Lower Bounds for Matrix Factorization

Mrinal Kumar[1]
Department of Computer Science and Engineering, IIT Bombay, India
mrinalkumar08@gmail.com

Ben Lee Volk
Center for the Mathematics of Information, California Institute of Technology, Pasadena, CA, USA
benleevolk@gmail.com

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 $n^{1+\Omega(1/d)}$. This improves upon the prior best lower bounds for this problem, which are barely super-linear, and were obtained by a long line of research based on the study of super-concentrators (albeit at the cost of a blow up in the time required to construct these matrices).

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.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Circuit complexity

Keywords and phrases Algebraic Complexity, Linear Circuits, Matrix Factorization, Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.CCC.2020.5

Related Version A full version of the paper is avaiable at `https://arxiv.org/abs/1904.01182`.

Acknowledgements A part of this work was done while the first 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. 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.

1 Introduction

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

▶ **Open Problem 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] A part of this work was done during 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.

35th Computational Complexity Conference (CCC 2020).
Editor: Shubhangi Saraf; Article No. 5; pp. 5:1–5:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 [50].

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 [10], Theorem 13.1; eliminating division gates requires that the field $\mathbb{F}$ in question is large enough. In this paper we will always make this assumption 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 poly(n) time, although more or less stringent definitions can be considered as well.

Despite decades of research and partial results, such lower bounds are not known.[2] 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 Subsection 1.5, 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 Subsection 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 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.

[2] We remark that super-linear lower bounds for general arithmetic circuits are known, but for polynomials of high degree [51, 7].

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 [53]: 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 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 [53, 41, 42, 14, 43, 6, 45].

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. 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 [22].

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 [54] 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 [43] 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 $(r, n^2/r \cdot \log(n/r))$ [21, 47], which is insufficient for proving lower bounds.

Observe that a decomposition $A = R + S$ where $\mathrm{rank}(R) = \varepsilon n$ 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

[3] In fact, one can obtain slightly better parameters. See, for example, [54] or [16].

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 $\Omega(n^2)$ lower bound for an explicit matrix, will imply a lower bound for depth $O(\log n)$.[4] This can be seen as the linear circuit analog of similar strong depth reduction theorems for general algebraic circuits [3, 29, 52, 24].

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. Indeed, 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 a constant $\delta > 0$, 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 remark that the argument above is very similar to the aforementioned result of Pudlák [43]; 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 Data structure lower bounds

The problem of matrix factorization into sparse matrices also appears in the context of proving lower bounds for 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 worst case update time of the database is the maximal number of non-zero elements in a column of U, and the worst case query time is the maximal number of non-zero elements in a row of Q. The value s denotes the space required by the data structure. It now directly follows that 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 worst case query

[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 [27]. 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.

or update time. It is also possible to define an analogous average case notion. Lower bounds for this model were proved by [19, 20, 40, 39, 30, 31, 32], 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, Golovnev and Weinstein [16] showed that proving lower bounds for this model is related to the problem of matrix rigidity from Subsection 1.2.

Despite the overall similarity, there are several key technical differences between the linear circuit complexity and the data structure problems. The first and obvious issue is that worst-case lower bounds on the update or query time do not necessarily imply that Q or U are dense matrices: the total sparsity of Q and U is related to the average-case update and query time. The second, more severe issue, is that in many applications 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 $n \operatorname{polylog}(n)$ is meaningless in that setting.

This issue also comes up when studying the so-called *succinct space* setting, where we require $s = n(1 + o(1))$. The lower bounds we are aware of for this setting are worst case lower bounds, and require the number of outputs q to be at least Cn for some $C > 1$ [23, 16], so that in the corresponding circuit the number of vertices in the middle layer is required to be much smaller than the number of outputs, which may be considered quite unnatural. In particular, we are unaware of any improved lower bounds on the sparsity of matrix factorization for $A \in \mathbb{F}^{n\times n}$ when $s = n(1 + o(1))$ or even $s = n$ which come from the data structure lower bounds literature.

1.4 Machine learning

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 [33][5] or sparse dictionary learning [36], and there are also connections to the analysis of deep neural networks [38].

1.5 Previous work

As mentioned in Subsection 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 and Smolensky [49] 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

[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 Subsection 1.5

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 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 [46]. 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 [50]) 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 ingenious proof [37] 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 [44] 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 [12] 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 [11, 8, 13, 34], 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 and some other related results, see the survey by Lokam [35].

