c\right)$ linear data structure, then there is a* square *matrix $M' \in \mathbb{F}^{m' \times m'}$ for $m' = m^{O(1)}$, such that M' is $(m', m', r, \frac{m'(\log n)^{c-2-\gamma}}{r})$-row rigid for some r (which depends on n).*

Moreover, if $|\mathbb{F}| = 2^{m^{O(1)}}$ *and* $M \in \mathbf{P}^{\mathbf{NP}}$*, then the family of matrices* M' *of high rigidity belongs to the class* $\mathbf{P}^{\mathbf{NP}} = \mathbf{DTIME}[\mathrm{poly}(m)]^{\mathbf{NP}}$.[14]

Proof.

1. Let us set $t = \frac{(\log n)^{c-1}}{\log 1/\varepsilon} - 1$. Now have that M cannot be solved by an $\left(\frac{n}{1-\varepsilon}, (t+1) \cdot \frac{\log(n/t)}{\log(1/\varepsilon)}\right)$ linear data structure. This, together with Lemma 6, implies that M contains an $(m, n', \varepsilon n', \frac{(\log n)^{c-1}}{\log 1/\varepsilon} - 1)$-row rigid submatrix.

2. Here we set $t = \frac{n^{\delta}}{\log(1/\varepsilon)\log(n)} - 1$. Again, Lemma 6 implies that M contains an $(m, n', \varepsilon n', t)$-row rigid submatrix for $n' \geq t$.

3. From the first bullet of this theorem, we get an $(m, n', \varepsilon n', \alpha \cdot (\log n)^{c-1})$-rigid submatrix $M' \in \mathbb{F}^{m \times n'}$. Now we apply Corollary 2 to get an $m' \times m'$ matrix which has row rigidity $m' \cdot \frac{(\log n)^{c-2-\gamma}}{n'}$ for the rank parameter $r = \varepsilon n'$.

□

We note that a data structure lower bound of $t \geq \omega\left((\log n)^2\right)$ will lead to a new bound on rigidity of rectangular matrices. Moreover, by the last bullet of this theorem, we have that a lower bound of of $t \geq \Omega\left((\log n)^{3+\varepsilon}\right)$ will lead to a new bound on rigidity of *square* matrices: it will give us a matrix which is (n, n, r, s)-row rigid for $s \geq \Omega\left(\frac{n}{r} \cdot \frac{t}{(\log n)^{2+\varepsilon/2}}\right)$ (which is better than the known bound of $s \geq \Omega\left(\frac{n}{r}\log\frac{n}{r}\right)$).

4.2 Super-linear Space

Recall that any data structure problem has two trivial solutions: $s = n, t = n$, and $s = m, t = 1$. A simple counting argument (see Appendix B) shows that for any $s < 0.99m$, a random linear problem requires $t = \Omega(n/\log s)$ query time. Here we show that near-optimal data structure lower bound, against space $s \geq \omega(m/\log\log m)$, would imply a super-linear circuit lower bound.

Theorem 8. *Let* $M \in \mathbb{F}^{m \times n}$ *be a matrix for* $m \geq n$*. If for some constant* $\varepsilon > 0$ *and every constant* $\delta > 0$

- M *cannot be computed by* $\left(\frac{\delta m}{\log\log m} + n, m^{\varepsilon}\right)$ *linear data structures,*
- *or* M *cannot be computed by* $\left(\varepsilon m + n, 2^{(\log m)^{1-\delta}}\right)$ *linear data structures,*

then M *cannot be computed by linear circuits of size* $O(m)$ *and depth* $O(\log m)$*.*

Proof. Assume towards a contradiction that M can be computed by a circuit of size cm and depth $c\log m$ for a constant c. Then, by Theorem 4, $M = A + C \cdot D$, where $A \in \mathbb{F}^{m \times n}, C \in \mathbb{F}^{m \times s}, D \in \mathbb{F}^{s \times n}$, A and C are t-row sparse. In particular, the column space of M is spanned by the column spaces of t-row sparse matrices A and C. That is, $D_M(t) \leq n + s$. By Lemma 2, M can be computed by an $(n + s, t)$ linear data structure. □

[14] When we say that a matrix M belongs to $\mathbf{P}^{\mathbf{NP}}$, we mean that there exists a family of matrices $M_n \in \mathbb{F}^{m(n) \times n}$ for infinitely many values of n such that each M_n can be computed by a polynomial time algorithm with an **NP** oracle.

