Kronecker Products, Low-Depth Circuits, and Matrix Rigidity

Josh Alman
Harvard University
Cambridge, MA, USA

ABSTRACT

For a matrix $M$ and a positive integer $r$, the rank $r$ rigidity of $M$ is the smallest number of entries of $M$ which one must change to make its rank at most $r$. There are many known applications of rigidity lower bounds to a variety of areas in complexity theory, but fewer known applications of rigidity upper bounds. In this paper, we use rigidity upper bounds to prove new upper bounds in a few different models of computation. Our results include:

- For any $d > 1$, and over any field $\mathbb{F}$, the $N \times N$ Walsh-Hadamard transform has a depth-$d$ linear circuit of size $O(d \cdot N^{1+0.96/d})$. This circumvents a known lower bound of $Ω(d \cdot N^{1+1/d})$ for circuits with bounded coefficients over $\mathbb{C}$ by Pudlák (2000), by using coefficients of magnitude polynomial in $N$. Our construction also generalizes to linear transformations given by a Kronecker power of any fixed $2 \times 2$ matrix.

- The $N \times N$ Walsh-Hadamard transform has a linear circuit of size $\lesssim (1.81 + o(1))N \log_2 N$, improving on the bound of $\approx 1.88N \log_2 N$ which one obtains from the standard fast Walsh-Hadamard transform.

- A new rigidity upper bound, showing that the following classes of matrices are not rigid enough to prove circuit lower bounds using Valiant’s approach: (1) for any field $\mathbb{F}$ and any function $f : \{0,1\}^N \rightarrow \mathbb{F}$, the matrix $V_f \in \mathbb{F}^{2^N \times 2^N}$ given by, for any $x, y \in \{0,1\}^N$, $V_f[x,y] = f(x \land y)$, and (2) for any field $\mathbb{F}$ and any fixed-size matrices $M_1, \ldots, M_n \in \mathbb{F}^{N \times N}$, the Kronecker product $M_1 \otimes M_2 \otimes \cdots \otimes M_n$. This generalizes recent results on non-rigidity, using a simpler approach which avoids needing the polynomial method.

- New connections between recursive linear transformations like Fourier and Walsh-Hadamard transforms, and circuits for matrix multiplication.

CCS CONCEPTS

- Theory of computation → Design and analysis of algorithms; Circuit complexity; Computing methodologies → Linear algebra algorithms.

KEYWORDS

Kronecker Product, Linear Circuit, Matrix Rigidity, Walsh-Hadamard Transform

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

STOC ’21, June 21–25, 2021, Virtual, Italy
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8053-9/21/06...$15.00
https://doi.org/10.1145/3406325.3451008

1 INTRODUCTION

For a matrix $M$ and a positive integer $r$, the rank $r$ rigidity of $M$, denoted $R_M(r)$, is the smallest number of entries of $M$ which one must change to make its rank at most $r$. Matrix rigidity was introduced by L. Valiant [31] as a tool for proving low-depth circuit lower bounds. He showed that for any family $\{M_N\}_{N \in \mathbb{N}}$ of matrices with $M_N \in \mathbb{F}^{N \times N}$, if $R_M(N) = O(N \log \log N)$ for any fixed $r > 0$, then the linear transformation which takes as input a vector $x \in \mathbb{F}^N$ and outputs $M_N x$ cannot be computed by an arithmetic circuit of size $O(N)$ and depth $O(\log N)$. We say $M_N$ is Valiant-rigid if it satisfies this rigidity lower bound. It remains a major open problem to prove that any explicit family of matrices cannot be computed by circuits of size $O(N)$ and depth $O(\log N)$, and one of the most-studied approaches to this problem is to try to construct an explicit family of Valiant-rigid matrices.

Many researchers have subsequently shown that rigidity lower bounds for explicit matrices, both in this parameter regime and others, would lead to new lower bounds in a variety of areas, including in arithmetic complexity, communication complexity, Boolean circuit complexity, and cryptography. We refer the reader to [20] for more on the background and known applications of matrix rigidity. However, despite 40+ years of efforts, and plenty of known applications, there are no known fully explicit constructions of rigid matrices.

A recent line of work [5, 11, 13] has instead shown that a number of families of explicit matrices are in fact not Valiant rigid, including the Walsh-Hadamard transform [5] and the discrete Fourier transform [13]. These had been some of the most-studied candidate rigid matrices, which are now ruled out for proving lower bounds using this approach. This raises the question: Do these rigidity upper bounds imply any other interesting upper bounds? Although there are many results showing that rigid matrices imply a variety of lower bounds, there are few known connections showing that rigidity upper bounds would yield new algorithms or circuits.

In this paper, we give new upper bounds in a few different models which make use of recent rigidity upper bounds. Some of them apply rigidity upper bounds directly, while others are inspired by the proof techniques of recent rigidity upper bounds.

1.1 Low-Depth Linear Circuits

We begin by studying linear circuits for computing a linear transformation $M \in \mathbb{F}^{N \times N}$. These are circuits in which the inputs are...
the $N$ entries of a vector $x \in \mathbb{F}^N$, the outputs must be the $N$ entries of $Mx$, and each gate computes an $\mathbb{F}$-linear combination of its inputs. We focus on low-depth circuits with unbounded fan-in gates, so we measure their size by the number of wires in the circuit. A special type of linear circuit which we focus on is a synchronous linear circuit, in which the inputs to each gate must all have the same depth. One can see that a synchronous linear circuit of size $s$ and depth $d$ for $M$ corresponds to $d$ matrices $M_1, \ldots, M_d$ such that $M = M_1 \times \cdots \times M_d$ and $\text{nnz}(M_1) + \cdots + \text{nnz}(M_d) = s$, where $\text{nnz}(A)$ denotes the number of nonzero entries in matrix $A$. A depth $d$ linear circuit can be converted into a depth $d$ synchronous linear circuit with a multiplicative size blowup of only $d$.

Rigidity upper bounds naturally give depth-2 linear circuit constructions. Indeed, it is not hard to see that any $M \in \mathbb{F}^{N \times N}$ has a depth-2 linear circuit of size $O(N \cdot \text{rank}(M))$, and a depth-1 linear circuit of size $O(\text{nnz}(M))$, and hence, for any $r$, a depth-2 linear circuit of size $O(N \cdot r + R_{\mathbb{F}}(r))$. Thus, for instance, letting $H_n$ denote the $N \times N$ Walsh-Hadamard transform for $N = 2^n$, using the rigidity upper bound $R_{\mathbb{F}}(H_n(N^{-10\log^2(1/\epsilon)})) \leq N^{1+\epsilon}$ for any $\epsilon > 0$ of [5], it follows that there is a fixed $\delta > 0$ such that $H_n$ has a depth-2 linear circuit of size $O(2^d\delta)$.