1.6 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 2.** *Let $\mathbb{F}$ be a field. There exists a family of matrices $\{A_n\}_{n\in\mathbb{N}}$, 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 $\mathbb{F}$, has size at least $n^{1+\Omega(1/d)}$.*

If $\mathbb{F} = \mathbb{Q}$, the entries of A are integers of bit complexity $\exp(n^{1-\Omega(1/d)})$. If $\mathbb{F} = \mathbb{F}_q$ is a finite field, the entries of A are elements of an extension $\mathbb{E}$ of $\mathbb{F}$ of degree $\exp(n^{1-\Omega(1/d)})$.

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, Golovnev and Weinstein [16] 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 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 [43].

In a related and independent work, Alman and Chen [4] 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. We refer to [4] for some further discussion on the differences and similarities.

While the statement in Theorem 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 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 [45]). 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 Subsection 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 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 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 5 will imply, using Theorem 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 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 6 crucially requires the hard matrix A to be non-invertible.

1.7 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.

[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 $L(UP)$.

Shoup-Smolensky dimension

For the proof of Theorem 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 8 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 and Smolensky [49] 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 $\mathrm{poly}(\log n)$. Informally, as the order t gets larger, this measure becomes useful against stronger models of computation; however, it 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 [46] 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 Subsection 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 4 and Theorem 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 $\langle A, H \rangle := \sum_{i,j} A_{i,j} H_{i,j} \neq 0$.

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 [1, 26].

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 $\left\langle A, \mathbf{x}\mathbf{y}^T \right\rangle = \mathbf{x}^T A \mathbf{y}$, 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 and Shpilka [18], 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.

We carry out this idea for two different classes in the proofs of Theorem 4 and Theorem 6. However, the following problem remains open.

▶ **Open Problem 7.** *For some $0 < \varepsilon \leq 1$, 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 $n^{1+\varepsilon}$ non-zero entries.*

A solution to Open Problem 7 will imply lower bounds of the form $n^{1+\varepsilon}$ for an explicit matrix. If $\varepsilon = 1$, this will imply lower bounds for logarithmic depth linear circuits.

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 Subsection 3.1 for details.

2 Lower bounds for constant depth linear circuits

In this section, we prove Theorem 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 8** (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 $\Pi_t(S)$ the set of t-wise products of distinct entries of S that is,*

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

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 $\Pi_t(S)$.*

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

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.

▶ **Lemma 9.** *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 $\sum_{i=1}^{d} \|P_i\|_0 \leq s$. Then, for every $t \leq n^2/4$ such that $s \geq dt$ it holds that*

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

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 $\Pi_t(A)$ is a sum of monomials of degree dt in the entries of $P_1, P_2, \ldots, P_d$, that is,

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

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

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

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 8) for a small tolerable penalty.

▶ **Lemma 10.** *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 9, 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 $\Pi_t(A)$ can be written as

$$\sum_{\alpha \in \mathcal{M}} c_\alpha \cdot \alpha \qquad (1)$$

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 (1), with $c_\alpha \leq |\Pi_t(A)| \cdot 2^{n^3} \leq 2^{2n^3}$. 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 9. ◀

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 $S = \{e_{1,1}, e_{1,2}, \dots, e_{n,n}\} \subseteq \mathbb{N}$ should have the property that $S^{(t)} := \{a_1 + a_2 + \dots + a_t : a_i \in S \text{ distinct}\}$ 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 poly(n) time (e.g., Lemma 60 in [9]). However, another important parameter in the construction is the degree of y, and such a set will inevitably contain integers of size roughly $n^{\Omega(t)}$. Thus, the construction of G would take time which is not polynomially bounded in n. Below we give an elementary construction of such a set in time $n^{O(t)}$ (cf. [2]).

▶ **Lemma 11.** *Let t be a positive integer. There is a set $S \subseteq \mathbb{N}$ of size m such that:*

1. *$S^{(t)} := \{a_1 + a_2 + \dots + a_t : a_i \in S \text{ distinct}\}$ has size $\binom{m}{t}$.*
2. *The maximal element in S is at most $m^{O(t)}$.*
3. *S can be constructed in time $m^{O(t)}$.*

Proof. Let $S' = \{1, 2, 2^2, \dots, 2^{m-1}\}$. 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 $0 \neq N \le (2^m)^{m^{O(t)}} = 2^{m^{O(t)}}$, 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 12.** *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 11. Then, the matrix $G_{t,n} \in \mathbb{F}[y]^{n\times n}$ is defined as follows as $(G_t)_{i,j} = y^{e_{i,j}}$.*

The useful properties of Construction 12 are given by the following lemma.

