comput. complex. (2021) 30:6

1016-3328/21/010001-40
published online April 2, 2021
https://doi.org/10.1007/s00037-021-00205-2

LOWER BOUNDS FOR MATRIX FACTORIZATION

Ben Lee Volk and Mrinal Kumar

Abstract.
We study the problem of constructing explicit families of matrices which cannot be expressed as a product of a few sparse matrices. In addition to being a natural mathematical question on its own, this problem appears in various incarnations in computer science; the most significant being in the context of lower bounds for algebraic circuits which compute linear transformations, matrix rigidity and data structure lower bounds.
We first show, for every constant d, a deterministic construction in time $\exp(n^{1-\Omega(1/d)})$ of a family $\{M_n\}$ of $n \times n$ matrices which cannot be expressed as a product $M_n = A_1 \cdots A_d$ where the total sparsity of $A_1, \ldots, A_d$ is less than $n^{1+1/(2d)}$. In other words, any depth-d linear circuit computing the linear transformation $M_n \cdot \mathbf{x}$ has size at least $n^{1+\Omega(1/d)}$. The prior best lower bounds for this problem were barely super-linear, and were obtained by a long line of research based on the study of super-concentrators. We improve these lower bounds at the cost of a blow up in the time required to construct these matrices. Previously, however, such constructions were not known even in time $2^{O(n)}$ with the aid of an NP oracle.
We then outline an approach for proving improved lower bounds through a certain derandomization problem, and use this approach to prove asymptotically optimal quadratic lower bounds for natural special cases, which generalize many of the common matrix decompositions.

Keywords. Algebraic complexity, Linear circuits, Matrix factorization, Lower bounds

Subject classification. 68Q04, 68Q15, 68Q17

1. Introduction

This work concerns the following (informally stated) very natural problem:

Open Question 1.1. *Exhibit an explicit matrix $A \in \mathbb{F}^{n\times n}$, such that A cannot be written as $A = BC$, where $B \in \mathbb{F}^{n\times m}$ and $C \in \mathbb{F}^{m\times n}$ are sparse matrices.*

Before bothering ourselves with the precise meaning of the words "explicit" and "sparse" in the above problem, we discuss the various contexts in which this problem presents itself.

1.1. Linear circuits and matrix factorization. Algebraic complexity theory studies the complexity of computing polynomials using arithmetic operations: addition, subtraction, multiplication and division. An algebraic circuit over a field $\mathbb{F}$ is an acyclic directed graph whose vertices of in-degree 0, also called inputs, are labeled by indetermeinates $\{x_1, \ldots, x_n\}$ or field elements from $\mathbb{F}$, and every internal node is labeled with an arithmetic operation. The circuit computes rational functions in the natural way, and the polynomials (or rational functions) computed by the circuit are those computed by its vertices of out-degree 0, called the outputs. This framework is general enough to encompass virtually all the known algorithms for algebraic computational problems. The size of the circuit is defined to be the number of edges in it. For a more detailed background on algebraic circuits, see Shpilka & Yehudayoff (2010).

Perhaps the simplest non-trivial class of polynomials is the class of linear (or affine) functions. Accordingly, such polynomials can be computed by a very simple class of circuits called *linear circuits*: these are algebraic circuits which are only allowed to use addition and multiplication by a scalar. It is often convenient to consider graphs with labels on the edges as well: every internal node is an addition gate, and for $c \in \mathbb{F}$, an edged labeled c from a vertex v to a vertex u denotes that the output of v is multiplied by c when feeding into u. Thus, every node computes a linear combination of its inputs.

It is not hard to show that any arithmetic circuit for computing a set of linear functions can be converted into a linear circuit with only a constant blow-up in size (see (Bürgisser *et al.* 1997), Theorem 13.1 for a proof of this statement, and note that eliminating division gates requires that the field $\mathbb{F}$ in question is large enough, an assumption we will make in this paper when needed).

Clearly, every set of n linear functions on n variables (represented by a matrix $A \in \mathbb{F}^{n \times n}$) can be computed by a linear circuit of size $O(n^2)$. Using counting arguments (over finite fields) or dimension arguments (over infinite fields), it can be shown that for a random or generic matrix this upper bound is fairly tight. Thus, a central open problem in algebraic complexity theory is to prove any super-linear lower bound for an *explicit* family of matrices $\{A_n\}$ where $A_n \in \mathbb{F}^{n \times n}$. The standard notion of explicitness in complexity theory is that there is a deterministic algorithm that outputs the matrix A_n in $\mathsf{poly}(n)$ time, although more or less stringent definitions can be considered as well.

Despite decades of research and multiple partial results, such lower bounds are not known.[1] In order to gain insight into the general model of computation, research has focused on limited models of linear circuits, such as monotone circuits, circuits with bounded coefficients, or bounded depth circuits. We defer a more thorough discussion on previous work to Section 1.3, and proceed to describe bounded depth circuits, which are the focus of this work.

The *depth* of a circuit is the length (in edges) of a longest path from an input to an output. Constant depth circuits appear to be a particularly weak model of computation. However, even this model is surprisingly powerful (see also Section 1.2).

The "easiest" non-trivial model is the model of depth-2 linear circuits. A depth 2 linear circuit computing a linear transformation $A \in \mathbb{F}^{n \times n}$ consists of a bottom layer of n input gates, a middle layer of m gates, and a top layer of n output gates. We assume, without loss of generality, that the circuit is *layered*, in the sense that every edge goes either from the bottom to the middle layer, or from the

[1] We remark that super-linear lower bounds for general arithmetic circuits are known, but for polynomials of high degree (Baur & Strassen 1983; Strassen 1973).

middle to the top layer. Indeed, every edge going directly from the bottom to the top layer can be replaced by a path of length 2; this transformation increases the size of the circuit by at most a factor of 2.

By letting $C \in \mathbb{F}^{m \times n}$ be the adjacency matrix of the (labeled) subgraph between the bottom and the middle layer, and $B \in \mathbb{F}^{n \times m}$ be the adjacency matrix as the subgraph between the middle and the top layer, it is clear that $A = BC$. Thus, a decomposition of A into the product of two sparse matrices is equivalent to saying that A has a small depth-2 linear circuit. This argument can be generalized, in exactly the same way, to depth-d circuits and decompositions of the form $A = A_1 \cdots A_d$, for constant d.

Weak super-linear lower bounds are known for constant depth linear circuits. They are based on the following observation, due to Valiant (1975): for subsets $S, T \subseteq [n]$ of size k, let $A_{S,T}$ denote the submatrix of A indexed by rows in S and columns in T. If $A_{S,T}$ has rank k, the minimal vertex cut in the subcircuit restricted to input from S and outputs from T[2] is of size at least k: indeed, a smaller cut corresponds to a factorization $A_{S,T} = PQ$ for $P \in \mathbb{F}^{k \times r}$ and $Q \in \mathbb{F}^{r \times k}$ for $r < k$, contradicting the rank assumption. Using Menger's theorem, it is now possible to deduce that if A is a matrix such that for every S, T as above the matrix $A_{S,T}$ is non-singular, then the circuit computing A contains, for every subcircuit which corresponds to such S, T, at least k vertex disjoint paths from S to T. Such graphs were named *superconcentrators* by Valiant, and their minimal size was extensively studied (Alon & Pudlák 1994; Dolev *et al.* 1983; Pippenger 1977, 1982; Pudlák 1994; Radhakrishnan & Ta-Shma 2000; Valiant 1975).

Superconcentrators of logarithmic depth and linear size do exist, so while this approach cannot show lower bounds for circuits of logarithmic depth, it is possible to show that for constant d, any depth-d superconcentrator has size at least $n \cdot \lambda_d(n)$, where $\lambda_d(n)$ is a function that unfortunately grows very slowly with n.

[2]Such a subcircuit is obtained from the original circuit by erasing all input gates not in S and output gates not in T, and their outgoing and incoming edges, and then recursively removing all inner gates whose in-degree or out-degree is 0, along with all edges touching them. The resulting subcircuit computes the transformation given by $A_{S,T}$.

For example, $\lambda_2(n) = \Theta(\log^2 n/\log\log n)$, $\lambda_3(n) = \Theta(\log\log n)$, $\lambda_4(n) = \lambda_5(n) = \log^*(n)$, and so on. Such lower bounds apply for any matrix whose minors of all orders are non-zero, e.g., a Cauchy matrix given by $A_{i,j} = 1/(x_i - y_j)$ for any distinct $x_1, \ldots, x_n, y_1, \ldots, y_n$. Over finite fields it is possible to modify the proof and obtain similar lower bounds for matrices defining good error correcting codes (Gál *et al.* 2013).

These lower bounds on the size of superconcentrators are tight: for every $d \in \mathbb{N}$, there exists a super-concentrator of depth d and size $O(n \cdot \lambda_d(n))$. It is thus impossible to improve the lower bounds only using this technique.

1.2. Matrix rigidity. A demonstration of the surprising power of depth-2 circuits can be seen using the notion of *matrix rigidity*, a pseudorandom property of matrices which we now recall. A matrix $A \in \mathbb{F}^{n\times n}$ is (r, s) rigid if A *cannot* be written as a sum $A = R + S$ where R is a matrix of rank r, and S is a matrix with at most s non-zero entries. Valiant (1977) famously proved that if A is computed by a linear circuit with bounded fan-in of depth $O(\log n)$ and size $O(n)$, then A is not $(\varepsilon n, n^{1+\delta})$ rigid for every $\varepsilon, \delta > 0$.[3] It follows that an explicit construction of $(\varepsilon n, n^{1+\delta})$ matrix, for some $\varepsilon, \delta > 0$, will imply a super-linear lower bound for linear circuits of depth $O(\log n)$. Pudlák (1994) observed that similar rigidity parameters will imply even stronger lower bounds for constant depth circuits. A random matrix (over infinite fields) is $(r, (n-r)^2)$-rigid, but the best explicit constructions have rigidity $(r, n^2/r \cdot \log(n/r))$ (Friedman 1993; Shokrollahi *et al.* 1997), which is insufficient for proving lower bounds.

Observe that a decomposition $A = R + S$ where $\mathsf{rank}(R) = \varepsilon n$ and S is $n^{1+\delta}$-sparse corresponds to a depth-2 circuit with a very special structure and with at most $2\varepsilon n^2 + n^{1+\delta}$ edges (this circuit is not layered, but as we explained above, this does not make a significant difference). In particular, one way of interpreting Valiant's result is as a non-trivial depth reduction from depth $O(\log n)$ to depth 2, so that proving *any* depth-2 linear circuit lower bound of

[3] In fact, one can obtain slightly better parameters. See, for example, Valiant (1977) or Dvir *et al.* (2019).