While the bounds given in this theorem are interesting in the regime $m, s \gg n$, they also give a curious corollary for $m = O(n)$. For example, in the regime of $m = O(n)$ for $s = \frac{\delta m}{\log\log m} + n$ we know a lower bound of $t \geq \Omega(\log n)$ [Lar12]. An improvement of this bound to sub-polynomial $2^{(\log n)^{1-\delta}}$ would give a super-linear circuit lower bound.

Corollary 4. *Let $m = O(n)$. If for some constant $\varepsilon > 0$ and every constant $\delta > 0$ a linear map $M \in \mathbb{F}^{m \times n}$ cannot be solved by $\left(n(1+\varepsilon), 2^{(\log n)^{1-\delta}}\right)$ linear data structures, then M cannot be computed by linear circuits of size $O(m)$ and depth $O(\log m)$.*

4.3 Succinct Space

In the succinct regime, data structures can only use space $s = n + o(n)$. In this regime we know strong lower bounds for data structures. Namely, if $s = n + r$ for $r = o(n)$, then the best know lower bound over $\mathbb{F}_2$ is $t \geq \frac{n}{r}$ [GM07]. We will show that the succinct case corresponds exactly to the case of strong rigidity in the regime $r = o(n)$, and will use this connection to improve the known data structure lower bound by a logarithmic factor. We remark that one can extract the same lower bound of $t \geq \frac{n \log n}{r}$ from [Lar12] for a problem with *polynomially* many queries $m = \Omega(n^{1+\varepsilon})$, while our simple construction gives it for linear number of queries $m = O(n)$.

Theorem 9. *Let $1 \leq r(n) \leq n^{1-\varepsilon}$ and $(\log n)^{\delta} \leq t(n) \leq n$ be non-decreasing and time-constructible functions for some constant $\varepsilon, \delta > 0$. Then*[15]

- *An $(m, n, r(n), t(n))$-strongly row rigid matrix $M \in \mathbb{F}^{m \times n}$ cannot be computed by $(n + r(n) - 1, t(n))$ linear data structures.*
- *If there exists a constant $\mu > 0$, such that $M \in \mathbb{F}^{m \times n}$ cannot be computed by $(n + (1+\mu)r(n), (1+\mu)t(n))$ linear data structure and $M \in \mathbf{P}^{\mathbf{NP}}$, then there is $(m, n', r(n'), t(n'))$-strongly row rigid $M' \in \mathbf{P}^{\mathbf{NP}} = \mathbf{DTIME}[\mathrm{poly}(m)]^{\mathbf{NP}}$ for some $n' \geq \mu t(n)/2$.*

Proof. The first item of the Theorem follows directly from Corollary 3. For the other direction, we will run Algorithm 1 with slightly modified parameters.

For a positive integer i, let $r^{(i)}$ denote the composition of r with itself i times:

$$r^{(i)}(n) = \underbrace{r \circ \cdots \circ r}_{i \text{ times}}(n)\,.$$

Let k be the smallest number such that $n^{\varepsilon^k} \leq \mu t(n)/2$. We define $n_0 = n$, then $n_i = r(n_{i-1})$ for $0 < i < k$, and $n_k = \mu t(n)/2$. For $0 \leq i \leq k$, we define $t_i = t(n_i)$ and $r_i = r(n_i)$. Let us now run Algorithm 1. In Step 2, the algorithm will check whether $d_{M_i}(t_i) < \mathsf{rk}(M_i) - r_i$. If this inequality is satisfied, then the algorithm returns an $(m, n', r(n'), t(n'))$-strongly row rigid matrix M'. Again, as in the proof of Theorem 5, this algorithm can be implemented in polynomial time with an **NP** oracle.

If the algorithm does not return a matrix M_i for any $0 \leq i \leq k-1$, then we get a factorization $M = AB$, where the matrix A has at most $t' = \sum_{i=0}^{k-1} t_k + n_k$ non-zero entries per row, and $B \in \mathbb{F}^{s \times n}$ for $s' = \sum_{i=0}^{k} n_i$. In particular, M can be computed by an (s', t') linear data structure.