▶ **Lemma 13.** *Let $t \le n$ 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 12. Then, the following are true.*

1. *Every entry of $G_{t,n}$ is a monomial of degree at most $n^{O(t)}$.*
2. *$\Gamma_{t,\mathbb{F}}((G_{t,n})) \ge \binom{n^2}{t} \ge \left(\frac{n^2}{t}\right)^t$.*

Proof. The first item follows from the definition of $G_{t,n}$ and the properties of the set S in Lemma 11. 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 12, 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 14.** *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 $\exp\left(O(t\log n)\right)$, 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 12, and let Δ be the maximum degree of any entry of $G_{t,n}$. Set $D = 10 \cdot t \cdot \Delta = \exp\left(O(t\log n)\right)$. We use Shoup's algorithm (see Theorem 3.2 in [48]) to construct an irreducible polynomial $g(z)$ of degree $D+1$ over $\mathbb{F}_p$ in deterministic $\mathrm{poly}(D, |\mathbb{F}_p|)$ 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 $\Pi_t(M_{t,n})$ 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 11. ◀

The following theorem now directly follows.

▶ **Theorem 15.** *Let p be any prime and $d \geq 2$ be a positive integer. Then, there exists a family of 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 $\exp(O(n^{1-1/2d}\log n))$.*

Proof. We invoke Lemma 14 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 9),

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

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

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

[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.

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 Equation 2, thereby implying a lower bound of $\Omega(n^{1+1/2d})$ on s. ◀

2.6 Hard matrices over $\mathbb{C}$

An analog for Lemma 14 can be proved over $\mathbb{C}$ by constructing a matrix whose Σ_t-measure (rather than $\Gamma_{t,\mathbb{F}}$ as before) is large. The full statement and its proof appear in the full version of the paper. The analog of Theorem 15 for $\mathbb{C}$ is given below, with the proof again deferred to the full version.

▶ **Theorem 16.** *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 $\exp(O(n^{1-1/2d}\log n))$.*

2.7 Lower bounds for depth-2 linear circuits

The lower bounds of Theorem 16 and Theorem 15 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 and Ta-Shma [45].

In the full version of the paper, we give two simple constructions of matrices in quasi-polynomial time which improve upon this bound.

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 17** (Hitting set for matrices, [18]). *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*

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

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 18** (Descartes' rule of signs). *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 18 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 19.** *For $i \in [n]$, let $\mathbf{v}_i := \left(1, i, i^2, \ldots, i^{n-1}\right) \in \mathbb{R}^n$. 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 18, 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 19 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 18 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 Subsection 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 19.

The proofs for these properties can be found in any standard reference on coding theory, e.g., Chapter 5 in [25].

▶ **Definition 20** (Reed Solomon codes). *Let $\mathbb{F}_q = \{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$ be the finite field with q elements and let $k \in \{0, 1, \ldots, q-1\}$. The Reed-Solomon code of block length q and dimension k are 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 21.** *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 20 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 22.** *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 21 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 21, is itself a Reed Solomon code of block length q and dimension $q - k$. From the first item of Lemma 21, it now follows that $\mathbf{w}$ has at least $k + 1$ non-zero coordinates. ◀

The following lemma is an analog of Lemma 19.

▶ **Lemma 23.** *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 22 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 $v_1, v_2, \ldots, v_s$ must be in the kernel of the matrix G_s^T and hence by Lemma 22 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 24.** *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 Subsection 1.7, 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 25.** *Let $\{\mathbf{v}_i : i \in [n]\}$ be the set of vectors defined in Lemma 19. There exists an explicit PSD matrix M of rank $n/2$ such that $\mathbf{v}_i^T M \mathbf{v}_i = 0$ for $i \in [n/2]$.*

Proof. We wish to find a matrix $\tilde{M}$ of high rank such that $\tilde{M}\mathbf{v}_i = 0$ for $i = 1, \ldots, n/2$. This can be done by completing $\{\mathbf{v}_i : i \in \{1, 2, \ldots, n/2\}\}$ to a basis (in an arbitrary way) and requiring that the other $n/2$ basis elements are mapped to linearly independent vectors under $\tilde{M}$. Conveniently, the set $\{\mathbf{v}_i : i \in [n]\}$ is itself a basis for $\mathbb{R}^n$: the matrix V whose rows are the $\mathbf{v}_i$'s is a Vandermonde matrix.