However, there is actually a smaller and simpler circuit known for $H_n$. Using an approach similar to the fast Walsh-Hadamard Transform, we can see that for any $d$, $H_n$ has a depth-$d$ synchronous linear circuit of size only $O(d \cdot N^{1+1/d})$. The circuit involves, at each depth, computing $N^{1-1/d}$ independent copies of the $N^{1/d} \times N^{1/d}$ Walsh-Hadamard transform $H_n(d)$. Thus, $H_n$ has a depth-2 circuit of size only $O(N^{1+\delta})$, which is much better than $O(2^d \delta)$. Despite a fair bit of work by the author, it is unclear how to use the rigidity upper bound of [5] to improve on $O(N^{1+\delta})$.

Nonetheless, we are able to construct smaller circuits for $H_n$, as well as any other family of transforms defined as the Kronecker power of a fixed matrix, by making use of new, different rigidity upper bounds for $H_n$. For a fixed $2 \times 2$ matrix

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

over a field $\mathbb{F}$, the family of Kronecker powers of $M$, denoted by $M^{\otimes n} \in \mathbb{F}^{2^n \times 2^n}$, is defined recursively by $M^{\otimes 1} = M$, and for $n \geq 1$,

$$M^{\otimes (n+1)} = \begin{bmatrix} a \cdot M^{\otimes n} & b \cdot M^{\otimes n} \\ c \cdot M^{\otimes n} & d \cdot M^{\otimes n} \end{bmatrix}.$$ 

For instance, the $2^n \times 2^n$ Walsh-Hadamard transform $H_n$ is defined as $H_n := H_1^{\otimes n}$, where

$$H_1 := \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$ 

Kronecker powers arise naturally in many settings. For instance, when

$$M = \begin{bmatrix} 1 & 1 \\ 1 & \omega \end{bmatrix}$$

for some element $\omega \in \mathbb{F}$, then the linear transformation $M^{\otimes n}$ corresponds to evaluating an $n$-variate multilinear polynomial over $\mathbb{F}$ on all inputs in $\{1, \omega\}^n$.

Our main result is as follows:

**Theorem 1.1.** Let $\mathbb{F}$ be any field, and let $M \in \mathbb{F}^{2 \times 2}$ be any matrix over $\mathbb{F}$. There is a constant $\epsilon > 0.01526$ such that, for any positive integers $n, d$, the linear transformation $M^{\otimes n} \in \mathbb{F}^{N \times N}$ for $N = 2^n$ has a depth-$d$ synchronous linear circuit of size $2^d \cdot d \cdot N^{1+(1-\epsilon)/d}$.

When $M = H_1$, so that $M^{\otimes n}$ is the Walsh-Hadamard transform $H_n$, we can improve the bound to $\epsilon > 0.04816$.

Our new result shows that $H_n$ has a depth-2 linear circuit of size only $O(N^{1.47592})$, and more generally improves the size of a depth-$d$ linear circuit for $H_n$ or any $n$th Kronecker power when $d < o(\log n)$. When $d$ divides $n$, we can improve the upper bound to $d \cdot N^{1+(1-\epsilon)/d}$, removing the $2^d$ factor. This construction may be of practical interest, as it improves on the previous bound of $d \cdot N^{1+1/d}$, even for small constant values of $N$ and $d$.

Theorem 1.1 is also particularly interesting when compared to a lower bound of Pudlák [26] against low-depth linear circuits with bounded coefficients for computing $H_n$ over $\mathbb{C}$. Recall that in a linear circuit over $\mathbb{C}$, each gate computes a $\mathbb{C}$-linear combination of its inputs. For a positive real number $c$, we say the circuit has $c$-bounded coefficients if, for each gate, the coefficients of the linear combination are complex numbers of magnitude at most $c$. Motivated by the fact that the best known linear circuits for many important linear transformations, including the Walsh-Hadamard transform and the discrete Fourier transform, use only 1-bounded coefficients (prior to this paper), a line of work [7, 8, 19, 22, 24, 26, 28] (see also [20, Section 3.3]) has shown strong, often tight lower bounds for linear circuits with bounded coefficients. Pudlák [26] showed that the aforementioned circuit of depth $d$ and size $O(d \cdot N^{1+1/d})$ is optimal for bounded coefficient circuits:

**Theorem 1.2 ([26]).** Any depth $d$ synchronous linear circuit with $c$-bounded coefficients for computing the Walsh-Hadamard transform $H_n \in \mathbb{C}^{N \times N}$ for $N = 2^n$ has size $\geq d \cdot N^{1+1/d}/c^2$.

Our Theorem 1.1 circumvents this lower bound by using large coefficients. Indeed, we will see that over $\mathbb{F} = \mathbb{C}$, we use coefficients which are integers of magnitude up to $N^{O(1)}$. That said, it should be noted that, since our coefficients are only $O(\log N)$-bit integers, the additional time required to do the arithmetic for the coefficients of our circuit is still negligible compared to the circuit size savings in any reasonable model of computation.

To our knowledge, this is the first non-trivial upper bound surpassing one of the aforementioned bounded-coefficient lower bounds. This shows that using larger coefficients can make a substantial difference in the circuit size required, even when computing the linear transformation of a matrix whose entries are all in $\{-1, 1\}$. At the same time, it is interesting to note that our Theorem 1.1 works over any field, even a constant-sized finite field like $\mathbb{F}_3$ where there are no ‘large’ coefficients. One could have imagined that overcoming bounded-coefficient lower bounds, when possible, requires using an infinite field and large coefficients, but at least in this setting, that is not the case.

Our proof of Theorem 1.1 begins with a new general framework for designing smaller low-depth circuits for recursively-defined families of matrices like $H_n$. We show that a nontrivial synchronous circuit construction for any fixed matrix in the family leads to a smaller circuit for every matrix in the family.

**Lemma 1.3.** Let $M \in \mathbb{F}^{q \times q}$ be a $q \times q$ matrix over any field $\mathbb{F}$, and suppose there are matrices $A_1, \ldots, A_d$ such that $M = \prod_{j=1}^d A_j$ and $\text{nnz}(A_i) \leq q^c$ for all $i \in [d]$. Then, for every positive integer $n$, letting
$N = q^n$, the $N \times N$ matrix $M^{\otimes n}$ has a depth-$d$ synchronous linear circuit of size $O(N^n)$.