$\Omega(n^2)$ for an explicit matrix, will imply a lower bound for linear circuits of depth $O(\log n)$.[4] This can be seen as the linear circuit analog of similar strong depth reduction theorems for general algebraic circuits (Agrawal & Vinay 2008; Gupta *et al.* 2016; Koiran 2012; Tavenas 2015).

However, we would like to argue that proving lower bounds for depth-2 circuits is in fact *necessary* for proving rigidity lower bounds, by observing that *upper bounds* on the depth-2 complexity of A give upper bounds on its rigidity parameters, or in the contrapositive form, high rigidity means high depth-2 complexity.

Observation. *Suppose A has a depth-2 circuit of size at most $n^{1+\varepsilon}$. Then for any $\delta > 0$, A is* not *$(2\delta n, n^{1+3\varepsilon}/\delta^2)$-rigid*

Proof. Suppose $A = BC$ can be computed by a depth-2 circuit of size $n^{1+\varepsilon}$. Let m be as before the number of columns of B (which equals the number of rows of C), and note that we may assume $m \leq n^{1+\varepsilon}$, as zero columns of B or zero rows of C can be omitted. For $i \in [m]$, let B_i denote the i-th column of B, and C_i the i-th row of C, so that $A = \sum_{i=1}^{m} B_i C_i$. Fix $\delta > 0$, and say $i \in [m]$ is *dense* if either B_i or C_i has more than n^{ε}/δ non-zero entries; otherwise, i is *sparse*. Since B can have at most δn columns with sparsity of more than n^{ε}/δ, and similarly for the rows of C, the number of dense i-s is at most $2\delta n$. It follows that

$$A = \sum_{i \text{ dense}} B_i C_i + \sum_{i \text{ sparse}} B_i C_i.$$

The first sum is a matrix of rank at most $2\delta n$, and the second is a matrix whose sparsity is at most $m \cdot n^{2\varepsilon}/\delta^2 = n^{1+3\varepsilon}/\delta^2$. □

Thus, proving rigidity lower bounds of the type required to carry out Valiant's approach necessarily means proving lower bounds of the form "$n^{1+\varepsilon}$" on the depth-2 complexity of A (we

[4] We note that this statement makes sense only over large fields, as over fixed finite fields, it is always possible to prove an *upper bound* of $O(n^2/\log n)$ on the depth-2 complexity of any matrix (Jukna & Sergeev 2013). This does not contradict the fact that rigid matrices exist over finite fields — a decomposition to $R + S$ is a very special type of depth-2 circuit.

remark that the argument above is very similar to the aforementioned result of Pudlák (1994); Pudlák's argument is stated in a slightly different language and in greater generality). Since proving rigidity lower bounds is a long-standing open problem, we view the problem of proving an $\Omega(n^{1+\varepsilon})$ lower bound for depth-2 circuits as an important milestone towards this.

1.3. Related work. As mentioned in Section 1.1, there are no non-trivial known lower bounds for general linear circuits, and for bounded depth circuits, the best lower bounds follow from the lower bounds on bounded depth super-concentrators, which are barely super-linear.

Shoup & Smolensky (1996) give a lower bound of $\Omega(dn^{1+1/d})$ for depth-d circuits computing a certain linear transformation given by a matrix $A \in \mathbb{R}^{n\times n}$. Unfortunately, the matrices for which their lower bound holds are not explicit from the complexity theoretic point of view, despite having a very succinct mathematical description (for example, one can take $A_{i,j} = \sqrt{p_{i,j}}$ for n^2 distinct prime numbers $p_{i,j}$). For the same matrix, they in fact prove super-linear lower bounds for circuits of depth up to $\mathsf{polylog}(n)$.

Quite informally, the intuition behind their lower bounds is that all small bounded depth linear circuits can be described as lying in the image of a low-degree polynomial map in a small number of variables, and thus, if the elements of A are sufficiently "algebraically rich", for a certain specific measure, A cannot be computed by such a circuit. This same philosophy lies behind Raz's elusive function approach for proving lower bounds for algebraic circuits (Raz 2010). In particular, among other results, Raz uses an argument which can be seen as a modification of the technique of Shoup and Smolensky (as worked out in (Shpilka & Yehudayoff 2010)) to prove lower bounds for bounded depth algebraic circuits computing bounded degree polynomials.

One class of linear circuits which has attracted significant attention is the class of circuits with bounded coefficients. Here, the circuit is only allowed to multiply by scalars with absolute value of at most some constant. For definiteness, we may assume this constant is 1 (this does not affect the complexity by more than a constant factor). The earliest result for this model is Morgenstern's inge-

nious proof (Morgenstern 1973) of an $\Omega(n \log n)$ lower bound on bounded coefficient circuits computing the discrete Fourier transform matrix (this lower bound is matched by the upper bound given by the Cooley-Tukey FFT algorithm, which is a bounded coefficient linear circuit). For depth-d circuits, Pudlák (2000) has proved lower bounds of the form $\Omega(dn^{1+1/d})$ for the same matrix.

Another natural subclass which was considered in earlier works is the class of monotone linear circuits. These are circuits which are defined over $\mathbb{R}$, and can only use non-negative scalars. Chazelle (2001) observed that it is possible to prove lower bounds in this model, even against unbounded-depth circuits, for any boolean matrix with no large monochromatic rectangle. Instantiated with the recent explicit constructions of bipartite Ramsey graphs (Ben-Aroya *et al.* 2017; Chattopadhyay & Zuckerman 2019; Cohen 2017; Li 2019), this gives an almost optimal $n^{2-o(1)}$ lower bound against such circuits. The main observation in the proof is that if A does not have monochromatic $t \times t$ rectangle, then since the model is monotone and no cancellations are allowed, every internal node which computes a linear function supported on at least t variables cannot be connected to more than t output gates.

For a more detailed survey on these results, see the survey by Lokam (2009).

The problem of matrix factorization into sparse matrices also appears in other areas in computer science. One such example is the study of data structures. A dynamic data structure with n inputs and q queries is a pair of algorithms whose purpose is to update and retrieve certain data under a sequence of operations, while minimizing the memory access. In the group model, it is given by a pair of algorithms. The update algorithm is represented by a matrix $U \in \mathbb{F}^{s \times n}$. Given $x \in \mathbb{F}^n$, thought of as assignment of weights to the n inputs, Ux computes a linear combination of those weights and stores them in memory. The query algorithm is given by a matrix $Q \in \mathbb{F}^{q \times s}$. Given a query, it computes a linear function of the s memory cells, and returns the answer. Hence, an "update" operation followed by a "retrieve" operation computes the linear transformation given by $A = QU$.

The sparsity of the matrices Q and U roughly correspond to the average case query and update time of the data structure. Thus, a matrix $A \in \mathbb{F}^{q \times n}$ which cannot be factored as $A = QU$ for a row-sparse Q and column-sparse U gives a data structure problem with a lower bound on its query or update time. Lower bounds for this model were proved by Fredman (1982), Fredman & Saks (1989), Pătraşcu & Demaine (2006), Pătraşcu (2007), Larsen (2012, 2014) and Larsen *et al.* (2018), but none of these results beats the lower bounds for depth-2 circuits obtained using superconcentrators.

A related model is that of a static data structures, which is again given by a factorization $A = QP$, where now we are interested in trade-offs between the space s of the data structure and its worst case query time, while not being charged for the total sparsity of P. A recent work of Dvir *et al.* (2019) showed that proving lower bounds for this model is related to the problem of matrix rigidity from Section 1.2.

Despite the overall similarity, there are several key technical differences between the linear circuit complexity and the data structure problems. One important issue is that in many examples in the data structures literature, the number of queries q is polynomially larger than n, while the lower bounds on running time are still measured as functions of the number of inputs n. This makes sense in the data structure settings, but from a circuit complexity point of view, a set of say n^3 linear functions trivially requires a circuit of size n^3, and thus a lower bound of say $n\,\mathsf{polylog}(n)$ is meaningless in that setting.

Finally, we briefly remark that the problem of factorizing a matrix into a product of two or more sparse matrices is also ubiquitous in machine learning and related areas. Naturally, research in those areas did not focus on lower bounds but rather on algorithms for finding such a representation, assuming it exists, sometimes heuristically, and it is usually enough to approximate the target matrix A. In particular, algorithms have been proposed for the very related problems of non-negative matrix factorization (Lee & Seung 2000)[5]

[5]It is interesting to observe that for the problem of factorizing matrices into non-negative matrices it is quite easy to prove almost-optimal lower bounds even for unbounded depth linear circuits, as mentioned in Section 1.3.

or sparse dictionary learning (Mairal *et al.* 2009), and there are also connections to the analysis of deep neural networks (Neyshabur & Panigrahy 2013).

1.4. Our results. In this paper, we prove several results regarding bounded depth linear circuits which we now discuss.

Lower bounds for depth-d linear circuits. We start by considering general depth-d circuits. We give the first deterministic construction in time $2^{o(n)}$ of matrices which require depth-d circuits of size $n^{1+\Omega(1/d)}$.

THEOREM 1.2. *Let $\mathbb{F}_q$ be a finite field and $d \geq 2$ an integer. Let $\mathbb{E}$ be an extension of $\mathbb{F}_q$ of degree $\exp(n^{1-\Omega(1/d)})$. There exists a family of $n \times n$ matrices $\{A_n\}_{n\in\mathbb{N}}$ over $\mathbb{E}$, which can be constructed in time $\exp(n^{1-\Omega(1/d)})$ such that every depth-d linear circuit computing A_n, even over the algebraic closure of $\mathbb{F}_q$, has size at least $n^{1+\Omega(1/d)}$.*

Similarly, there exists a family of $n \times n$ matrices $\{B_n\}_{n\in\mathbb{N}}$ over $\mathbb{Q}$, whose entries are integers of bit complexity $\exp(n^{1-\Omega(1/d)})$, such that every depth-d linear circuit computing B_n over $\mathbb{C}$ has size at least $n^{1+\Omega(1/d)}$.

This theorem is proved in Section 2. We remark again that the previous best lower bounds against general depth-d linear circuits (for matrices that can be constructed in polynomial time) are barely super-linear and much weaker than $n^{1+\varepsilon}$. In the recent work of Dvir *et al.* (2019) it was pointed out that currently there are not even known constructions of rigid matrices (with parameters that would imply lower bounds) in classes such as $\mathbf{E}^{\mathbf{NP}}$. By arguing directly about circuit size, and not about rigidity, Theorem 1.2 gives constructions of matrices in a much smaller complexity class, which enjoy the same bounded-depth complexity lower bounds as would follow from optimal constructions of rigid matrices using the results of Pudlák (1994).