[15] This lemma applies to the whole spectrum of $r = o(n)$, but for the ease of presentation we restrict our attention to the regime of $r \leq n^{1-\varepsilon}$.

From $n_i = r(n_{i-1}) \leq (n_{i-1})^{1-\varepsilon}$, we have $s' \leq n + (1+\mu)r(n)$ for large enough values of n. Now, from $r(n) \leq n^{1-\varepsilon}$ and $t(n) \geq (\log n)^{\delta}$, we have

$$t_i = t(n_i) = t(r(n_{i-1})) \leq (1-\varepsilon)^{\delta} \cdot t(n_{i-1}) = (1-\varepsilon)^{\delta} \cdot t_{i-1}\,.$$

Thus,

$$t' = \sum_{i=0}^{k-1} t_k + n_k \leq t(n) \cdot \frac{1}{1-(1-\varepsilon)^{\delta}} + \mu t(n)/2 \leq (1+\mu)t(n)$$

for $\mu = \frac{2}{1-(1-\varepsilon)^{\delta}}$. This contradicts the assumption that M cannot be computed by $(n + (1+\mu)r(n), (1+\mu)t(n))$ linear data structure. □

We remark that one can also apply Corollary 2 here to obtain square rigid matrices (see, e.g., Corollary 6).

We claim that the best known rigidity lower bound $t = \Omega\left(\frac{n}{r}\log\frac{n}{r}\right)$ gives us the same lower bound on strong rigidity, and, thus, improves the known succinct data structures lower bounds by a logarithmic factor.

In the following we will make use of error-correcting codes with constant rate and constant relative distance (see, e.g., Justesen and Goppa codes).

Proposition 3 ([MS77, LG88, VL12])**.** *For any finite field $\mathbb{F}$ there exists an explicit family of linear error correcting codes with rate $1/4$ and constant relative distance $\delta = \Theta(1)$.*

In Appendix A we modily the proof of Friedman [Fri93] to get strong rigidity.

Proposition 4 ([Fri93])**.** *Let $\mathbb{F}$ be a finite filed of size $|\mathbb{F}| = q$, and let $M \in \mathbb{F}^{n\times n/4}$ be a transposed generator matrix of a code with constant relative distance δ. That is, the columns $c_1, \ldots, c_{n/4} \in \mathbb{F}^n$ of M form a basis of a linear code. Then M is $(n, n/4, r, t)$-strongly row rigid for any $r \geq \log n$ and any*

$$1 \leq t \leq O\left(\frac{n}{r}\left(\log_q\left(\frac{n}{r}\right) + \log_q(q-1)\right)\right) = O\left(\frac{n}{r}\cdot\max\left(\log_q\left(\frac{n}{r}\right), 1\right)\right)\,.$$

As a corollary of Theorem 9 and Proposition 4, we get a new data structure lower bound for the succinct case.

Corollary 5. *Let $\mathbb{F}$ be a finite filed of size $|\mathbb{F}| = q$, and let $M \in \mathbb{F}^{4n\times n}$ be a transposed generator matrix of a code with constant relative distance. Then for any $\log n \leq r \leq n^{1-\varepsilon}$ and any $1 \leq t \leq O\left(\frac{n}{r}\left(\log_q\left(\frac{n}{r}\right) + \log_q(q-1)\right)\right)$, M cannot be computed by a linear $(n+r, t)$ data structure for large enough n.*

We also note that improving this data structure lower bound for $s = n + 2^{(\log\log n)^{\omega(1)}}$ would resolve a big open problem in communication complexity.

Proposition 5 ([Raz89, Wun12])**.** *If $M \in \mathbb{F}^{n\times n}$ is (r,t)-row rigid for $r = 2^{(\log\log n)^{\omega(1)}}$ and $t = n/2^{(\log\log n)^{O(1)}}$, then the language L corresponding to M is not in the polynomial hierarchy for communication complexity $L \notin \mathbf{PH}^{cc}$.*

Theorem 9, Proposition 5, and Corollary 2 give us the following result.