Lemma 1.3 follows by simply calculating how a Kronecker power changes the given circuit for $M$, but it is nonetheless conceptually interesting: in order to design a small circuit for the entire family of matrices $M^{\otimes n}$, it suffices to design one for any fixed matrix in the family. Lemma 1.3 is similar to the approach for designing matrix multiplication algorithms spearheaded by Strassen [30], where an identity for quickly multiplying fixed size matrices implies asymptotic improvements for multiplying matrices of any sizes. Our proof was inspired by this, as Kronecker products also play a central role in the definition and study of matrix multiplication tensors.

We then use rigidity upper-bounds for the $q \times q$ matrix $M$ to construct fixed upper bounds. One can see by concatenating the two parts of a non-rigidity expression for $M$ that, for any rank $r$, we can find matrices $B, C$ with $M = B \times C$, $nnz(B) = q(r + 1)$, and $nnz(C) = q \cdot r + R_M(r)$. We can `symmetrize’ this construction using a Kronecker product trick, then apply Lemma 1.3 to yield:

**Lemma 1.4.** Let $M \in \mathbb{F}^{q \times q}$ be a $q \times q$ matrix over any field $\mathbb{F}$, and $1 \leq r \leq q$ be any rank, and define $c := \log_q(q(r + 1) \cdot (r + R_M(r)/q))$.

Then, for any positive integer $n$, setting $N = q^n$, the $N \times N$ matrix $M^{\otimes n}$ has a depth-$d$ synchronous circuit of size $O(d \cdot N^{1+ce/d})$.

Thus, rigidity upper bounds on $H_n$ for a fixed $m$ can give nontrivial low-depth circuit upper bounds for $H_n$ for all $n$. Unfortunately, we cannot simply substitute in the rigidity upper bound of $\Omega$ to prove our result. Indeed, to achieve $c < 1$ in Lemma 1.4 when applying it to the $q \times q$ matrix $H_m$ for $q = 2^m$, it is not hard to see that we need $r < \sqrt{q}$. By comparison, the bound from [5] is primarily interesting for higher rank $r > q^{1-\epsilon'}$ for small $\epsilon' > 0$. Other known constructions, including those from probabilistic polynomials [4], do not seem to give a nontrivial bound here either. Instead, to prove our upper bound, we use a new rigidity upper bound for $H_n$ for rank $r = 1$, and more specifically, Theorem 1.1 ultimately follows from a new construction we give for the $16 \times 16$ matrix $H_4$ showing that $R_{H_4}(1) \leq 96$.

Using rigidity upper bounds naturally leads to `symmetric’ circuits to use in Lemma 1.3, but one could imagine other approaches that lead to more `lopsided’ constructions. We additionally prove a generalization of Lemma 1.3, that even such constructions can lead to upper bounds for $M^{\otimes n}$ for all $n$:

**Lemma 1.5.** Let $M \in \mathbb{F}^{q \times q}$ be a $q \times q$ matrix over any field $\mathbb{F}$, and suppose there are matrices $A_1, \ldots, A_d$ such that $M = \prod_{j=1}^{d} A_j$, which is nontrivial in the sense that $\prod_{j=1}^{d} \text{nnz}(A_j) \leq q^{d+\epsilon'}$ for some $\epsilon' < 1$. Then, for every positive integer $n$, setting $N = q^n$, the $N \times N$ matrix $M^{\otimes n}$ has a depth-$d$ synchronous circuit of size $O(N^{1+\epsilon'/d})$ for a constant $\epsilon' < 1$ which depends only on $\epsilon'$.

Note that one could achieve $\epsilon' = 1$ in Lemma 1.5 trivially by picking $A_1 = M$ and $A_2 = \cdots = A_d = I_q$, the $q \times q$ identity matrix. Lemma 1.5 shows that any construction which improves on this at all leads to an asymptotically smaller circuit for $M^{\otimes n}$. While Lemma 1.3 required that each $A_j$ has $\text{nnz}(A_j) < q^{1+1/d}$, Lemma 1.5 instead only requires that the geometric mean of all the $\text{nnz}(A_j)$ is less than $q^{1+1/d}$. However, it results in a slightly worse final size bound, which is why we use Lemma 1.3 to prove Theorem 1.1.

### 1.2 Surpassing Other Bounded-Coefficient Lower Bounds?

It is natural to ask next whether our techniques can be used to overcome other bounded-coefficient lower bounds. We discuss a few more:

**Unbounded-Depth Circuits for $H_n$.** Pudlák [26] also showed a lower bound against unbounded-depth bounded-coefficient synchronous linear circuits for computing $H_n$.

**Theorem 1.6 ([26]).** Any synchronous linear circuit with $c$-bounded coefficients for computing the Walsh-Hadamard transform $H_n \in \mathbb{C}^{N \times N}$ for $N = 2^n$ has size $\Omega\left(\frac{\log \log N}{c^2}\right) N \log_2 N$.

For $c = 1$ (as is the case in all previous circuits for $H_n$), this gives a lower bound of $\Omega \left(\log_2(2) \cdot N \log_2 N \cdot \log N\right)$. This is known to be tight, as optimizing for $d$ in the usual fast Walsh-Hadamard transform gives a matching upper bound. In fact, we give a new construction which also beats this lower bound, although only by a constant factor.

**Theorem 1.7.** Let $\mathbb{F}$ be any field, and let $M \in \mathbb{F}^{2 \times 2}$ be any matrix over $\mathbb{F}$. There is a constant $\epsilon > 0.01526$ such that, for any positive integer $n$, the linear transformation $M^{\otimes n} \in \mathbb{F}^{2 \times 2n}$ for $N = 2^n$ has a synchronous linear circuit of size $\left(1 - \epsilon + o(1)\right) \cdot e \cdot \log_2(2) \cdot N \log_2 N$. When $M = H_1$, so that $M^{\otimes n}$ is the Walsh-Hadamard transform $H_n$, we can improve the bound to $\epsilon > 0.04816$.

It is no coincidence that our bounds on $\epsilon$ in Theorem 1.7 are the same as those in Theorem 1.1: We prove Theorem 1.7 by introducing a gadget which increases the depth in Theorem 1.1 but removes the additional unwanted $2^r$ term in the circuit size (which would otherwise impact our constant-factor savings), and then optimizing over all choices of $d$.

Of course, it would be much more exciting to design a circuit of size $o(N \log N)$ for $H_n$, but that is currently beyond our techniques. That said, we believe Theorem 1.7 gives the first improvement of any kind on the standard fast Hadamard transform for computing $H_n$, and we are optimistic that further improvements are possible.