In a related and independent work, Alman & Chen (2019) constructed in $\mathbf{P}^{\mathbf{NP}}$ (i.e., in polynomial time and using an **NP** oracle), for infinitely many n's, an $n \times n$ matrix with rigidity parameters which suffice for proving a lower bound of $\Omega(n \cdot 2^{\log(n)^{1/4-\varepsilon}})$ on its

depth-2 complexity. Compared to their work, our construction lies in an incomparable complexity class (we do not use an NP oracle at the expense of a longer running time), extends for all depths $d \geq 2$, works for all large enough n, and provides stronger lower bounds. Furthermore, Alman and Chen use complexity theoretic techniques which are very different from our algebraic techniques. The parameters of their construction was recently improved in Bhangale *et al.* (2020). We refer to Alman & Chen (2019) for some further discussion on the differences and similarities.

While the statement in Theorem 1.2 holds for any $d \geq 2$, for $d = 2$ there is a much simpler construction of a hard family of matrices in quasi-polynomial time.

THEOREM 1.3. *Let $\mathbb{F}$ be any field and c be any positive constant. Then, there is a family $\{A_n\}_{n\in\mathbb{N}}$ of $n \times n$ matrices which can be constructed in time $\exp(O(\log^{2c+1} n))$ such that any depth-2 linear circuit computing A_n even over the algebraic closure of $\mathbb{F}$ has size at least $\Omega(n \log^c n)$.*

For every constant $c \geq 2$, this theorem already improves upon the current best lower bound of $\Omega(n \log^2 n/\log\log n)$ known for this problem (see (Radhakrishnan & Ta-Shma 2000)). This construction is based on an exponential time construction of a small hard matrix, and then amplifying its hardness using a direct sum construction (note, however, that over infinite fields even the fact that a hard matrix can be constructed in exponential time, while not very hard to prove, is not *completely* obvious). For completeness, we describe this simple construction in Section 2.7.

Lower bounds for restricted depth-2 linear circuits. Given the importance of the model of depth-2 linear circuits, as explained above, and its resistance to strong lower bounds, we then move on to consider several natural subclasses of depth-2 circuits. These classes in particular correspond to almost all common matrix decompositions. We are able to prove asymptotically optimal $\Omega(n^2)$ lower bounds for these restricted models. As mentioned above, such lower bounds for general depth-2 circuits will imply super-

linear lower bounds for logarithmic depth linear circuits, thus resolving a major open problem.

Symmetric circuits. A symmetric depth-2 circuit (over $\mathbb{R}$) is a circuit of the form $B^T B$ for some $B \in \mathbb{R}^{m \times n}$ (considered as a graph, the subgraph between the middle and the top layer is the "mirror image" of the subgraph between the bottom and middle layer). Over $\mathbb{C}$, one should take the conjugate transpose B^* instead of B^T.

Symmetric circuits are a natural computational model for computing positive semi-definite (PSD) matrix. Clearly, every symmetric circuit computes a PSD matrix, and every PSD matrix has a (non-unique) symmetric circuit. In particular, a Cholesky decomposition of PSD matrices corresponds to a computation by a symmetric circuit (of a very special form).

We prove asymptotically optimal lower bounds for this model.

THEOREM 1.4. *There exists an explicit family of real $n \times n$ PSD matrices $\{A_n\}_{n \in \mathbb{N}}$ such that every symmetric circuit computing A_n (over $\mathbb{R}$ or $\mathbb{C}$) has size $\Omega(n^2)$.*

We do not know whether every depth-2 linear circuit for a PSD matrix can be converted to a symmetric circuit with a small blow-up in size. One way to phrase this question is given below.

QUESTION 1.5. *Is there a constant $c < 2$, such that every PSD matrix $A \in \mathbb{R}^{n \times n}$ which can be computed by a linear circuit of size s, can be computed by a symmetric circuit of size $O(s^c)$?*

A positive answer for Question 1.5 will imply, using Theorem 1.4, an $\Omega(n^{1+\varepsilon})$ lower bound for depth-2 linear circuits.

Invertible circuits. Invertible circuits are circuits of the form BC, where either B or C are invertible (but not necessarily both). We stress that invertible circuits can (and do) compute non-invertible matrices. In particular, if $B \in \mathbb{F}^{n \times m}$ and $C \in \mathbb{F}^{m \times n}$, here we require $m = n$.

Invertible circuits generalize many of the common matrix decompositions, such as QR decomposition, eigendecomposition, singular value decomposition[6] and LUP decomposition (in the case where the matrix L is required to be unit lower triangular).[7]

We prove optimal lower bounds for invertible circuits.

THEOREM 1.6. *Let $\mathbb{F}$ be a large enough field. There exists an explicit family of $n \times n$ matrices $\{A_n\}_{n\in\mathbb{N}}$ over $\mathbb{F}$ such that every invertible circuit computing A_n has size $\Omega(n^2)$.*

If A is an invertible matrix, then clearly every depth-2 circuit with $m = n$ must be an invertible circuit. However, our technique for proving Theorem 1.6 crucially requires the hard matrix A to be non-invertible.

1.5. Proof Overview. Our proofs rely on a few different ideas coming from algebraic complexity theory, coding theory, arithmetic combinatorics and the theory of derandomization. We now discuss some of the key aspects.

Shoup-Smolensky dimension. For the proof of Theorem 1.2, we rely on the notion of *Shoup-Smolensky* dimension as a measure of complexity of matrices. Shoup-Smolensky dimensions are a family of measures, parametrized by $t \in \mathbb{N}$, of "algebraic richness" of the entries of a matrix (see Definition 2.1 for details), which is supposed to capture the intuition that matrices with small circuits should depend on a few "parameters" and thus should not posses much richness.

Shoup & Smolensky (1996) showed that for an appropriate choice of parameters, this measure is non-trivially small for linear transformations with small linear circuits of depth at most $\mathsf{poly}(\log n)$. Informally, as the order t gets larger, this measure becomes useful against stronger models of computation; however, it

[6] A diagonal matrix can be multiplied with the matrix to its left or to its right, without increasing the sparsity, to obtain an invertible depth-2 circuit.

[7] The sparsity of UP equals the sparsity of U, as P simply permutes the columns of U, so every LUP decomposition corresponds to the invertible depth-2 circuit given by $L(UP)$.

also becomes harder to construct matrices which have a large complexity with respect to this measure (and hence cannot be computed by a small linear circuit). Shoup and Smolensky do this by constructing hard matrices which do not have small bit complexity (and hence this construction is not complexity theoretically explicit) but do have short and succinct mathematical description.

For our proof, we first observe that for bounded depth circuits it suffices to use much smaller order t than what Shoup and Smolensky used. This observation was also made by Raz (2010) in a similar context, but using the language of elusive functions.

We then use this observation to "derandomize", in a certain sense, an exponential time construction of a hard matrix, by giving deterministic constructions of matrices with large Shoup-Smolensky dimension.

A key ingredient of our proof is a connection between the notion of Sidon Sets in arithmetic combinatorics and Shoup-Smolensky dimension (see Section 2.4 for details). Our construction is in two steps. In the first step we construct matrices with entries in $\mathbb{F}[y]$ which have a large Shoup-Smolensky dimension over $\mathbb{F}$, and degree of every entry is not too large. In the next step, we go from these univariate matrices to a matrix with entries in an appropriate low degree extension of $\mathbb{F}$ while still maintaining the Shoup-Smolensky dimension over $\mathbb{F}$. Our construction of hard matrices over the field of complex numbers is based on similar ideas but differs in some minor details.

Lower bounds via Polynomial Identity Testing. Our proofs for Theorem 1.4 and Theorem 1.6 are based on a derandomization argument. Connections between derandomization and lower bounds are prevalent in algebraic and Boolean complexity, but in our current setting they have not been widely studied before.

We say that a set $\mathcal{H}$ of $n \times n$ matrices is a *hitting set* for a class $\mathcal{C}$ of matrices if for every non-zero $A \in \mathcal{C}$ there is $H \in \mathcal{H}$ such that $\langle A, H \rangle := \sum_{i,j} A_{i,j} H_{i,j} \neq 0$.

As we describe in Section 3, It is simple yet important observation that a hitting set of size strictly less than n^2 for a class $\mathcal{C}$ of matrices implies an efficient construction of a matrix which is not in $\mathcal{C}$.

We carry out this idea for two different classes in the proofs of Theorem 1.4 and Theorem 1.6. A useful ingredient in our constructions is the use of maximum distance separable (MDS) codes (for example, Reed-Solomon codes), as their dual subspace is a small dimensional subspace which does not contain sparse non-zero vectors. Over the reals, it is also easy to give such construction based on the well known Descartes' rule of signs which says that a sparse univariate real polynomial cannot have too many real roots. We refer the reader to Section 3.1 for details.

2. Lower bounds for constant depth linear circuits

In this section, we prove Theorem 1.2. We start by describing the notion of Shoup-Smolensky dimension, but first we set up some notation.

2.1. Notation. We work with matrices whose entries lie in an appropriate extension of a base finite field $\mathbb{F}_p$. We follow the natural convention that the elements of this extension will be represented as univariate polynomials of appropriate degree over the base field, and the arithmetic is done modulo an explicitly given irreducible polynomial.

We use boldface letters $(\mathbf{x}, \mathbf{y})$ to denote vectors. The length of the vectors is understood from the context.

For a matrix M, $\|M\|_0$ denotes the number of non-zero entries in M.

2.2. Shoup-Smolensky Dimension. A useful concept will be the notion of Shoup-Smolensky dimension of sequences of elements of an extension $\mathbb{E}$ of a field $\mathbb{F}$.

Definition 2.1 (Shoup-Smolensky dimension). *Let $\mathbb{F}$ be a field, and $\mathbb{E}$ be an extension field of $\mathbb{F}$. Let $S = (a_1, \ldots, a_m)$ a sequence of elements of $\mathbb{E}$. For $t \in \mathbb{N}$, denote by $\mathsf{Prod}_t(S)$ the set of t-wise products of distinct entries of S that is,*

$$\mathsf{Prod}_t(S) = \left\{ \prod_{j=1}^{t} a_{i_j} : 1 \leq i_1 < i_2 < \cdots < i_t \leq m \right\}.$$

The Shoup-Smolensky dimension *of S of order t, denoted by $\Gamma_{t,\mathbb{F}}(S)$ is defined to be the dimension, over $\mathbb{F}$, of the vector space spanned by $\mathsf{Prod}_t(S)$.*

We also denote by $\Sigma_t(S)$ the number of distinct elements of $\mathbb{E}$ that can be obtained by summing *distinct elements of $\mathsf{Prod}_t(S)$.*

When $M \in \mathbb{E}^{n\times n}$ is a matrix we also regard it as a sequence of $m = n^2$ elements of $\mathbb{E}$ (under some order on the entries) and refer to the Shoup-Smolensky dimension of M.