We now describe this in some more detail. For $i \in [n]$, let $\mathbf{e}_i$ by the i-th elementary basis vector. For a set of n^2 variables $Y = (y_{i,j})_{n \times n}$ consider the system of (non-homogeneous) linear equations on the variables Y given by the n constraints.

$$\begin{aligned} &Y \cdot \mathbf{v}_i = 0 \quad \text{for } i \in \{1, 2, \ldots, n/2\} \\ &Y \cdot \mathbf{v}_i = \mathbf{e}_i \quad \text{for } i \in \{n/2 + 1, \ldots, n\}. \end{aligned}$$

Since the vectors $\{\mathbf{v}_i : i \in [n]\}$ are linearly independent, this system has a solution, which can be found in polynomial time using basic linear algebra. More explicitly the j-th row of Y, $\mathbf{y}_j$, is given by the solution to the linear system $V \cdot (\mathbf{y}_j)^T = 0$ for $1 \leq j \leq n/2$ and $V \cdot (\mathbf{y}_j)^T = \mathbf{e}_j$ for $n/2 + 1 \leq j \leq n$ where V is the Vandermonde matrix whose rows are the $\mathbf{v}_i$'s. Let $\tilde{M}$ be the matrix whose rows are the solution to the system above. Also, note that the rank of $\tilde{M}$ is at least $n/2$, as linearly independent vectors $\mathbf{e}_{n/2+1}, \mathbf{e}_{n/2+2}, \ldots, \mathbf{e}_n$ are in the image of the linear transformation given by $\tilde{M}$.

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

$$\mathbf{v}_i^T M \mathbf{v}_i = (\mathbf{v}_i^T \tilde{M^T})(\tilde{M}\mathbf{v}_i) = 0. \qquad \blacktriangleleft$$

We are now ready to prove Theorem 24.

Proof of Theorem 24. Let M be the matrix from Lemma 25. 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 19, 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 25. ◀

We remark that the proof of Theorem 24 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 them form BC where either B or C is invertible. In this section, we prove Theorem 6, which shows a quadratic lower bound for such circuits. For convenience, we restate the theorem.

▶ **Theorem 26.** *There exists an explicit family of $n \times n$ matrices $\{A_n\}$, over any field $\mathbb{F}$ such that $\mathbb{F} \geq \operatorname{poly}(n)$, such that every invertible circuit computing A_n has size $n^2/4$.*

The proof of this theorem appears in the full version of the paper.

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 [5, 15, 17] 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 $\mathbf{y}^T A\mathbf{x}$ [7, 28] (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 [46] for a discussion on the importance of this problem.

References

1 Manindra Agrawal. 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)*, pages 92–105, 2005. doi:10.1007/11590156_6.

2 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM J. Comput.*, 44(3):669–697, 2015. doi:10.1137/140975103.

3 Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 67–75, 2008. doi:10.1109/FOCS.2008.32.

4 Josh Alman and Lijie Chen. Efficient construction of rigid matrices using an np oracle. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2019)*, 2019. URL: http://joshalman.com/AlmanChenFOCS19.pdf.

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

6 Noga Alon and Pavel Pudlák. Superconcentrators of depths 2 and 3; odd levels help (rarely). *J. Comput. Syst. Sci.*, 48(1):194–202, 1994. doi:10.1016/S0022-0000(05)80027-3.

7 Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22:317–330, 1983. doi:10.1016/0304-3975(83)90110-X.

8 Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. 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)*, pages 1185–1194. ACM, 2017. doi:10.1145/3055399.3055423.

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

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

11 Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 670–683. ACM, 2016. doi:10.1145/2897518.2897528.

12 Bernard Chazelle. *The discrepancy method - randomness and complexity*. Cambridge University Press, 2001. URL: https://www.cs.princeton.edu/~chazelle/pubs/book.pdf.

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

14 Danny Dolev, Cynthia Dwork, Nicholas Pippenger, and Avi Wigderson. Superconcentrators, generalizers and generalized connectors with limited depth (preliminary version). In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC 1983)*, pages 42–51. ACM, 1983. doi:10.1145/800061.808731.

15 Zeev Dvir and Benjamin Edelman. Matrix rigidity and the croot-lev-pach lemma. *CoRR*, abs/1708.01646, 2017. arXiv:1708.01646.

16 Zeev Dvir, Alexander Golovnev, and Omri Weinstein. Static data structure lower bounds imply rigidity. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:188, 2018. URL: https://eccc.weizmann.ac.il/report/2018/188.