**Circuits for the Fourier Transform.** Pudlák showed that both Theorem 1.2 and Theorem 1.6 also hold for the Discrete Fourier transform [2], $F_N \in \mathbb{C}^{N \times N}$. Can our approach be used to beat these lower bounds as well? We remark that $F_N$ is actually too rigid for our approach using Lemma 1.4 to apply to overcome this bound. Interestingly, the rigidity lower bound we use to show this is not the asymptotically best known bound of $R_{F_N}(r) \geq \Omega\left(\frac{N}{r+1/2}\log(N/r)\right)$, but instead the bound $R_{F_N}(r) \geq (N - r)^2/(r + 1)$ [29] which has better known constant factors for small $r$.

It should be noted that we do not rule out the existence of $o(d \cdot N^{1+1/d})$ size-depth-$d$ linear circuits for $F_N$, or even rule out that Lemma 1.3 could be used to construct such circuits. However, an approach different from our non-rigidity approach would be needed to give the nontrivial construction needed by Lemma 1.3.

---

[2] Morgenstern [22] first showed such a result for linear circuits which need not be synchronous, with slightly lower leading constant factors.
Matrix Multiplication. Raz [28] showed that any bilinear circuit with bounded coefficients for computing the product of two $N \times N$ matrices over $\mathbb{C}$ requires size $\Omega(N^2 \log N)$. This is not known to be tight: the best known circuit for $N \times N \times N$ matrix multiplication has size $N^{\omega+o(1)}$ where $\omega \leq 2.373$ [6, 17, 35] is the matrix multiplication exponent. That said, as we will discuss soon in more detail in Section 1.4, there is a strong connection between this lower bound and the aforementioned bounded-coefficient lower bounds: if one could surpass Raz’s lower bound and design an $o(N^2 \log N)$ size circuit for matrix multiplication, it would lead to linear circuits of size $o(N \log N)$ for both the $N \times N$ discrete Fourier transform and the $N \times N$ Walsh-Hadamard transform, as well as many related linear transformations.

1.3 More Matrices Are Not Valiant-Rigid

Our next upper bound is a new non-rigidity result, which generalizes and sheds new light on the non-rigidity of the Walsh-Hadamard transform [5]. We focus on two families of matrices $M$ which generalize $H_N$.

(1) Matrices $M \in \mathbb{F}^n \times \mathbb{F}^n$ of the form $M = \bigotimes_{i=1}^n M_i$ for positive integers $q, n$ and any matrices $M_1, \ldots, M_n \in \mathbb{F}^{q \times q}$ (where $\otimes$ denotes the Kronecker product). Kronecker power matrices like $H_N$ which we discussed earlier are of this form with $M_1 = M_2 = \ldots = M_n$, but here we also allow for different choices of the matrices $M_1, \ldots, M_n$.

(2) Matrices $M \in \mathbb{F}^n \times \mathbb{F}^n$ whose entries are given by, for $x, y \in \{0, 1, \ldots, q-1\}^n$:

$$M[x, y] = f(\max\{x[1], y[1]\}, \max\{x[2], y[2]\}, \ldots, \max\{x[n], y[n]\})$$

for any function $f : \{0, 1, \ldots, q-1\}^n \to \mathbb{F}$. For instance, $H_n$ is of this form with $q = 2$ when $f$ is the parity function, but we also allow for more complicated choices of $f$.

**Theorem 1.8.** Any matrix of either of the above forms with $q \leq O(\log n)$ is not Valiant-rigid. More precisely, setting $N = q^n$, any such $M$ satisfies, for any sufficiently small $\epsilon > 0$:

$$R_M(N^{1 - \frac{1}{2^n} \log^2(1/\epsilon)}) \leq N^{1+\epsilon}.$$

The constant hidden by the $O$ in Theorem 1.8 is not too small; for instance, we show that when $q = 2$, any such $M$ has $R_M(O(2^{0.81}N)) < o(N^2)$.

Theorem 1.8 shows that it was not just a ‘coincidence’ that $H_n$ is not rigid, but in fact a number of big families of matrices generalizing $H_n$ are also not rigid. It, of course, rules out the Valiant-rigidity approach for proving circuit lower bounds for any of these linear transformations. We now discuss the two families of matrices in some more detail.

(1) Aside from being a natural generalization of $H_n$, Kronecker products like this are ubiquitous in many areas of computational science (see e.g. [32]). The non-rigidity of these matrices is also interesting compared with our observation which we discuss in detail in the upcoming Section 1.4 that: if there are Valiant-rigid matrices in this family for any fixed $n$ and growing $q$, then we would get a lower bound for $N \times N \times N^n$ matrix multiplication. By comparison, Theorem 1.8 shows there are no Valiant-rigid matrices in this family for fixed $n$ and growing $n$. The difference between this family of matrices when $n$ is growing versus when $q$ is growing is not unlike the difference between the families of Walsh-Hadamard transforms and Fourier transforms (which are both Hadamard matrices for different choices of which of the two defining parameters is growing). Perhaps the techniques of [13] for showing that Fourier transforms are not rigid could help to approach this other setting.

(2) As noticed by [5], matrices of this form for different choices of the function $f : \{0, 1, \ldots, q-1\}^n \to \mathbb{F}$ arise frequently in fine-grained complexity, especially in the case $q = 2$. In fact, the best known algorithms for a number of different problems have used, as their key insight, that this type of matrix $M$ is not rigid, including the Orthogonal Vectors problem [1] (for $f = \text{AND}$), All-Pairs Shortest Paths [34] (also for $f = \text{AND}$), and Hamming Nearest Neighbors [2, 4] (for $f = \text{MAJORITY}$). These algorithms all use the ‘polynomial method’ to show that $M$ is not rigid in a low-rank, high-error regime, but it is unclear how to extend them to less structured functions $f$. By comparison, Theorem 1.8 shows that $M$ is not rigid in a higher-rank, lower-error regime, and it applies to any function $f$.

In fact, in addition to these aforementioned algorithms, all the prior work on showing that matrices of interest are not Valiant-rigid [5, 11, 13] has used the polynomial method. For instance, the previous proof of the non-rigidity of the Walsh-Hadamard transform [5] critically used the fact that the corresponding function $f = \text{PARITY}$ has low-degree polynomial approximations (which are correct on most inputs) over any field. Our rigidity upper bound does not use the polynomial method (at least explicitly), and applies to any function $f$ without any restriction on how well it can be approximated by polynomials. In other words, this central property of $f$ that was used by prior work is actually unnecessary for proving that $M$ is not Valiant-rigid.