2.3. Upper bounding the Shoup-Smolensky dimension for Sparse Products. The following lemma shows that any matrix computable by a depth-d linear circuit of size at most s has a somewhat small Shoup-Smolensky dimension. This statement was proved by Shoup & Smolensky (1996) (although we believe the formulation using matrix product makes the notation somewhat simpler). For completeness, we give the proof.

LEMMA 2.2. *Let $\mathbb{F}$ be a field, $\mathbb{E}$ an extension of $\mathbb{F}$ and $A \in \mathbb{E}^{n\times n}$ be a matrix such that $A = \prod_{i=1}^{d} P_i$ for $P_i \in \mathbb{E}^{n_i \times m_i}$, where $\sum_{i=1}^{d} \|P_i\|_0 \le s$. Then, for every $t \le n^2/4$ such that $s \ge dt$ it holds that*

$$\Gamma_{t,\mathbb{F}}(A) \le \left(e^d (2s/dt)^d\right)^t .$$

PROOF. Since

$$A_{i,j} = \left(\prod_{\ell=1}^{d} P_\ell\right)_{i,j} = \sum_{k_1,\ldots,k_{d-1}} (P_1)_{i,k_1} \cdot \left(\prod_{\ell=2}^{d-1} (P_\ell)_{k_{\ell-1},k_\ell}\right) \cdot (P_d)_{k_{d-1},j} ,$$

every element in $\mathsf{Prod}_t(A)$ is a sum of monomials of degree dt in the entries of $P_1, P_2, \ldots, P_d$, that is,

$$\Gamma_{t,\mathbb{F}}\left(\prod_{i=1}^{d} P_i\right) \le \binom{s+dt}{dt},$$

with the right hand side being the number of monomials of degree dt in s variables. Using the inequality $\binom{n}{k} \le (en/k)^k$,

$$\Gamma_{t,\mathbb{F}}(A) \le (e(1+s/dt))^{dt} \le \left(e^d (2s/dt)^d\right)^t ,$$

as desired. □

Over $\mathbb{Q}$, we do not wish to use field extensions (which would give rise to elements with infinite bit complexity). Thus, we use a similar argument that replaces the measure $\Gamma_{t,\mathbb{F}}$ with Σ_t (recall Definition 2.1) for a small tolerable penalty.

LEMMA 2.3. *Let d be a positive integer. Let $A \in \mathbb{Q}^{n\times n}$ be a matrix such that $A = \prod_{i=1}^{d} P_i$ for $P_i \in \mathbb{Q}^{n_i \times m_i}$, where $\sum_{i=1}^{d} \|P_i\|_0 \leq s$. Assume that for each i, $n_i \leq n^2$ and $m_i \leq n^2$. Then, for every $t \leq n^2/4$ such that $s \geq dt$ it holds that*

$$\Sigma_t(A) \leq 2^{2n^3 \cdot \left(e^d (2s/dt)^d\right)^t}.$$

PROOF. We follow the same steps as in the proof of Lemma 2.2, replacing the measure $\Gamma_{t,\mathbb{F}}(A)$ by $\Sigma_t(A)$. As before,

$$A_{i,j} = \left(\prod_{\ell=1}^{d} P_\ell\right)_{i,j} = \sum_{k_1,\ldots,k_{d-1}} (P_1)_{i,k_1} \cdot \left(\prod_{\ell=2}^{d-1} (P_\ell)_{k_{\ell-1},k_\ell}\right) \cdot (P_d)_{k_{d-1},j}\,.$$

Every element in $\mathsf{Prod}_t(A)$ can be written as

$$\sum_{\alpha\in\mathcal{M}} c_\alpha \cdot \alpha \tag{2.4}$$

where $\mathcal{M}$ is the set of monomials of degree dt in the entries of $P_1, P_2, \ldots, P_d$, and each c_α is a non-negative integer of absolute value at most $s^{dt} \leq 2^{n^3}$ (since $s \leq n^2 d$ and d is $O(1)$). It now follows that each element in $\Sigma_t(A)$ has the same form as in ((2.4)), with $c_\alpha \leq |\mathsf{Prod}_t(A)| \cdot 2^{n^3} \leq 2^{2n^3}$. We conclude that

$$\Sigma_t(A) \leq (2^{2n^3})^{\binom{s+dt}{dt}},$$

which implies the statement of the lemma using the same bounds on binomial coefficients as in Lemma 2.2. □

We now move on to describe constructions of matrices which have large Shoup-Smolensky dimension, and then deduce lower bounds for them.

2.4. Sidon sets and hard univariate matrices. In this section, we describe a construction of a matrix $G \in \mathbb{F}[y]^{n\times n}$ which has a large value of $\Gamma_{t,\mathbb{F}}$. Let us denote $G_{i,j} = y^{e_{i,j}}$ for some non-negative integer $e_{i,j}$. For G to have a large Shoup-Smolensky dimension of order t, the set $S = \{e_{1,1}, e_{1,2}, \ldots, e_{n,n}\} \subseteq \mathbb{N}$ should have the property that $S^{(t)} := \{a_1 + a_2 + \cdots + a_t : a_i \in S \text{ distinct}\}$ has size comparable to $\binom{|S|}{t}$. A set S such that every subset of size t of S has a distinct sum is called a *t-wise Sidon set*. These are very well studied objects in arithmetic combinatorics, and explicit constructions are known for them in $\mathsf{poly}(n)$ time (e.g., Lemma 60 in Bshouty (2014)). However, another important parameter in the construction of G is the degree of y. A t-Sidon set will inevitably contain integers of size roughly $n^{\Omega(t)}$, which implies that even if the Sidon set itself can be constructed in polynomial time, the construction of G itself would still take time which is not polynomially bounded in n. Below we give an elementary construction of such a set in time $n^{O(t)}$, which is similar to a related construction given in Agrawal *et al.* (2015).

LEMMA 2.5. *For every integers $t \le m$ there is a set $S \subseteq \mathbb{N}$ of size m such that:*

(i) $S^{(t)} := \{a_1 + a_2 + \cdots + a_t : a_i \in S \text{ distinct}\}$ *has size* $\binom{m}{t}$.

(ii) The maximal element in S is at most $m^{O(t)}$.

(iii) S can be constructed in time $m^{O(t)}$.

PROOF. Let $S' = \{1, 2, 2^2, \ldots, 2^{m-1}\}$. Clearly, every subset of S' has a distinct sum. For a prime p we denote $S_p = S' \bmod p = \{a \bmod p : a \in S'\}$, and we claim that there exists a prime $p \le m^{O(t)}$ such that $|(S_p)^{(t)}| = \binom{m}{t}$. Since this condition can be checked in time $m^{O(t)}$, this would immediately imply the statement of the lemma, by checking this condition for every $p \le m^{O(t)}$ and letting $S = S_p$ for a p which satisfies this condition.

For every subset $T \subseteq S'$ of size t, let σ_T denote the sum of its elements, and observe that $\sigma_T \le 2^m$. Clearly, $\sigma_T \bmod p = \sigma_{T'} \bmod p$ if and only if $p \mid \sigma_T - \sigma_{T'}$, so it is enough to show that there exists $p \le m^{O(t)}$ which does not divide

$$N := \prod_{\substack{T \neq T' \subseteq S' \\ |T|=|T'|=t}} (\sigma_T - \sigma_{T'}),$$

and therefore does not divide any of the terms on the right hand size. It further holds that $0 \neq N \leq (2^m)^{m^{O(t)}} = 2^{m^{O(t)}}$, so the existence of p now follows from the fact that N can have at most $\log N = m^{O(t)}$ distinct prime divisors, and from the prime number theorem. □

Given the above construction of t-wise Sidon sets, we now describe the construction of matrices with univariate polynomial entries which has large Shoup-Smolensky dimension.

CONSTRUCTION 2.6. *Let $S = \{e_{i,j} : i, j \in [n]\}$ be a t-wise Sidon set of positive integers of size n^2 as in Lemma 2.5. Then, the matrix $G_{t,n} \in \mathbb{F}[y]^{n \times n}$ is defined as $(G_t)_{i,j} = y^{e_{i,j}}$.*

The useful properties of Construction 2.6 are given by the following lemma.

LEMMA 2.7. *Let $t \leq n$ be a parameter, $S \subseteq N$ be a t-wise Sidon set of size n^2 and let $G_{t,n}$ be the matrix defined in Construction 2.6. Then, the following are true.*

(i) Every entry of $G_{t,n}$ is a monomial of degree at most $n^{O(t)}$.

(ii) $\Gamma_{t,\mathbb{F}}((G_{t,n})) \geq \binom{n^2}{t} \geq \left(\frac{n^2}{t}\right)^t$.

PROOF. The first item follows from the definition of $G_{t,n}$ and the properties of the set S in Lemma 2.5. The second item also follows from the properties of S and the definition of Shoup-Smolensky dimension, since every t-wise product of elements of $G_{t,n}$ gives a distinct monomial in y, and thus they are all linearly independent over the base field $\mathbb{F}$. □

2.5. Hard matrices over finite fields. From the univariate matrix in Construction 2.6, we now construct, for every p and parameter t, a matrix M over an extension of $\mathbb{F}_p$ which has large Shoup-Smolensky dimension over $\overline{\mathbb{F}}_p$ with the same parameters as $G_{t,n}$.

LEMMA 2.8. *Let p be a prime, and t be any positive integer. There is a matrix $M_{t,n} \in \mathbb{E}^{n\times n}$ over an extension $\mathbb{E}$ of $\mathbb{F}_p$ of degree $\exp(O(t \log n))$, which can be deterministically constructed in time $n^{O(t)}$, and satisfies*

$$\Gamma_{t,\mathbb{F}_p}(M_{t,n}) \geq \left(\frac{n^2}{t}\right)^t$$

PROOF. Let $G_{t,n}$ be as in Construction 2.6, and let Δ be the maximum degree of any entry of $G_{t,n}$. Set $D = 10 \cdot t \cdot \Delta = \exp(O(t \log n))$. We use Shoup's algorithm (see Theorem 3.2 in Shoup (1990)) to construct an irreducible polynomial $g(z)$ of degree $D+1$ over $\mathbb{F}_p$ in deterministic $\mathsf{poly}(D, |\mathbb{F}_p|)$ time. Let α be a root of $g(z)$ in an extension $\mathbb{E}$ of $\mathbb{F}_p$, where $\mathbb{E} \equiv \mathbb{F}_p[z]/\langle g(z)\rangle$.[8] Then, it follows that $1, \alpha, \alpha^2, \ldots, \alpha^D$ are linearly independent over $\mathbb{F}$.