Corollary 6. *If $M \in \mathbb{F}^{m \times n}$ cannot be computed by $(n + r, t)$ linear data structures for $m = n^{O(1)}$, $r = 2^{(\log \log n)^{\omega(1)}}$ and $t = n/2^{(\log \log n)^{O(1)}}$, and $M \in \mathbf{P^{NP}}$, then there exists a language $L \in \mathbf{P^{NP}}$ such that $L \notin \mathbf{PH}^{cc}$.*

Proof. By Theorem 9, M contains an $(m, n', r(n'), t(n'))$-row rigid submatrix M'. Now Corollary 2 gives us a matrix A which is $(m', m', r(n'), m' \frac{t}{n((\log m)^{1+\alpha})})$-row rigid. From $m' = n^{O(1)}$, we have that A is $(m', m', 2^{(\log \log m')^{\omega(1)}}, m'/2^{(\log \log m')^{O(1)}})$-row rigid, which finishes the proof. □

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[Aga04] Pankaj K. Agarwal. Range searching. In *Handbook of Discrete and Computational Geometry, Second Edition.*, pages 809–837. Chapman and Hall/CRC, 2004.

[APY09] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In *RANDOM 2009*, pages 339–351, 2009.

[BL13] Karl Bringmann and Kasper Green Larsen. Succinct sampling from discrete distributions. In *STOC 2013*, pages 775–782, 2013.

[BL15] Joshua Brody and Kasper Green Larsen. Adapt or die: Polynomial lower bounds for non-adaptive dynamic data structures. *Theory Comput.*, 11:471–489, 2015.

[Cha90] Bernard Chazelle. Lower bounds for orthogonal range searching: Part II. The arithmetic model. *J. ACM*, 37(3):439–463, 1990.

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007.

[Dvi11] Zeev Dvir. On matrix rigidity and locally self-correctable codes. *Comput. Complexity*, 20(2):367–388, 2011.

[EGS75] Paul Erdös, Ronald L. Graham, and Endre Szemerédi. On sparse graphs with dense long paths. *Comp. and Math. with Appl*, 1:145–161, 1975.

[Fre81] Michael L. Fredman. A lower bound on the complexity of orthogonal range queries. *J. ACM*, 28(4):696–705, 1981.

[Fri93] Joel Friedman. A note on matrix rigidity. *Combinatorica*, 13(2):235–239, 1993.

[GKST02] Oded Goldreich, Howard Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *CCC 2002*, pages 175–183, 2002.

[GM07] Anna Gál and Peter Bro Miltersen. The cell probe complexity of succinct data structures. *Theor. Comput. Sci.*, 379:405–417, July 2007.

[GT16] Oded Goldreich and Avishay Tal. Matrix rigidity of random toeplitz matrices. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 91–104. ACM, 2016.

[IW01] Russell Impagliazzo and Avi Wigderson. Randomness vs time: Derandomization under a uniform assumption. *J. Comput. Syst. Sci.*, 63(4):672–688, 2001.

[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.

[Lar12] Kasper Green Larsen. Higher cell probe lower bounds for evaluating polynomials. In *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.

[LG88] Jacobus Hendricus van Lint and Gerard van der Geer. *Introduction to coding theory and algebraic geometry*. Birkhäuser Basel, 1988.

[Lok09] Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Found. Trends Theor. Comput. Sci.*, 4(1-2):1–155, 2009.

[Mil93] Peter Bro Miltersen. The bit probe complexity measure revisited. In *STACS 1993*, pages 662–671, 1993.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*. Elsevier, 1977.

[Păt07] Mihai Pătraşcu. Lower bounds for 2-dimensional range counting. In *STOC 2007*, pages 40–46, 2007.

[Păt08] Mihai Pătraşcu. Unifying the landscape of cell-probe lower bounds. In *FOCS 2008*, pages 434–443, 2008.

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

[PR94] Pavel Pudlák and Vojtech Rödl. Some combinatorial-algebraic problems from complexity theory. *Discrete Math.*, 136(1-3):253–279, 1994.

[PTW10] Rina Panigrahy, Kunal Talwar, and Udi Wieder. Lower bounds on near neighbor search via metric expansion. In *FOCS 2010*, pages 805–814, 2010.