Our proof of Theorem 1.8 in the case $q = 2$ is actually quite simple, and it simplifies the previous proof of the non-rigidity of the Walsh-Hadamard transform. Inspired by Dvir and Liu [13], who frequently make use of the fact that the product of a constant number of matrices which are not Valiant-rigid is, itself, not Valiant-rigid (see Lemma 2.11 below), we begin by noticing that any matrix $M$ from either of the two families can be written as

$$M = D \times R_n \times D' \times R_n \times D'' ,$$

where $D, D', D'' \in \mathbb{F}^{2^n \times 2^n}$ are three carefully-chosen diagonal matrices (which are evidently not Valiant-rigid), and $R_n \in \{0, 1\}^{2^n \times 2^n}$ is the disjointness matrix, given by $R_n := R_n^\otimes$ where

$$R_n := \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} .$$

Thus, to show that any such $M$ is not Valiant-rigid, it suffices to show that $R_n$ is not Valiant-rigid. However, this is not too difficult, since $R_n$ is a fairly sparse matrix to begin with! Indeed, $R_n$ is a $2^n \times 2^n$ matrix, but has only $3^n$ nonzero entries. Moreover, most of these nonzero entries are concentrated in a few rows and columns: for each integer $0 \leq k \leq n$, the matrix $R_n$ has $\binom{n}{k}$ rows (or columns)
with $2^k$ nonzero entries. Using standard bounds on binomial coefficients, we thus see that, by removing only the $2n(1-\Theta(n^2 \log (1/e)))$ densest rows and columns of $R_N$, we are left with a matrix with only $2^{n+k}$ nonzero entries per row or column. Since changing a single row or column of a matrix is a rank-1 update, this shows that $R_N$ is not Valiant-rigid as desired.

Extending this result to larger $q$ is quite a bit more involved. Let us focus now on family 1 of matrices above (Kronecker products of $n$ different $q \times q$ matrices); the proof for family 2 is similar. We will proceed by induction on $q$. Our starting point is the remark that any $q \times q$ matrix $M_i$ can be written as the sum of a $q \times q$ rank-1 matrix $J_i$, and a $(q-1) \times (q-1)$ matrix $L_i$ (padded with a row and column of 0s). For instance, in the case $q = 3$ we have (assuming the top-left entry $a$ is nonzero):

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & e - bd & f - be \\ 0 & h - be & i - bc \end{bmatrix}.$$

We have now written $M_i = J_i + L_i$, and we know that $\bigotimes_{i=1}^n J_i$ is not Valiant-rigid (in fact, it has rank 1), and $\bigotimes_{i=1}^n L_i$ is not Valiant-rigid, even when thought of as a $(q-1)^n \times (q-1)^n$ matrix, by the inductive hypothesis. This does not imply that $\bigotimes_{i=1}^n M_i$ is not Valiant-rigid on its own, however, because there are cross-terms:

$$\bigotimes_{i=1}^n M_i = \bigotimes_{i=1}^n (J_i + L_i) = \bigotimes_{K \subseteq \{1, 2, \ldots, n\}} \bigotimes_{i=1}^n (J_i \oplus L_i).$$

(Here, we are using $\{l \in K \? L_i : J_i\}$ as the ternary operator, which equals $L_i$ when $i \in K$, and equals $J_i$ when $i \notin K$.) For any particular $K$, the matrix $M_K := \bigotimes_{i=1}^n (\{l \in K \? L_i : J_i\})$ can be seen as the Kronecker product of a $q^{|K|} \times q^{|K|}$ matrix of rank 1, and a $q^n - |K| \times q^n - |K| \times q^n - |K| \times q^n - |K|$ matrix which, by the inductive hypothesis, is not Valiant-rigid. It can be shown (see e.g. [13, Section 6]) that the Kronecker product of matrices which are not Valiant-rigid is itself not Valiant-rigid, and hence that $M_K$ is not Valiant-rigid. However, this is still not sufficient: we have now only expressed $M$ as the sum of $2^n$ matrices which are not Valiant-rigid, but whose sum might still be.

We instead first perform a number of low-rank updates to $M$ to simplify the problem. We first subtract away all the matrices $M_K$ for which $|K|$ is not close to $(q-1)n/q$. Next, we remove all rows and columns corresponding to $x \in \{0, 1, \ldots, q-1\}^n$ for which $\text{nnz}(x)$ is not close to $(q-1)n/q$. Finally, we observe that each remaining row of $M$ only intersects with a nonzero row of $\Theta(x)$ different choices of remaining matrices $M_K$ (compared with $q^n$ before). Hence, the fact that each $M_K$ is not Valiant-rigid implies our desired non-rigidity, as the sparsity per row is now only multiplied by $\Theta(x)$. We have, of course, glossed over many important and intricate aspects of the proof; we refer the reader to the full version of the paper for the details.

We briefly remark that the techniques for manipulating Kronecker products used by Dvir and Liu [13] do not appear sufficient to prove our Theorem 1.8. They observed that the Kronecker product of matrices $M_1, \ldots, M_n$ which are not Valiant-rigid is itself not Valiant-rigid. In particular, they begin with a decomposition $M_i = J_i + L_i$ where $J_i$ has low rank like in our setting, but they further assume that $L_i$ is very sparse. In our case, $M_1, \ldots, M_n$ are arbitrary matrices, and may all be very rigid on their own, and so a more intricate argument seems necessary.

### 1.4 Connections between Matrix Multiplication and Kronecker Product Linear Transformations

As we previously mentioned, Raz [28] showed that any bilinear circuit with bounded coefficients for computing the product of two $N \times N$ matrices over $C$ requires size $\Omega(N^2 \log N)$. A key insight behind Raz’s lower bound is that, for a fixed matrix $A \in \mathbb{F}^{N \times N}$, the following two problems are equivalent:

- Given as input a matrix $B \in \mathbb{F}^{N \times N}$, output the matrix $A \times B$.
- Given as input a vector $b \in \mathbb{F}^N$, output the linear transformation $(I_N \otimes A)b$.

In particular, if one could show that there is any matrix $A \in \mathbb{F}^{N \times N}$ for which the linear transformation $I_N \otimes A \in \mathbb{F}^{N \times N^2}$ does not have $O(N^2)$ size circuits, then $N \times N$ matrix multiplication does not have $O(N^2)$ size circuits. One intriguing avenue toward showing this is to show that there exists an $A \in \mathbb{F}^{N \times N}$ such that $I_N \otimes A$ is Valiant-rigid. In contrast with the usual setting in matrix rigidity, here, to show a lower bound against a particular problem (matrix multiplication), it suffices to show that there exists a rigid matrix among a large family of matrices. (Roughly, Raz’s lower bound is proved by showing there exists an $A \in \mathbb{F}^{N \times N}$ such that $I_N \otimes A$ has a high value of a variant of rigidity which corresponds to bounded-coefficient circuits.)