The matrix $M_{t,n}$ is obtained from G_t by just replacing every occurrence of the variable y by α. We now need to argue that $M_{t,n}$ continues to satisfy $\Gamma_{t,\mathbb{F}_p}(M_{t,n}) \geq \left(\frac{n^2}{t}\right)^t$. By the choice of α, it immediately follows that $\Gamma_{t,\mathbb{F}_p}(M_{t,n}) = \Gamma_{t,\mathbb{F}_p}(G_{t,n})$, since every monomial in the set $\mathsf{Prod}_t(M_{t,n})$ is mapped to a distinct power of α in $\{0, 1, \ldots, D\}$, which are all linearly independent over $\mathbb{F}_p$.

The upper bound on the running time needed to construct $M_{t,n}$ now follows from the upper bound on the degree of the extension $\mathbb{E}$, and from Lemma 2.5. □

The following theorem now directly follows.

THEOREM 2.9. *Let p be any prime and $d \geq 2$ be a positive integer. Then, there exists a family of $n \times n$ matrices $\{A_n\}_{n\in\mathbb{N}}$ which can be constructed in time $n^{O(n^{1-1/2d})}$ such that every depth-d linear circuit $\overline{\mathbb{F}}_p$ computing A_n has size at least $\Omega(n^{1+1/2d})$. Moreover, the entries of A_n lie in an extension of $\mathbb{F}_p$ of degree at most $\exp(O(n^{1-1/2d} \log n))$.*

[8] We identify the elements of $\mathbb{E}$ with coefficient vectors of polynomials of degree at most D in $\mathbb{F}_p[z]$, and in this representation α is identified with the polynomial z.

PROOF. We invoke Lemma 2.8 with parameter t set to $n^{1-1/2d}$ to get matrices $\{A_n\}$ in time $n^{O(t)}$ with the following lower bound on their Shoup-Smolensky dimension.

$$\Gamma_{t,\mathbb{F}_p}(M_n) \geq \left(\frac{n^2}{t}\right)^t .$$

If there is a depth d linear circuit of size s computing the linear transformation $A_n \cdot \mathbf{x}$, the following inequality must hold (from Lemma 2.2),

$$\left(e^d(2s/dt)^d\right)^t \geq \left(\frac{n^2}{t}\right)^t . \tag{2.10}$$

If $s \leq n^{1+1/2d}/2$, we have,

$$\left(e^d(2s/dt)^d\right)^t \leq (O(e/d))^{dt} \cdot n^t .$$

We also have,

$$\left(\frac{n^2}{t}\right)^t \geq \left(n^{1+1/2d}\right)^t .$$

For any constant d, these estimates contradict (2.10), thereby implying a lower bound of $\Omega(n^{1+1/2d})$ on s. □

2.6. Hard matrices over $\mathbb{C}$. We now prove an analog for Lemma 2.8. We construct a matrix whose entries are positive integers that can be represented by at most $\exp(O(t \log n))$ bits, and give a lower bound for its Σ_t-measure (rather than $\Gamma_{t,\mathbb{F}}$ as before).

LEMMA 2.11. *Let t be any positive integer. There is a matrix $M_{t,n} \in \mathbb{Q}^{n\times n}$, which can be deterministically constructed in time $n^{O(t)}$, such that every entry of $M_{t,n}$ is an integer of bit complexity at most $\exp(O(t \log n))$, and it holds that*

$$\Sigma_t(M_{t,n}) \geq 2^{\left(\frac{n^2}{t}\right)^t} .$$

PROOF. Let $G_{t,n} \in \mathbb{F}[y]^{n\times n}$ be as in Construction 2.6. Define $M_{t,n} \in \mathbb{Q}^{n\times n}$ as

$$(M_{t,n})_{a,b} = (G_{t,n})_{a,b}(2),$$

that is, $(M_{t,n})_{a,b}$ is simply the polynomial $(G_{t,n})_{a,b}(y)$ evaluated at $y = 2$.

As in the proof of Lemma 2.7, each element in $\mathsf{Prod}_t(M_{t,n})$ is now a distinct power of 2, which implies that $\Sigma_t(M_{t,n}) = 2^{\binom{n^2}{t}}$.

The statement on the running time follows directly from Lemma 2.7. □

The analog of Theorem 2.9 for $\mathbb{C}$ is given below.

THEOREM 2.12. *There exists a family of matrices* $\{A_n\}_{n\in\mathbb{N}}$ *over* $\mathbb{Q}$ *which can be constructed in time* $n^{O(n^{1-1/2d})}$ *such that every depth-d linear circuit* $\mathbb{C}$ *computing* A_n *has size at least* $\Omega(n^{1+1/2d})$. *Moreover, the entries of* A_n *are positive integers of bit complexity at most* $\exp(O(n^{1-1/2d}\log n))$.

PROOF. Let $s = n^{1+1/2d}/2$ and $t = n^{1-1/2d}$ and let $A_n = M_{t,n}$, where $M_{t,n}$ is as in Lemma 2.11. A depth-d circuit for M_n implies a factorization $M_n = \prod_{i=1}^{d} P_i$, with $P_i \in \mathbb{C}^{n_i \times m_i}$, such that $\sum_{i=1}^{d} \|P_i\|_0 \leq s$. Observe that by removing, if necessary, zero columns or rows from $P_1, \ldots, P_d$ without affecting the product, we may assume $n_i, m_i \leq n^2$, as otherwise the lower bound trivially holds. By Lemma 2.3 and Lemma 2.11, this implies that

$$(n^2/t)^t \leq \log \Sigma_t(A_n) \leq 2n^3 \cdot \left(e^d(2s/t)^d\right)^t.$$

If $s \leq n^{1+1/2d}/2$, we have,

$$\left(e^d(2s/dt)^d\right)^t \leq (O(e/d))^{dt} \cdot n^t.$$

We also have

$$\left(\frac{n^2}{t}\right)^t \geq \left(n^{1+1/2d}\right)^t.$$

For any constant d, these estimates contradict the inequality above, thus implying a lower bound of $\Omega(n^{1+1/2d})$ on s.

The statement on the running time for constructing A_n follows again from Lemma 2.11. □

2.7. Lower bounds for depth-2 linear circuits. The lower bounds of Theorem 2.9 and Theorem 2.12 apply to any constant depth. However, here we briefly remark that in the special case of $d = 2$ there is in fact a much simpler construction. As discussed in the introduction, for depth-2 linear circuits, the best lower bounds currently known is a lower bound of $\Omega\left(n\frac{\log^2 n}{\log\log n}\right)$ based on the study of super-concentrator graphs in the work of Radhakrishnan & Ta-Shma (2000). We now discuss two constructions of matrices in quasi-polynomial time which improve upon this bound. More formally, we prove the following theorem.

THEOREM 2.13. *Let c be any positive constant. Then, there is a family $\{A_n\}_{n\in\mathbb{N}}$ of $n\times n$ matrices with entries in $\mathbb{N}$ of bit complexity at most $\exp(O(\log^{2c+1} n))$ such that A_n can be constructed in time $\exp(O(\log^{2c+1} n))$ and any depth-2 linear circuit over $\mathbb{C}$ computing A_n has size at least $\Omega(n\log^c n)$.*

The first construction directly follows from Lemma 2.11 when invoked with $t = 10\cdot\log^{2c} n$. Once we have the matrices guaranteed by Lemma 2.11, we just follow the proof of Theorem 2.12 as is by taking $d = 2$ and $t = 10\log^{2c} n$. We skip the technical details and now discuss the second construction, which is based on the following observation, first proved by Shoup & Smolensky (1996).

OBSERVATION 2.14. *Let $\{A_n\}_{n\in\mathbb{N}}$ be a family of matrices where $(A_n)_{i,j} = 2^{2^{(n+1)(i-1)+j}}$. Then, any depth$-2$ linear circuit computing A_n has size $\Omega(n^2)$.*

PROOF. The key to the proof is to observe that for $t = n^2/4$, $\Sigma_t(A_n) \geq 2^{\binom{n^2}{n^2/4}} \geq 2^{2^{n^2/2}}$. This follows from the fact that each t wise product of the entries of A_n is a power of 2 where the exponent is a sum of powers of 2 and for any two distinct degree t multilinear monomials in the entries of A_n, the set of powers of 2 that appear in the exponent are distinct. On the other hand, from Lemma 2.3, we know that if A_n can be computed by a depth-2 linear circuit of size at most s, then

$$\Sigma_t(A_n) \leq 2^{2n^3\left(e^2(4s/n^2)\right)^{n^2/4}}.$$

Now, for $s \leq n^2/100$, this upper bound is much smaller than the lower bound of $2^{2^{n^2/2}}$. Thus, any depth-2 linear circuit for A_n over $\mathbb{C}$ has size at least $n^2/100$. □

If we directly use this observation to construct hard matrices, the bit complexity of the entries of A_n (and hence the time complexity of constructing A_n) is as large as $2^{\Theta(n^2)}$. However, it also gives a much stronger (quadratic) lower bound on the depth-2 linear circuit size for A_n than what is promised in Theorem 2.13. For our second construction for hard matrices for Theorem 2.13, we invoke Observation 2.14 to construct *small* hard matrices (thus saving on the running time) and then construct a larger block diagonal matrix by taking a Kronecker product of this small hard matrix with a large identity matrix. The following lemma then guarantees a non-trivial lower bound on the size of any depth-2 linear circuit computing this larger block diagonal matrix.

Lemma 2.15. *Let A be an $k \times k$ matrix, such that any depth-2 linear circuit computing A has size at least s. Let B be an $mk \times mk$ matrix defined as $B = \mathbf{I}_m \otimes A$, where $\otimes$ denotes the Kronecker product, and $\mathbf{I}_m$ the $m \times m$ identity matrix. Then, any depth-2 linear circuit computing B has size at least $m \cdot s$.*

Proof. A depth-2 linear circuit for B gives a factorization of B as $P \cdot Q$ for an $mk \times r$ matrix P and an $r \times mk$ matrix Q for some parameter r. We partition the rows of P into m contiguous blocks of size k each, and let P_i be the $k \times r$ submatrix which consists of the i^{th} block (i.e. rows $(i-1)k+1, (i-1)k+2, \ldots, ik$ of P). Similarly, we partition the columns of Q into m contiguous blocks of size k each and let Q_i be the $r \times k$ submatrix of Q corresponding to the i^{th} block. From the structure of B, it follows that for every $i \in \{1, 2, \ldots, m\}$, $P_i \cdot Q_i = A$. From the lower bound on the size of any depth-2 linear circuit for A, we get that $\|P_i\|_0 + \|Q_i\|_0 \geq s$. Combining this lower bound for $i = 1, 2, \ldots, m$, we get $\|P\|_0 + \|Q\|_0 = \sum_{i=1}^{m} (\|P_i\|_0 + \|Q_i\|_0) \geq m \cdot s$. □

We now note that Observation 2.14 and Lemma 2.15 imply another family of matrices for which Theorem 2.13 holds.