[Raz89] Alexander A. Razborov. On rigid matrices. *Manuscript*, 1989. In Russian.

[Sie04] Alan Siegel. On universal classes of extremely random constant-time hash functions. *SIAM J. Comput.*, 33(3):505–543, 2004.

[SSS97] Mohammad Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. A remark on matrix rigidity. *Inf. Process. Lett.*, 64(6):283–285, 1997.

[SY11] Shubhangi Saraf and Sergey Yekhanin. Noisy interpolation of sparse polynomials, and applications. In *CCC 2011*, pages 86–92, 2011.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *MFCS 1977*, pages 162–176, 1977.

[vEB90] Peter van Emde Boas. Machine models and simulation. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, pages 1–66. MIT Press, 1990.

[VL12] Jacobus Hendricus Van Lint. *Introduction to coding theory*, volume 86. Springer Science & Business Media, 2012.

[Wun12] Henning Wunderlich. On a theorem of Razborov. *Comput. Complex.*, 21(3):431–477, 2012.

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

A Omitted Proofs

In this appendix we give proofs of some known statements adjusted to our definitions. Propositions 2 and 4 were proven in [PP06] and [Fri93] for definition concerning column sparsity rather than row sparsity (which matters in this context), Proposition 1 was proven in [JS11] for linear depth-2 circuits rather than for linear data structures. We remark that the proofs in this appendix are not new, but are rather adjustments of the known proofs to our framework.

Proposition 1 (Lemma 2.5 [JS11], Ex. 13.7 [Juk12])**.** *Given an (s,t)-data structure $\mathcal{D}$ computing a linear transformation Vx for $V \in \mathbb{F}^{m\times n}, x \in \mathbb{F}^n$ with linear query function Q, one can efficiently construct an equivalent (s,t)-data structure where* both *the query function Q and the preprocessing function P are linear.*

Proof. Let $P\colon \mathbb{F}^n \to \mathbb{F}^s$ be the preprocessing function of $\mathcal{D}$, and let $Q \in \mathbb{F}^{m\times s}$ be a linear transformation computed by the query function of $\mathcal{D}$. Let $e_1, \ldots, e_n$ be the unit vectors in $\mathbb{F}^n$. Consider a new linear data structure $\mathcal{D}'$ where the preprocessing function computes a linear transformation, which for a vector $x = \sum_{i=1}^n x_i e_i$ outputs $P'(x) = \sum_{i=1}^n x_i P(e_i)$, and the query function stays the same: $Q'(x) = Q(x)$.

Since the original data structure $\mathcal{D}$ computes the linear transformation V, it holds that:

$$\forall x \in \mathbb{F}^n\colon Vx = Q \cdot P(x)\,.$$

Now, by the linearity of matrix-vector products, the new data structure computes

$$Q' \cdot P'(x) = Q \cdot \left(\sum_{i=1}^n x_i P(e_i)\right) = \sum_{i=1}^n x_i Q \cdot P(e_i) = \sum_{i=1}^n x_i V e_i = V \cdot \left(\sum_{i=1}^n x_i e_i\right) = Vx\,.$$

□

Proposition 2 ([PP06])**.** $d_V(t) + D_V(t) \geq 2 \dim V$.

Proof. Let $U \supseteq V$ be a t-sparse subspace of dimension $D_V(t)$ (the existence of U is guaranteed by the definition of outer dimension). Let A_U be a t-sparse matrix generating U, and let A_W be the first $\dim(V)$ columns of A_U. Now let W be the column space of A_W. Clearly A_W and W are also t-sparse. From the definition of the inner dimension, we have that $d_V(t) \geq \dim(V \cap W)$.