17 Zeev Dvir and Allen Liu. Fourier and circulant matrices are not rigid. *CoRR*, abs/1902.07334, 2019. `arXiv:1902.07334`.

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

19 Michael L. Fredman. The complexity of maintaining an array and computing its partial sums. *J. ACM*, 29(1):250–260, 1982. `doi:10.1145/322290.322305`.

20 Michael L. Fredman and Michael E. Saks. The cell probe complexity of dynamic data structures. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC 1989)*, pages 345–354. ACM, 1989. `doi:10.1145/73007.73040`.

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

22 Anna Gál, Kristoffer Arnsfelt Hansen, Michal Koucký, Pavel Pudlák, and Emanuele Viola. Tight bounds on computing error-correcting codes by bounded-depth circuits with arbitrary gates. *IEEE Trans. Information Theory*, 59(10):6611–6627, 2013. `doi:10.1109/TIT.2013.2270275`.

23 Anna Gál and Peter Bro Miltersen. The cell probe complexity of succinct data structures. *Theor. Comput. Sci.*, 379(3):405–417, 2007. `doi:10.1016/j.tcs.2007.02.047`.

24 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. `doi:10.1137/140957123`.

25 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory, 2018. URL: `https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/`.

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

27 Stasys Jukna and Igor Sergeev. Complexity of linear boolean operators. *Foundations and Trends in Theoretical Computer Science*, 9(1):1–123, 2013. `doi:10.1561/0400000063`.

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

29 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012. `doi:10.1016/j.tcs.2012.03.041`.

30 Kasper Green Larsen. The cell probe complexity of dynamic range counting. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 85–94. ACM, 2012. `doi:10.1145/2213977.2213987`.

31 Kasper Green Larsen. On range searching in the group model and combinatorial discrepancy. *SIAM J. Comput.*, 43(2):673–686, 2014. `doi:10.1137/120865240`.

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

33 Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000*, pages 556–562. MIT Press, 2000. URL: `http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization`.

34 Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. *CoRR*, abs/1804.04005, 2018. `arXiv:1804.04005`.

35 Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1-2):1–155, 2009. `doi:10.1561/0400000011`.

36 Julien Mairal, Francis R. Bach, Jean Ponce, and Guillermo Sapiro. 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*, pages 689–696. ACM, 2009. `doi:10.1145/1553374.1553463`.

37 Jacques Morgenstern. Note on a lower bound on the linear complexity of the fast fourier transform. *J. ACM*, 20(2):305–306, 1973. doi:10.1145/321752.321761.

38 Behnam Neyshabur and Rina Panigrahy. Sparse matrix factorization. *CoRR*, abs/1311.3315, 2013. arXiv:1311.3315.

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

40 Mihai Pătraşcu and Erik D. Demaine. Logarithmic lower bounds in the cell-probe model. *SIAM J. Comput.*, 35(4):932–963, 2006. doi:10.1137/S0097539705447256.

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

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

43 Pavel Pudlák. Communication in bounded depth circuits. *Combinatorica*, 14(2):203–216, 1994. doi:10.1007/BF01215351.

44 Pavel Pudlák. 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, 2000. doi:10.1016/S0020-0190(00)00058-2.

45 Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. Discrete Math.*, 13(1):2–24, 2000. doi:10.1137/S0895480197329508.

46 Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010. doi:10.4086/toc.2010.v006a007.

47 Mohammad Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. A remark on matrix rigidity. *Inf. Process. Lett.*, 64(6):283–285, 1997. doi:10.1016/S0020-0190(97)00190-7.

48 Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54:435–447, 1990. URL: https://www.ams.org/journals/mcom/1990-54-189/S0025-5718-1990-0993933-0/S0025-5718-1990-0993933-0.pdf.

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

50 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010. doi:10.1561/0400000039.

51 Volker Strassen. Die berechnungskomplexität von elementarsymmetrischen funktionen und von interpolationskoeffizienten. *Numerische Mathematik*, 20(3):238–251, June 1973. doi:10.1007/BF01436566.

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

53 Leslie G. Valiant. On non-linear lower bounds in computational complexity. In *Proceedings of the 7th Annual ACM Symposium on Theory of Computing (STOC 1975)*, pages 45–53. ACM, 1975. doi:10.1145/800116.803752.

54 Leslie G. Valiant. 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*, pages 162–176. Springer, 1977. doi:10.1007/3-540-08353-7_135.