PROOF (Second proof of Theorem 2.13). Pick $k = \Theta(\log^c n)$ such that k divdes n, and let M_k be the matrix defined as $(M_k)_{i,j} = 2^{2^{(k+1)(i-1)+j}}$. Let $A_n = \mathbf{I}_{n/k} \otimes M_k$. Clearly, A_n can be constructed in time $2^{O(k^2)}$. Moreover, from Observation 2.14 and Lemma 2.15 it follows that any depth-2 linear circuit computing A_n has size at least $\Omega(n/k \cdot k^2) = \Omega(n \log^c n)$. □

We note that even though the discussion in this section was confined to depth-2 linear circuit lower bounds over $\mathbb{C}$, similar ideas can be extended to other fields as well.

In light of the above construction, a natural question is to ask if this idea also extends to the construction of hard matrices for depth-d circuits for arbitrary constant d. While this is a reasonable conjecture, the easy proof of Lemma 2.15 breaks down even at depth 3.

There are some variations of this idea, such us looking at $\mathbf{J}_{n/k} \otimes M_k$, where $\mathbf{J}$ is the all-1 matrix, which would work equally well to prove a lower bound for depth-2 circuits, but for which it is possible to prove an $O(n)$ upper bound in depth-3.

Furthermore, it can be seen that upper bounds on matrix multiplication in bounded depth will give small linear circuits for computing $\mathbf{I}_{n/k} \otimes M_k$. Thus, improved lower bounds using this construction, even for depth-3 circuits, will require proving new lower bounds for matrix multiplication in bounded depth (the current best lower bounds are again barely super-linear (Raz & Shpilka 2003)).

3. Lower bounds via Hitting Sets

In this section, we prove lower bounds for several classes of depth 2 circuits using hitting sets for matrices. We first recall the definition.

DEFINITION 3.1 (Hitting set for matrices, Forbes & Shpilka 2012). *Let $\mathcal{C} \subseteq \mathbb{F}^{n \times n}$ be a set of matrices. A set $\mathcal{H} \subseteq \mathbb{F}^n \times \mathbb{F}^n$ is said to be a* hitting set *for $\mathcal{C}$, if for every non-zero $M \in \mathcal{C}$, there is a pair $(\mathbf{a}, \mathbf{b}) \in \mathcal{H}$ such that*

$$\langle \mathbf{a}, M \cdot \mathbf{b} \rangle = \sum_{i \in [n], j \in [n]} M_{i,j} a_i b_j \neq 0.$$

Every class $\mathcal{C}$ has a hitting set of size n^2, namely the indicator matrices of each of the entries. A hitting set is non-trivial if its size is at most $n^2 - 1$. Observe that a non-trivial hitting set for $\mathcal{C}$ gives an efficient algorithm for finding a matrix $M \notin \mathcal{C}$, by finding a non-zero A such that $\langle A, H \rangle = 0$ for every $H \in \mathcal{H}$. Such an A exists and can be found in polynomial time because the set $\mathcal{H}$ imposes at most $n^2 - 1$ homogeneous linear constraints on the n^2 entries of A. This argument is a special case of a more general theorem showing how efficient algorithms for black box polynomial identity testing give lower bounds for algebraic circuits (Agrawal 2005; Heintz & Schnorr 1980).

In practice, it is often convenient (although by no means necessary) to consider hitting sets that contain only rank 1 matrices $\mathbf{x}\mathbf{y}^T$, since $\langle A, \mathbf{x}\mathbf{y}^T \rangle = \mathbf{x}^T A \mathbf{y}$, and thus we find ourselves in the more familiar territory of polynomial identity testing, trying to construct a hitting set for the class of polynomials of the form $\mathbf{x}^T A \mathbf{y}$ for $A \in \mathcal{C}$. This approach was also taken by Forbes & Shpilka (2012), who considered this exact problem where $\mathcal{C}$ is the class of low-rank matrices, and remarked that hitting sets for the class of low-rank matrices plus sparse matrices will give an explicit construction of a rigid matrix.

As mentioned in Section 1.5, our proofs of Theorem 1.4 and Theorem 1.6 involve constructions of hitting sets for classes which correspond to restricted classes of depth-2 circuits. However, the following problem remains open.

Open Question 3.2. *For some $0 < \epsilon \leq 1$, construct an explicit hitting set of size at most $n^2 - 1$ for the class of $n \times n$ matrices A which can be written as $A = BC$ where B, C have at most $n^{1+\epsilon}$ non-zero entries.*

A solution to Open Question 3.2 will imply lower bounds of the form $n^{1+\varepsilon}$ for an explicit matrix. If $\epsilon = 1$, this will imply lower bounds for logarithmic depth linear circuits.

3.1. Matrices with no sparse vectors in their kernel. In this section, we recall some simple, deterministic and efficient constructions of matrices which do not have any sparse non-zero vector

in their kernel. Such a construction forms the basic building block for building hard instances of matrices for various cases of the matrix factorization problem that we discuss in the rest of this paper. We start by describing such a construction over the field of real numbers.

3.1.1. Construction over $\mathbb{R}$. The following is a weak form of a classical lemma of Descartes.

LEMMA 3.3 (Descartes' rule of signs, Anderson *et al.* 1998). *Let $d_1 < d_2 < \cdots < d_k$ be non-negative integers, and let $a_1, a_2, \ldots, a_k$ be arbitrary real numbers. Then, the number of distinct positive roots of the polynomial $\sum_{i=1}^{k} a_i x^{d_i}$ is at most $k-1$.*

Lemma 3.3 immediately gives the following construction of a small set of vectors, such that not all of them can lie in the kernel of any matrix with at least one sparse row.

LEMMA 3.4. *For $i \in [n]$, let $\mathbf{v}_i := (1, i, i^2, \ldots, i^{n-1}) \in \mathbb{R}^n$. Then, for every $1 \leq s \leq n$ and for every $m \times n$ matrix B over real numbers that has a non-zero row with at most s non-zero entries, there is an $i \in [s]$ such that $B \cdot \mathbf{v}_i \neq \mathbf{0}$.*

PROOF. Let $(a_0, a_1, \ldots, a_{n-1}) \in \mathbb{R}^n$ be any non-zero vector with at most s non zero entries. So, the polynomial $P(x) = \sum_{i=0}^{n-1} a_i x^i$ has sparsity at most s. From Lemma 3.3, it follows that P has at most $s-1$ positive real roots. Therefore, there exists an $i \in [s]$ such that i is *not* a root of $P(x)$, i.e., $P(i) \neq 0$. The lemma now follows immediately by taking $(a_0, a_1, \ldots, a_{n-1})$ to be any non-zero s-sparse row of B. □

We remark that Lemma 3.4 also holds for matrices over $\mathbb{C}$ which have a sparse non-zero row for the choice of the vectors v_i as above. This follows from the application of Lemma 3.3 separately for the real and complex parts of a sparse complex polynomial, both of which are individually sparse, with real coefficients and at least one of them is not identically zero. This observation extends our results over $\mathbb{R}$ in Section 3.2 to the field of complex numbers.

3.1.2. Construction over finite fields. We now recall some basic properties of Reed-Solomon codes, and observe they can be used as well in lieu of the construction in Lemma 3.4.

The proofs for these properties can be found in any standard reference on coding theory, e.g., Chapter 5 in Guruswami *et al.* (2018).

Definition 3.5 (Reed Solomon codes). *Let $\mathbb{F}_q$ be the finite field with q elements, which we denote $\{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$. Let $k \in \{0, 1, \ldots, q-1\}$. The Reed-Solomon code of block length q and dimension k is defined as follows.*

$$RS_q[q,k] = \{(P(\alpha_0), P(\alpha_1), \ldots, P(\alpha_{q-1})) : P(z) \in \mathbb{F}_q[z], \deg(P) \leq k-1\}.$$

Lemma 3.6. *Let $\mathbb{F}_q$ be the finite field with q elements and let $k \in \{0, 1, \ldots, q-1\}$. The linear space $RS_q[q,k]$ as in Definition 3.5 satisfies the following properties.*

- *Every non-zero vector in $RS_q[q,k]$ has at least $q-k+1$ non-zero coordinates.*
- *The dual of $RS_q[q,k]$ is the space of Reed Solomon codes of block length q and dimension $q-k$.*

Lemma 3.7. *Let $\mathbb{F}_q = \{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$ be the finite field with q elements. For any $k \leq q-1$, let G_k be the $q \times k$ matrix over $\mathbb{F}_q$ whose i-th row is $(1, \alpha_{i-1}, \alpha_{i-1}^2, \ldots, \alpha_{i-1}^{k-1})$. Then, every non-zero vector in $\mathbb{F}_q^q$ in the kernel of $(G_k)^T$ has at least $k+1$ non-zero coordinates.*

Proof. Observe that G_k is the precisely the generator matrix of Reed Solomon codes of block length q and dimension k over $\mathbb{F}_q$. In particular, the linear space $RS_q[q,k]$ as in Lemma 3.6 is spanned by the columns of G_k. Thus any vector $\mathbf{w}$ in the kernel of $(G_k)^T$ is in fact a codeword of the dual of these codes, which as we know from Item 2 of Lemma 3.6, is itself a Reed Solomon code of block length q and dimension $q-k$. From the first item of Lemma 3.6, it now follows that $\mathbf{w}$ has at least $k+1$ non-zero coordinates. □

The following lemma is an analog of Lemma 3.4.

LEMMA 3.8. *Let $\mathbb{F}_q = \{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$ be the finite field with q elements, $s \in [q]$ be a parameter and let $\mathbf{v}_i$ be the i-th column of the matrix G_k as in Lemma 3.7 for $k = s$.*

Then, for every $m \times n$ matrix B over $\mathbb{F}_q$ that has a non-zero row with at most s non zero entries, there is an $i \in [s]$ such that $B \cdot \mathbf{v}_i \neq 0$.

PROOF. The proof follows from the observation that any non-zero vector orthogonal to all the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s$ must be in the kernel of the matrix G_s^T and hence by Lemma 3.7 must have at least $s + 1$ non-zero entries. □

3.2. Lower bounds for symmetric circuits. We now prove our lower bounds for symmetric circuits. Recall that a symmetric circuit is a linear depth-2 circuit of the form $B^T B$.

THEOREM 3.9. *There is an explicit family of positive semidefinite matrices $\{M_n\}$ such that every symmetric circuit computing M_n has size at least $n^2/4$.*

For the proof of this theorem, we give an efficient deterministic construction of a hitting set $\mathcal{H}$ for the set of matrices which factor as $B^T \cdot B$ for B of sparsity less than $n^2/4$, and as outlined in Section 1.5, we construct a hard matrix $M = \tilde{M}^T \cdot \tilde{M}$ which is not hit by such a hitting set and has a high rank.

We start by describing the construction of M.