On the other hand, U contains V and W. Thus,

$$D_V(t) = \dim(U) \geq \dim(V + W) = \dim(V) + \dim(W) - \dim(V \cap W) \geq 2\dim(V) - d_V(t)\,.$$

□

Proposition 4 ([Fri93])**.** *Let $\mathbb{F}$ be a finite filed of size $|\mathbb{F}| = q$, and let $M \in \mathbb{F}^{n \times n/4}$ be a transposed generator matrix of a code with constant relative distance δ. That is, the columns $c_1, \ldots, c_{n/4} \in \mathbb{F}^n$ of M form a basis of a linear code. Then M is $(n, n/4, r, t)$-strongly row rigid for any $r \geq \log n$ and any*

$$1 \leq t \leq O\left(\frac{n}{r}\left(\log_q\left(\frac{n}{r}\right) + \log_q(q-1)\right)\right) = O\left(\frac{n}{r} \cdot \max\left(\log_q\left(\frac{n}{r}\right), 1\right)\right)\,.$$

Proof. By Lemma 4, it suffices to show that $d_M(t) \leq n/4 - r$. Let $B \in \mathbb{F}^{n \times n/4}$ be a t-sparse matrix, and let $C = (c_1, \ldots, c_{n/8}) \in \mathbb{F}^{n \times n/8}$ be the $n/8$ sparsest columns of B. Note that by Markov's inequality, each column of C has at most $8t$ non-zero entries. Let V be the column space of M, and U be the column space of C. We will show that $\dim(U \cap V) \leq n/8 - r$, which will finish the proof.

Let

$$W = \left\{ w = (w_1, \ldots, w_{n/8}) \in \mathbb{F}^{n/8} \colon \sum_{i=1}^{n/8} w_i c_i \in U \right\}\,.$$

Assume towards a contradiction that $\dim(W) = \dim(U \cap V) > n/8 - r$. This implies that W contains a non-zero point of Hamming weight at most a for any a such that

$$\left|\text{Hamming ball of radius } a/2 \text{ in } \mathbb{F}^{n/8}\right| \geq q^r\,.$$

In particular, there is a point of Hamming weight at most a for a satisfying

$$\binom{n/8}{a/2}(q-1)^{a/2} \geq q^r\,.$$

On the other hand, a point of Hamming weight a in W, gives a non-zero codeword of Hamming weight at most $a \cdot 8k$. Since we know that all non-zero codewords have Hamming weight at least δn, we get $a \geq \frac{\delta n}{8k}$. Now we have

$$\begin{aligned} r &\geq \log_q\left(\binom{n/8}{a/2}(q-1)^{a/2}\right) \\ &\geq \frac{a}{2}\log_q\left(\frac{n}{4a}\right) + \frac{a}{2}\log_q(q-1) \\ &\geq \frac{\delta n}{16t}\left(\log_q\left(\frac{2t}{\delta}\right) + \log_q(q-1)\right) \\ &= \Omega\left(\frac{n}{t}\left(\log_q(t) + \log_q(q-1)\right)\right)\,. \end{aligned}$$

Or, equivalently, $t \geq \Omega\left(\frac{n}{r}\left(\log_q\left(\frac{n}{r}\right) + \log_q(q-1)\right)\right)$, which leads to a contradiction. □

B Complexity of Random Problems

Lemma 9. *Let $\mathbb{F}$ be a finite field of size $|\mathbb{F}| = q$, and let $\varepsilon > 0$ be a constant.*

- *For any $n \le s \le (1-\varepsilon)m$, there exists a linear problem $M \in \mathbb{F}^{m\times n}$ that can only be solved by (s,t) linear data structures with $t \ge \Omega\left(\min\left(n, \frac{n\log q}{\log s}\right)\right)$.*
- *For any $s \ge n^{1+\varepsilon}$, every linear problem $M \in \mathbb{F}^{m\times n}$ can be solved by an (s,t) linear data structure with $t \le O\left(\min\left(n, \frac{n\log q}{\log s}\right)\right)$.*

Proof.