We take this observation further, showing that there is a much larger family of matrices for which a circuit lower bound would imply lower bounds for matrix multiplication. The key idea is the following algorithm for using matrix multiplication to compute linear transformations defined by Kronecker products (which is not very difficult to prove, and is likely folklore):

**Proposition 1.9. For any field $\mathbb{F}$, and any fixed positive integer $k$, suppose that $N \times N \times N^{k-1}$ matrix multiplication over $\mathbb{F}$ has an arithmetic circuit of size $o(N^k \log N)$. Then, the $N \times N$ Fourier transform, $N \times N$ Walsh-Hadamard transform, and any transform which can be written as the Kronecker product of $k$ different $N^{1/k} \times N^{1/k}$ size matrices, have arithmetic circuits of size $o(N \log N)$.

Applying Proposition 1.9 with $k = 2$, we see that if one shows there are any matrices $A, B \in \mathbb{F}^{N \times N}$ such that $A \otimes B \in \mathbb{F}^{N^2 \times N^2}$ requires circuits of size $\Omega(N^2 \log N)$ (perhaps making use of a proof that $A \otimes B$ is Valiant-rigid\(^3\), or in some other way), then $N \times N$ matrix multiplication requires circuits of size $\Omega(N^2 \log N)$. By comparison, even for very simple matrices of the form $A \otimes B$ such as the $N^2 \times N^2$ Discrete Fourier transform or Walsh-Hadamard transform, the best known circuit size is only $O(N^2 \log N)$.

Proposition 1.9 becomes more exciting from an algorithmic perspective as we consider larger $k$. For $k = 2$, the upper bound of

\(^3\)Actually, showing that $A \otimes B$ is Valiant-rigid would only prove a $\tilde{O}(N^2)$ lower bound against $O((\log N)^k)$-depth circuits for $N \times N \times N^k$ matrix multiplication. Normally, a $O((\log N)^k)$-depth restriction on circuits for $N \times N \times N$ matrix multiplication is not very limiting, since it is known that arithmetic circuits for matrix multiplication can be converted into logarithmic-depth circuits with only a $O(N^k)$ blowup in size for any $\epsilon > 0$ (which, in particular, does not affect the value of the matrix multiplication exponent $\epsilon$). However, in our setting where the resulting lower bounds are only for size $O(N^2 \log N)$, this $N^\epsilon$ term may be non-negligible.
$\Omega(N^2 \log N)$ needed for $N \times N \times N$ matrix multiplication is quite far away from our current best upper bound of roughly $O(N^{2.373})$. However, as $k$ grows, the exponent is known to approach $k$ as well.

**Proposition 1.10** ([15]). For every field $\mathbb{F}$ and integer $k > 1$, there is a circuit of size $O(N^{k \cdot \log_2 k - (k)})$ for performing $N \times N \times N^{k-1}$ matrix multiplication. Here, the $O$ is hiding a function of $k$. Note that the exponent is

$$k \cdot \log_{k-1}(k) = k + O\left(\frac{1}{\log k}\right).$$

In fact, working through the details (see the full version of the paper for details), we find that for a slightly super-constant choice of $k = \log N / \log \log N$, a circuit of size $O(N^{k \cdot \log^2_{k-1}(k)})$ for $N \times N \times N^{k-1}$ matrix multiplication would lead to an $o(N \log N)$ time algorithm for the $N \times N$ Fourier transform and the $N \times N$ Walsh-Hadamard transform. Unfortunately, this is not exactly what is a faster folklore algorithm running in time only $O(N)$ for the Orthogonal Vectors problem (which corresponds to the proof of Theorem 1.8 can be used to extend certain algorithms to a more general class of problems. Recall

$$f : \{0, 1\}^d \rightarrow \mathbb{F}$$

is a function which can be evaluated in time $T$. Then, given a set $S \subseteq \{0, 1\}^d$ of size $|S| = m$, there is an algorithm running in time $O(m + (d + T) \cdot 2^d)$ for computing, for all $s \in S$, the sum $\sum_{t \in S} f(t[1] \wedge t[2] \wedge \ldots \wedge t[d])$. When $f = \text{NOR}$, this algorithm counts the number of Orthogonal Vectors. However, other functions $f$ correspond to other interesting tasks. For instance, when $f$ is a threshold function (such as $\text{MAJORITY}$), this algorithm counts the number of pairs of points which share a certain number of $1$s in common, which is a basic nearest neighbor search problem, in time $O(m \cdot d \cdot 2^d)$. This improves on the more straightforward $O(m \cdot d \cdot 2^d)$ time algorithm for this problem when $d = o(m)$.

### 1.6 Other Related Work

**Rigidity Upper Bounds from Low-Depth Circuit Upper Bounds.** Our results discussed in Section 1.1 above show how rigidity upper bounds for a matrix $M$ can be used to construct small low-depth circuits for $M$. Relatedly, Pudlák [25] showed a type of converse: that low-depth circuit upper bounds can be used to show rigidity upper bounds.

**Proposition 1.11** ([25, Proposition 2]). For any field $\mathbb{F}$, positive integers $r, d, \epsilon, \delta \geq 0$ and $M \in \mathbb{F}^{N \times N}$, if $M$ has a depth-$d$ linear circuit of size $O(d \cdot N^{1+c/d})$, then $R_M(\epsilon \cdot N) \leq (d/\epsilon)^d \cdot N^{1+c}$.

Although this can be combined with our Theorem 1.1 to prove rigidity upper bounds for $H_n$ and other Kronecker power matrices, the resulting bounds are weaker than what we prove in Theorem 1.8 using a different approach, and do not suffice to prove that these matrices are not Valiant-rigid. Perhaps there is a different way to reconcile the two?

Data Structures and Rigidity. Rigidity upper bounds are known to give rise to data structure bounds: Dvir, Golovnev, and Weinstein [12] recently showed this for static data structures, and Ramamoorthy and Rashtchian [23] showed this for systematic linear data structures.