LEMMA 3.10. *Let $\{\mathbf{v}_i : i \in [n]\}$ be the set of vectors defined in Lemma 3.4. There exists an explicit PSD matrix $M \in \mathbb{R}^{n \times n}$ of rank $n/2$ such that $\mathbf{v}_i^T M \mathbf{v}_i = 0$ for $i \in [n/2]$.*

PROOF. We first wish to construct a matrix $\tilde{M}$ of high rank such that $\tilde{M}\mathbf{v}_i = 0$ for $i = 1, \ldots, n/2$. This can always be done in polynomial time by finding $n/2$ independent vectors in the vector space (of dimension $n/2$) which is orthogonal to all the vectors $\mathbf{v}_i$.

We now describe such a construction more explicitly. Let V be the Vandermonde matrix whose rows are the $\mathbf{v}_i$'s. Let $U = V^{-1}$, and for $j \in [n]$ let $\mathbf{u}_j$ denote the j-th column of U.

Let $\tilde{M}$ be an $n \times n$ matrix whose first $n/2$ rows are 0, and for $j \in \{n/2+1, \ldots, n\}$, the i-th row of $\tilde{M}$ is $\mathbf{u}_j^T$. Since $\langle \mathbf{v}_i \mathbf{u}_j \rangle$ equals 1 if $i = j$ and 0 otherwise, it now follows that $\tilde{M}\mathbf{v}_i = 0$ for $i = 1, \ldots, n/2$, and since $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are linearly independet, $\tilde{M}$ has rank $n/2$.

Now let $M = (\tilde{M}^T) \cdot \tilde{M}$, so that indeed M is a positive semi-definite matrix, and $\mathsf{rank}\, M = n/2$ as well. It immediately follows that

$$\mathbf{v}_i^T M \mathbf{v}_i = (\mathbf{v}_i^T \tilde{M}^T)(\tilde{M}\mathbf{v}_i) = 0,$$

as stated in the Lemma. □

We are now ready to prove Theorem 3.9.

Proof (Proof of Theorem 3.9). Let M be the matrix from Lemma 3.10. Let $B \in \mathbb{R}^{m \times n}$ be real matrix such that $\|B\|_0 < n^2/4$, and suppose towards contradiction that $M = B^T B$.

It follows that the rank of B must be at least $n/2$. Thus, B must have at least $n/2$ non-zero rows. Now, since the total sparsity of B is at most $n^2/4 - 1$, there must be a non-zero row of B with sparsity at most $(n^2/4 - 1)/(n/2) \leq n/2$. From Lemma 3.4, it follows that there is an $i \in [n/2]$ such that $B \cdot \mathbf{v}_i$ is non-zero. Thus, for this index i, we have that

$$\mathbf{v}_i^T (B^T B)\mathbf{v}_i = \|B\mathbf{v}_i\|_2^2 \neq 0,$$

contradicting Lemma 3.10. □

We remark that the proof of Theorem 3.9 goes through almost verbatim for symmetric circuits over $\mathbb{C}$ (recall that over $\mathbb{C}$ these are circuits of form B^*B, where B^* is the conjugate transpose of B).

3.3. Lower bounds for invertible circuits. Recall that an invertible circuit is a circuit of the form BC where either B or C is invertible. In this section, we prove Theorem 1.6, which shows a quadratic lower bound for such circuits. For convenience, we restate the theorem.

THEOREM 3.11. *There exists an explicit family of $n \times n$ matrices $\{A_n\}$, over any field $\mathbb{F}$ such that $\mathbb{F} \geq \mathsf{poly}(n)$, such that every invertible circuit computing A_n has size $n^2/4$.*

PROOF. We give a proof over the field of real numbers and highlight the ideas necessary to extend the argument to work over large enough finite fields.

Fix n, and let $M = \tilde{M}^T\tilde{M}$ be the matrix constructed in Lemma 3.10. Let B and C be $n \times n$ matrices over $\mathbb{R}$ such that $M = BC$. Suppose first that B is invertible and C has sparsity less than $n^2/4$.

Since $\mathsf{rank}(M) \geq n/2$, the same applies for $\mathsf{rank}(C)$, and hence the number of non-zero rows in C must be at least $n/2$. Thus, C must have a non-zero row with at most $(n^2/4 - 1)/(n/2) \leq n/2$ non-zero entries. Along with Lemma 3.4, this implies that there is an $i \in [n/2]$ such that $C \cdot \mathbf{v}_i \neq \mathbf{0}$, where $\mathbf{v}_i$ is as in Lemma 3.4. Since B is invertible, we get that $(B \cdot C \cdot \mathbf{v}_i)$ is a non-zero vector, so for some $j \in [n]$,

$$\mathbf{e}_j^T(BC)\mathbf{v}_i \neq 0.$$

However, as in the proof of Lemma 3.10

$$\mathbf{e}_j^T(M)\mathbf{v}_i = \mathbf{e}_j^T\tilde{M}^T\tilde{M}\mathbf{v}_i = 0,$$

since $\tilde{M}\mathbf{v}_i = 0$ for all $i \in [n/2]$.

The case that B is sparse and C is invertible is virtually the same, by considering $\mathbf{v}_i^T(BC)\mathbf{e}_j$, and replacing the argument on the rows of C by a similar one on the columns of B.

For the proof over finite fields, we replace every application of Lemma 3.4 by Lemma 3.8. Note that this requires the n-th matrix in the family to be defined over a field of size more than n. The rest of the argument essentially remains the same. □

Over fixed finite fields (for example, $\mathbb{F}_2$), it is possible to prove an analog of Theorem 3.11, with worse constants, by replacing the use of Reed-Solomon codes with any good explicit error-correcting code C of dimension αn and distance δn for some fixed constants $\alpha, \delta > 0$. The proof proceeds as above by finding a matrix $\tilde{M}$ of rank αn such that $M\mathbf{v} = 0$ for every $\mathbf{v} \in C^{\perp}$.

4. Open problems

An important problem that continues to remain open is to prove a lower bound of the form $\Omega(n^{1+\varepsilon})$ for some constant $\varepsilon > 0$ for the depth-2 complexity of an explicit matrix. Such a lower bound would follow from an explicit hitting set of size at most n^2-1 for the class of polynomials of the form $\mathbf{x}^T BC\mathbf{y}$ such that $\|B\|_0 + \|C\|_0 \leq n^{1+\varepsilon}$.

Another natural question here is to understand if this PIT based approach can be used for explicit constructions of rigid matrices, which improve the state of art. One concrete question in this direction would be to construct explicit hitting sets for the set of matrices which are *not* (r, s) rigid for $rs > \omega(n^2 \log(n/r))$. Using the techniques in this paper, it is possible to construct hitting sets of size $O(rs)$ for matrices which are not (r, s) rigid. But, this is non-trivial only when $rs \leq cn^2$ for some constant $c < 1$, which is a regime of parameters for which explicit construction of rigid matrices is already known. A sequence of recent results (Alman & Williams 2017; Dvir & Edelman 2019; Dvir & Liu 2020) showed that many natural candidates for rigid matrices that posses certain symmetries are in fact not as rigid as suspected. This approach might circumvent these obstacles by giving an explicit construction which is not ruled out by these results.

A lower bound of s on the size of depth d linear circuits computing the linear transformation $A\mathbf{x}$ implies a lower bound of $\Omega(s)$ for depth $\Omega(d)$ algebraic circuits computing the degree-2 polynomial $\mathbf{y}^T A\mathbf{x}$ (Baur & Strassen 1983; Kaltofen & Singer 1991) (so, we can convert lower bounds for circuits with n outputs to lower bounds for circuits with 1 output). A notable open problem in algebraic complexity, which is very related to this work, is to prove any super-linear lower bound for algebraic circuits of depth $O(\log n)$ computing a polynomial with constant total degree. We refer to Raz (2010) for a discussion on the importance of this problem.

Acknowledgements

We thank Swastik Kopparty for an insightful discussion on explicit construction of Sidon sets over finite fields. We also thank

Rohit Gurjar, Nutan Limaye, Srikanth Srinivasan and Joel Tropp for helpful discussions, and to the anonymous reviewers for many useful comments which have improved the presentation of this paper.

A part of this work was done while the first named author was at the semester on Lower Bounds in Computational Complexity at Simons Institute for the Theory of Computing, Berkeley, USA, and at the Department of Computer Science, University of Toronto, Canada; and while the second named author was at the Center for the Mathematics of Information, California Institute of Technology, USA.

A preliminary version of this paper has appeared in the proceedings of CCC 2020 (Kumar & Volk 2020).

References

Manindra Agrawal (2005). Proving Lower Bounds Via Pseudo-random Generators. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005)*, 92–105. URL https://doi.org/10.1007/11590156_6.

Manindra Agrawal, Rohit Gurjar, Arpita Korwar & Nitin Saxena (2015). Hitting-Sets for ROABP and Sum of Set-Multilinear Circuits. *SIAM J. Comput.* **44**(3), 669–697. URL https://doi.org/10.1137/140975103.

Manindra Agrawal & V. Vinay (2008). Arithmetic Circuits: A Chasm at Depth Four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, 67–75. URL https://doi.org/10.1109/FOCS.2008.32.

Josh Alman & Lijie Chen (2019). Efficient Construction of Rigid Matrices Using an NP Oracle. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2019)*, 1034–1055. IEEE Computer Society. URL https://doi.org/10.1109/FOCS.2019.00067.

Josh Alman & R. Ryan Williams (2017). Probabilistic rank and matrix rigidity. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC 2017)*, 641–652. ACM. URL https://doi.org/10.1145/3055399.3055484.

Noga Alon & Pavel Pudlák (1994). Superconcentrators of Depths 2 and 3; Odd Levels Help (Rarely). *J. Comput. Syst. Sci.* **48**(1), 194–202. URL https://doi.org/10.1016/S0022-0000(05)80027-3.

Bruce Anderson, Jeffrey Jackson & Meera Sitharam (1998). Descartes' rule of signs revisited. *Amer. Math. Monthly* **105**(5), 447–451. ISSN 0002-9890. URL https://doi.org/10.2307/3109807.

Walter Baur & Volker Strassen (1983). The Complexity of Partial Derivatives. *Theoretical Computer Science* **22**, 317–330. URL https://doi.org/10.1016/0304-3975(83)90110-X.

Avraham Ben-Aroya, Dean Doron & Amnon Ta-Shma (2017). An efficient reduction from two-source to non-malleable extractors: achieving near-logarithmic min-entropy. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC 2017)*, 1185–1194. ACM. URL https://doi.org/10.1145/3055399.3055423.

Amey Bhangale, Prahladh Harsha, Orr Paradise & Avishay Tal (2020). Rigid Matrices From Rectangular PCPs or: Hard Claims Have Complex Proofs. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2020)*, 858–869. URL https://doi.org/10.1109/FOCS46700.2020.00084.