- The total number of homogeneous linear functions of n arguments is q^n. There are q^{sn} ways to choose s linear functions computed in the memory cells. For a fixed choice of s functions, there are at most $s^t \cdot q^t$ different functions which can be computed as linear compositions of t out of s elements. Thus, there are at most
$$q^{sn} \cdot (s^t \cdot q^t)^m$$
m-tuples of linear functions which can be computed by data structures with s memory cells. On the other hand, there are q^{nm} distinct m-tuples of linear functions. Therefore, as long as
$$q^{sn+mt} s^{mt} < q^{nm}\,,$$
there is a linear data structure problem with m outputs which cannot be computed by a data structure with s memory cells and query time t. Let us take $t = \frac{\varepsilon n \log q}{2\log(qs)}$. Then, from $s \le (1-\varepsilon)m$, we have
$$q^{sn+mt}s^{mt} \le q^{(1-\varepsilon)nm}(qs)^{mt} = q^{(1-\varepsilon)nm}(qs)^{\frac{\varepsilon nm\log q}{2\log(qs)}} = q^{(1-\varepsilon)nm}q^{\varepsilon nm/2} < q^{nm}\,.$$
In particular, there is a linear data structure problem which requires
$$t > \frac{\varepsilon n\log q}{2\log(qs)} \ge \Omega\left(\min\left(n, \frac{n\log q}{\log s}\right)\right)\,.$$
- Let $\mu = 1 + \frac{1}{\varepsilon}$. It is trivial to see that any data structure can be solved with space $s = n$ and query time $t = n$. Thus, it suffices to show that if $\log s > 2\mu\log q$, then M can be solved by a data structure with $t \le O\left(\frac{n\log q}{\log s}\right)$. Let
$$t = \left\lceil \frac{n}{\frac{\log s}{\mu\log q} - 1} \right\rceil\,. \tag{3}$$
Let us partition the n inputs into t parts each of size $\lfloor n/t\rfloor$ or $\lceil n/t\rceil$. Let the preprocessing function $P\colon \mathbb{F}^n \to \mathbb{F}^s$ compute all $q^{\lceil n/t\rceil}$ homogeneous linear combinations of the inputs of each part. In order to store this, we need $s' = t\cdot q^{\lceil n/t\rceil}$ memory cells. Now, every query can be computed as a sum of at most t memory cells. It remains to show that $s' \le s$:
$$s' = t\cdot q^{\lceil n/t\rceil} \le n\cdot q^{\lceil n/t\rceil} \le n\cdot q^{\frac{\log s}{\mu\log q}} = n\cdot s^{\frac{1}{\mu}} \le s^{\frac{1}{1+\varepsilon}+\frac{1}{\mu}} \le s\,,$$
where the first inequality follows from (3) and $\log s > 2\mu\log q$, the second inequality is due to (3), the third inequality follows from $s \ge n^{1+\varepsilon}$, and the last one is due to the choice of $\mu = 1 + \frac{1}{\varepsilon}$.

□

C Rigidity Implies Low Inner Dimension

In this appendix we give an alternative (more constructive) proof of Lemma 4 (with a small loss in the upper bound on the inner dimension).

Lemma 10. *Let $M \in \mathbb{F}^{m\times n}$ be a matrix of rank n, and $V \subseteq V^m$ be its column space. If M is (m, n, r, t)-row rigid, then $d_V(t) < n - 2r$.*

Proof. Assume towards a contradiction that M is not (m, n, r, t)-row rigid. Then, by Definition 5, there exist matrices $A, B \in \mathbb{F}^{m\times n}$, where A is t-sparse, and $\mathsf{rk}(B) \leq r$, such that $M = A + B$.

Let $L \in \mathbb{F}^{m\times m}$ be a matrix representing a linear map which vanishes on V: $\ker(L) = V$. To construct such an L, one can take a basis $(v_1, \ldots, v_k)$ of V, and extend it to a basis $(v_1, \ldots, v_k, w_1, \ldots, w_{m-k})$ of F^m. Then define $L(v_i) = 0$ for $1 \leq i \leq k$, and $L(w_j) = w_j$ for $1 \leq j \leq m - k$, and extend L by linearity.

Now, observe that if we apply L to the equality $M = A+B$, we have $0 = LM = LA+LB$, and, in particular, $\mathsf{rk}(LA) = \mathsf{rk}(LB)$. Note that the rank of the matrix on the right side is $\mathsf{rk}(LB) \leq \mathsf{rk}(B) \leq r$. By subadditivity of rank, we have $\mathsf{rk}(A) \geq \mathsf{rk}(M) - \mathsf{rk}(B) \geq n - r$.

Let U be the column space of t-sparse matrix A. From $\mathsf{rk}(A) \geq n - r$ and $\mathsf{rk}(LA) \leq r$, we have that $\dim(U \cap \ker(L)) = \dim(U \cap V) \geq n - 2r$, which finishes the proof. □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il