Small Depth Circuit Lower Bounds. The best-known lower bounds on the size of a depth-$2$ linear circuit for computing an explicit $N \times N$ linear transformation are only $\Omega(N \log^2 N/\log \log N)$ for efficient error-correcting codes over constant-size finite fields [14], or $\Omega(N \log^2 N / \log \log N)$ for matrices arising from super-concentrator graphs over larger fields [27]. Two recent lower bounds were also shown for less-explicit matrices: Kumar and Volk [33] constructed a matrix in time $\exp(N^{O(1)})$, over a field of size $\exp(N^{O(1)})$, which requires depth-$d$ circuits of size $N^{1+1/2d}$. With Chen [3], we construct a matrix in $P^{NP}$ which has $\{0, 1\}$ entries over any fixed-size finite field and which requires depth-$2$ circuits of size $\Omega(N \cdot 2^{|\log N|^{1/3-\delta}})$ for any $\delta > 0$. In other words, the known techniques are far from proving that any of the depth-$d$ upper bounds presented here, which are of the form $O(N^{1+1/(1-\epsilon)d})$ for somewhat small constants $\epsilon > 0$, are tight.

Other Circuit Models for Matrices. Circuit models other than linear circuits have also been studied for computing matrices in certain settings. For instance, when working with matrices over a semigroup (like the OR semigroup) or a semiring (like the SUM semiring) instead of a field, one can consider circuits where the gates compute sums from that semigroup or semiring instead. See, for instance, the book by Jukna and Sergeev which these models in detail [16]. These models have applications to areas like communication complexity, and the techniques for constructing circuits in these models often apply to the linear circuit model as well. For instance, we remark in Section 4.4 below that a construction by Jukna and Sergeev for the disjointness matrix $R_n$, which
takes advantage of both the recursive definition and the sparsity of $R_n$, leads to a better upper bound for low-depth circuits for $R_n$ than we are able to prove using our rigidity approach.

1.7 Outline

In Section 2, we introduce the notions and notation we will use, and we present a number of basic tools for working with Kronecker products and linear circuits. We then prove Theorem 1.1 in Sections 3 and 4: we prove Lemma 1.3 and Lemma 1.4 in Section 3, and then we study low-rank rigidity upper bounds for a number of families of matrices in Section 4. The remaining proofs can be found in the full version of the paper.

2 PRELIMINARIES

2.1 Notation and Basic Properties

2.1.1 Matrix Indexing. For a positive integer $n$, we write $[n] := \{1, 2, \ldots, n\}$ and $[n]_0 := \{0, 1, \ldots, n-1\}$.

By default, we use zero-based numbering for the indices of matrices, meaning, for any set $S$, positive integers $m, n$, matrix $M \in \mathbb{F}^{m \times n}$, $i \in [m]_0$ and $j \in [n]_0$, we write $M[i, j]$ for the corresponding entry of $M$. That said, if $S_n, S_m$ are sets of sizes $|S_n| = n$ and $|S_m| = m$, we may sometimes say that the rows and columns of $M$ are indexed by $S_n$ and $S_m$, respectively. In this case, we implicitly define bijections $f_{S_n} : S_n \to [m]_0$ and $f_{S_m} : S_m \to [n]_0$, and then for $n \in S_n$ and $m \in S_m$ we write $M[f_{S_n}(n), f_{S_m}(m)]$.

2.1.2 Matrix Products.

Definition 2.1. For any field $\mathbb{F}$, positive integers $n_a, n_b, m_a, m_b$, and matrices $A \in \mathbb{F}^{n_a \times m_a}, B \in \mathbb{F}^{n_b \times m_b}$, the Kronecker product of $A$ and $B$, denoted $A \otimes B$, is the matrix $A \otimes B \in \mathbb{F}^{n_a n_b \times m_a m_b}$, whose rows and columns are indexed by $[a_i]_0 \times [b_j]_0$ and $[a_i]_0 \times [b_j]_0$, respectively, and whose entries are given by

$$A \otimes B[(i_A, i_B), (j_A, j_B)] := A[i_A, j_A] \cdot B[i_B, j_B].$$

The Kronecker product is not commutative in general, however, there are always permutation matrices $P \in \{0, 1\}^{(a_i-1) \times (a_i-1)}$ and $P' \in \{0, 1\}^{(a_i-1) \times (a_i-1)}$, which depend only on $a_1, a_2, b_1, b_2$, such that

$$A \otimes B = P \times (B \otimes A) \times P'.$$

For a matrix $A$ and positive integer $n$, we write $A^\otimes n$ to denote the Kronecker product of $n$ copies of $A$, i.e., $A^{\otimes 1} = A$ and $A^{\otimes n} = A^{\otimes (n-1)} \otimes A$.

We will need some additional notation for dealing with more complicated Kronecker products. For positive integers $n, q$, matrices $A, B \in \mathbb{F}^{q \times q}$, and sets $S_A \subseteq [n]$ and $S_B = [n] \setminus S_A$, we write $A^{\otimes S_A} \otimes B^{\otimes S_B}$ for the matrix in $\mathbb{F}^{q^n \times q^m}$ given by, for $i, j \in [q]^n$,

$$A^{\otimes S_A} \otimes B^{\otimes S_B}[i, j] := \left(\prod_{\ell \in S_A} A[i[\ell]], j[\ell]\right) \cdot \left(\prod_{\ell \in S_B} B[i[\ell]], j[\ell]\right).$$

Similarly, if $A \in \mathbb{F}^{q \times q}$ and $B \in \mathbb{F}^{S_A \times q^{S_A}}$ then we write $A^{\otimes S_A} \otimes B^{i_{S_B}}$ for the matrix in $\mathbb{F}^{q^n \times q^m}$ given by, for $i, j \in [q]^n$,

$$A^{\otimes S_A} \otimes B^{i_{S_B}[i, j]} := \left(\prod_{\ell \in S_A} A[i[\ell]], j[\ell]\right) \cdot \left(\prod_{\ell \in S_B} B[i[\ell]], j[\ell]\right).$$

Here, $'i_{S_B}'$ denotes $i$ restricted to the coordinates of $S_B$.

In addition to using $\otimes$ to denote the Kronecker product of matrices, we will use $\times$ to denote the (usual) product of matrices, and for emphasis, we will use $\cdot$ to denote the product of field elements.

2.1.3 Matrix Sparsity and Rigidity. For a matrix $A \in \mathbb{F}^{n_a \times m_a}$, its sparsity, written $\text{nnz}(A)$, denotes number of non-zero entries in $A$. We similarly define its row sparsity, $\text{nnz}_R(A)$, to be the maximum number of non-zero entries in a row of $A$, and its column sparsity, $\text{nnz}_C(A)$, to be the maximum number of non-zero entries in a column of $A$. Some basic properties we will use are that, for any $A \in \mathbb{F}^{n_a \times m_a}$ and $B \in \mathbb{F}^{p_b \times q_b}$,

- $\text{nnz}(A \otimes B) = \text{nnz}(A) \cdot \text{nnz}(B)$,
- $\text{nnz}(A \otimes B) = \text{nnz}_R(A) \cdot \text{nnz}_C(B)$,
- if $a_2 = b_1$ then $\text{nnz}_R(A \times B) \leq \text{nnz}_R(A) \cdot \text{nnz}_C(B)$, and
- if $D \in \mathbb{F}^{\text{nnz}_R(A) \times \text{nnz}_C(A)}$ is a diagonal matrix, then $\text{nnz}(D \times A) \leq \text{nnz}(A)$ and $\text{nnz}(D \times A) \leq \text{nnz}(A)$.