Nader H. Bshouty (2014). Testers and their applications. In *Innovations in Theoretical Computer Science, ITCS'14, 2014*, 327–352. URL https://doi.org/10.1145/2554797.2554828.

Peter Bürgisser, Michael Clausen & Mohammad A. Shokrollahi (1997). *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag. URL https://doi.org/10.1007/978-3-662-03338-8.

Eshan Chattopadhyay & David Zuckerman (2019). Explicit two-source extractors and resilient functions. *Ann. of Math. (2)* **189**(3), 653–705. ISSN 0003-486X. URL https://doi.org/10.4007/annals.2019.189.3.1.

Bernard Chazelle (2001). *The discrepancy method - randomness and complexity*. Cambridge University Press. ISBN 978-0-521-00357-5. URL https://www.cs.princeton.edu/~chazelle/pubs/book.pdf.

Gil Cohen (2017). Towards optimal two-source extractors and Ramsey graphs. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC 2017)*, 1157–1170. ACM. URL https://doi.org/10.1145/3055399.3055429.

Danny Dolev, Cynthia Dwork, Nicholas Pippenger & Avi Wigderson (1983). Superconcentrators, Generalizers and Generalized Connectors with Limited Depth (Preliminary Version). In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC 1983)*, 42–51. ACM. URL https://doi.org/10.1145/800061.808731.

Zeev Dvir & Benjamin L. Edelman (2019). Matrix Rigidity and the Croot-Lev-Pach Lemma. *Theory Comput.* **15**, 1–7. URL https://doi.org/10.4086/toc.2019.v015a008.

Zeev Dvir, Alexander Golovnev & Omri Weinstein (2019). Static data structure lower bounds imply rigidity. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC 2019)*, 967–978. ACM. URL https://doi.org/10.1145/3313276.3316348.

Zeev Dvir & Allen Liu (2020). Fourier and Circulant Matrices are Not Rigid. *Theory Comput.* **16**, 1–48. URL https://doi.org/10.4086/toc.2020.v016a020.

Michael A. Forbes & Amir Shpilka (2012). On identity testing of tensors, low-rank recovery and compressed sensing. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, Howard J. Karloff & Toniann Pitassi, editors, 163–172. ACM. URL https://doi.org/10.1145/2213977.2213995.

Michael L. Fredman (1982). The Complexity of Maintaining an Array and Computing Its Partial Sums. *J. ACM* **29**(1), 250–260. URL https://doi.org/10.1145/322290.322305.

Michael L. Fredman & Michael E. Saks (1989). The Cell Probe Complexity of Dynamic Data Structures. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC 1989)*, David S. Johnson, editor, 345–354. ACM. URL https://doi.org/10.1145/73007.73040.

Joel Friedman (1993). A note on matrix rigidity. *Combinatorica* **13**(2), 235–239. URL https://doi.org/10.1007/BF01303207.

Anna Gál, Kristoffer Arnsfelt Hansen, Michal Koucký, Pavel Pudlák & Emanuele Viola (2013). Tight Bounds on Computing Error-Correcting Codes by Bounded-Depth Circuits With Arbitrary Gates. *IEEE Trans. Information Theory* **59**(10), 6611–6627. URL https://doi.org/10.1109/TIT.2013.2270275.

Ankit Gupta, Pritish Kamath, Neeraj Kayal & Ramprasad Saptharishi (2016). Arithmetic Circuits: A Chasm at Depth 3. *SIAM J. Comput.* **45**(3), 1064–1079. URL https://doi.org/10.1137/140957123.

Venkatesan Guruswami, Atri Rudra & Madhu Sudan (2018). Essential Coding Theory. URL https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/.

Joos Heintz & Claus-Peter Schnorr (1980). Testing Polynomials which Are Easy to Compute (Extended Abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC 1980)*, 262–272. URL https://doi.org/10.1145/800141.804674.

Stasys Jukna & Igor Sergeev (2013). Complexity of Linear Boolean Operators. *Foundations and Trends in Theoretical Computer Science* **9**(1), 1–123. ISSN 1551-305X. URL https://doi.org/10.1561/0400000063.

Erich Kaltofen & Michael F. Singer (1991). Size efficient parallel algebraic circuits for partial derivatives. In *IV International Conference on Computer Algebra in Physical Research*, 133–145. URL https://users.cs.duke.edu/~elk27/bibliography/91/KaSi91.pdf.

Pascal Koiran (2012). Arithmetic Circuits: The Chasm at Depth Four Gets Wider. *Theoretical Computer Science* **448**, 56–65. URL https://doi.org/10.1016/j.tcs.2012.03.041.

Mrinal Kumar & Ben Lee Volk (2020). Lower Bounds for Matrix Factorization. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC 2020)*, volume 169, 5:1–5:20. URL https://doi.org/10.4230/LIPIcs.CCC.2020.5.

Kasper Green Larsen (2012). The cell probe complexity of dynamic range counting. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, 85–94. ACM. URL https://doi.org/10.1145/2213977.2213987.

Kasper Green Larsen (2014). On Range Searching in the Group Model and Combinatorial Discrepancy. *SIAM J. Comput.* **43**(2), 673–686. URL https://doi.org/10.1137/120865240.

Kasper Green Larsen, Omri Weinstein & Huacheng Yu (2018). Crossing the logarithmic barrier for dynamic Boolean data structure lower bounds. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, 978–989. ACM. URL https://doi.org/10.1145/3188745.3188790.

Daniel D. Lee & H. Sebastian Seung (2000). Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000*, 556–562. MIT Press. URL http://proceedings.neurips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.

Xin Li (2019). Non-Malleable Extractors and Non-Malleable Codes: Partially Optimal Constructions. In *Proceedings of the 34th Annual Computational Complexity Conference (CCC 2019)*, volume 137, 28:1–28:49. URL https://doi.org/10.4230/LIPIcs.CCC.2019.28.

Satyanarayana V. Lokam (2009). Complexity Lower Bounds using Linear Algebra. *Foundations and Trends in Theoretical Computer Science* **4**(1-2), 1–155. URL https://doi.org/10.1561/0400000011.

Julien Mairal, Francis R. Bach, Jean Ponce & Guillermo Sapiro (2009). Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, volume 382 of *ACM International Conference Proceeding Series*, 689–696. ACM. URL https://doi.org/10.1145/1553374.1553463.

Jacques Morgenstern (1973). Note on a Lower Bound on the Linear Complexity of the Fast Fourier Transform. *J. ACM* **20**(2), 305–306. URL https://doi.org/10.1145/321752.321761.

Behnam Neyshabur & Rina Panigrahy (2013). Sparse Matrix Factorization. *CoRR* **abs/1311.3315**. URL http://arxiv.org/abs/1311.3315.

Mihai Pătraşcu (2007). Lower bounds for 2-dimensional range counting. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC 2007)*, 40–46. ACM. URL https://doi.org/10.1145/1250790.1250797.

Mihai Pătraşcu & Erik D. Demaine (2006). Logarithmic Lower Bounds in the Cell-Probe Model. *SIAM J. Comput.* **35**(4), 932–963. URL https://doi.org/10.1137/S0097539705447256.

Nicholas Pippenger (1977). Superconcentrators. *SIAM J. Comput.* **6**(2), 298–304. URL https://doi.org/10.1137/0206022.

Nicholas Pippenger (1982). Superconcentrators of Depth 2. *J. Comput. Syst. Sci.* **24**(1), 82–90. URL https://doi.org/10.1016/0022-0000(82)90056-3.

Pavel Pudlák (1994). Communication in Bounded Depth Circuits. *Combinatorica* **14**(2), 203–216. URL https://doi.org/10.1007/BF01215351.

Pavel Pudlák (2000). A note on the use of determinant for proving lower bounds on the size of linear circuits. *Inf. Process. Lett.* **74**(5-6), 197–201. URL https://doi.org/10.1016/S0020-0190(00)00058-2.

Jaikumar Radhakrishnan & Amnon Ta-Shma (2000). Bounds for Dispersers, Extractors, and Depth-Two Superconcentrators. *SIAM J. Discrete Math.* **13**(1), 2–24. URL https://doi.org/10.1137/S0895480197329508.

Ran Raz (2010). Elusive Functions and Lower Bounds for Arithmetic Circuits. *Theory of Computing* **6**(1), 135–177. URL https://doi.org/10.4086/toc.2010.v006a007.

Ran Raz & Amir Shpilka (2003). Lower Bounds for Matrix Product in Bounded Depth Circuits with Arbitrary Gates. *SIAM J. Comput.* **32**(2), 488–513. URL https://doi.org/10.1137/S009753970138462X.

Mohammad Amin Shokrollahi, Daniel A. Spielman & Volker Stemann (1997). A Remark on Matrix Rigidity. *Inf. Process. Lett.* **64**(6), 283–285. URL https://doi.org/10.1016/S0020-0190(97)00190-7.

Victor Shoup (1990). New Algorithms for Finding Irreducible Polynomials over Finite Fields. *Mathematics of Computation* **54**, 435–447. URL https://www.ams.org/journals/mcom/1990-54-189/S0025-5718-1990-0993933-0/S0025-5718-1990-0993933-0.pdf.

Victor Shoup & Roman Smolensky (1996). Lower bounds for polynomial evaluation and interpolation problems. *Computational Complexity* **6**(4), 301–311. URL https://doi.org/10.1007/BF01270384.

Amir Shpilka & Amir Yehudayoff (2010). Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science* **5**, 207–388. ISSN 1551-305X. URL https://doi.org/10.1561/0400000039.

Volker Strassen (1973). Die Berechnungskomplexität Von Elementarsymmetrischen Funktionen Und Von Interpolationskoeffizienten. *Numerische Mathematik* **20**(3), 238–251. ISSN 0029-599X. URL https://doi.org/10.1007/BF01436566.

Sébastien Tavenas (2015). Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.* **240**, 2–11. URL https://doi.org/10.1016/j.ic.2014.09.004. Preliminary version in the *38th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2013)*.

Leslie G. Valiant (1975). On Non-linear Lower Bounds in Computational Complexity. In *Proceedings of the 7th Annual ACM Symposium on Theory of Computing (STOC 1975)*, 45–53. ACM. URL http://doi.acm.org/10.1145/800116.803752.

Leslie G. Valiant (1977). Graph-Theoretic Arguments in Low-Level Complexity. In *Proceedings of the 2nd Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1977)*, volume 53 of *Lecture Notes in Computer Science*, 162–176. Springer. URL https://doi.org/10.1007/3-540-08353-7_135.

Manuscript received July 31 2020

Ben Lee Volk
Department of Computer Science
University of Texas at Austin
Austin, USA
benleevolk@gmail.com

Mrinal Kumar
Department of Computer Science and Engineering
IIT Bombay
Mumbai, India
mrinal@cse.iitb.ac.in