For a matrix $A \in \mathbb{F}^{n_a \times m_a}$ and a nonnegative integer $r$, we write $R_A(r)$ to denote the rank-$r$ rigidity of $A$ over $\mathbb{F}$, which is the minimum number of entries of $A$ which must be changed to other values in $\mathbb{F}$ to make its rank at most $r$. In other words:

$$R_A(r) := \min_{\text{rank}(A+B) \leq r} \text{nnz}(B).$$

The definition of $R_A(r)$ depends on the field $\mathbb{F}$, which we will explicitly mention when it is not clear from context.

We similarly define the rank-$r$ row/column rigidity of $A$, denoted $R^A_r(r)$, to be the minimum number of entries which must be changed per row or column of $A$ to make its rank at most $r$, i.e.

$$R^A_r(r) := \min_{\text{rank}(A+B) \leq r} \max\{\text{nnz}_R(B), \text{nnz}_C(B)\}.$$ 

It follows that, for any positive integer $r$ and any $A \in \mathbb{F}^{n_a \times m_a}$, we have

$$R_A(r) \leq a \cdot R^A_r(r).$$

2.1.4 Important Families of Matrices.

- The Walsh-Hadamard transform $H_n \in \{-1, 1\}^{2^n \times 2^n}$ is defined by

$$H_n := \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

and for $n \in \mathbb{N}$, $H_n = H_{n-1}^\otimes n$.

- The Disjointness matrix $R_n \in \{0, 1\}^{2^n \times 2^n}$ is defined by

$$R_n := \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},$$

and for $n \in \mathbb{N}$, $R_n = R_{n-1}^\otimes n$.

- The Fourier transform $F_N \in \mathbb{C}^{N \times N}$ is defined by picking $\omega_N := e^{2\pi i/N}$ to be a primitive $N$th root of unity, then setting $F_N[i, j] = \omega_N^{ij}$.

- For $k \in \mathbb{N}$ we write $I_k$ to denote the $k \times k$ identity matrix.

- A diagonal matrix $D \in \mathbb{F}^{N \times N}$ is any matrix such that, if $i \neq j$, then $D[i, j] = 0$. $D$ has full rank if and only if $D[i, i] \neq 0$ for all $i$. 

778
• A weighted permutation matrix $\Pi \in \mathbb{F}^{N \times N}$ is a matrix with exactly one nonzero entry in each row or column. A permutation matrix is a weighted permutation matrix in which each nonzero entry is 1.

2.1.5 Arithmetic Circuits and Linear Circuits. An arithmetic circuit over a field $\mathbb{F}$ is a circuit whose inputs are variables and constants from $\mathbb{F}$, and whose gates compute the product or the sum over $\mathbb{F}$ of their inputs. A linear circuit over $\mathbb{F}$ is a circuit whose inputs are variables from $\mathbb{F}$, and whose gates compute $\mathbb{F}$-linear combinations of their inputs. The depth of a circuit is the length (number of edges) of the longest path from an input to an output. The size might either be measured by number of gates, or number of wires.

For a field $\mathbb{F}$ and matrix $A \in \mathbb{F}^{m \times n}$, we say that a circuit $C$ computes the linear transformation $A$ (or simply ‘computes $A$’) if $C$ has $q_1$ inputs and $q_2$ outputs, such that on input $x \in \mathbb{F}^m$, the output of $C$ is $A \cdot x$.

In a synchronous linear circuit, the inputs to each gate must all have the same depth. A synchronous linear circuit $C$ of depth $d$ for a matrix $A$ corresponds to matrices $A_1, \ldots, A_d$ such that $A = \prod_{j=1}^d A_j$, and the size (number of wires) of $C$ is given by $\sum_{j=1}^d \text{nnz}(A_j)$. Any depth-$d$ linear circuit can be converted into a depth-$d$ synchronous linear circuit for the same linear transformation with at most a $O(d)$ multiplicative blow-up in the size. In this paper, $O(d)$ will typically be negligible, so we will focus on synchronous linear circuits.

2.1.6 Binary Entropy Function. The binary entropy function $H : [0, 1] \to [0, 1]$ is defined by

$$H(p) := -p \cdot \log_2(p) - (1-p) \cdot \log_2(1-p),$$

where we take $0 \cdot \log_2(0) = 0$. For every integer $n > 1$ and every $p \in (0, 1)$, it is known that

$$\frac{1}{n+1} 2^n H(p) \leq \left( \frac{n}{p \cdot n} \right) \leq 2^n H(p).$$

We will make use of the following calculations:

Lemma 2.2. For any integer $q > 1$ and any real $0 < \delta < 1/q - 1/(q+1)$ we have:

1. $H(1/q) = \log_2(q) - \frac{q-1}{q} \log_2(q-1)$,
2. $H(1/q+\delta) - H(1/q) \leq \delta \cdot \log_2(q-1) - \delta^2 \cdot \frac{q^2}{(q-1) \log_2(q)} + O(\delta^3)$, and
3. $H(1/q) - H(1/q-\delta) \leq \delta \cdot \log_2(q-1) + \delta^2 \cdot \frac{q^2}{(q-1) \log_2(q)} + O(\delta^3)$.

Proof. (1) is a simple rearrangement of the definition:

$$H(1/q) = \frac{1}{q} \log_2(q) + \frac{q-1}{q} \log_2(q/(q-1)) = \log_2(q) - \frac{q-1}{q} \log_2(q-1).$$

To prove (2), start by writing

$$H(1/q) - H(1/q-\delta) = \int_{1/q-\delta}^{1/q} H'(z) dz \geq \int_{1/q-\delta}^{1/q} \log_2 \left( \frac{1-z}{z} \right) dz.$$

Since $\log((1-z)/z)$ is convex, we can bound this above using the midpoint value by

$$\delta \cdot \log_2 \left( \frac{1-(1/2+\delta/2)}{1/2+\delta/2} \right) dz = \delta \cdot \log_2(q-1) - \delta^2 \cdot \frac{q^2}{(q-1) \log_2(q)} + O(\delta^3),$$

where the last step is the Taylor expansion at $\delta = 0$. Similarly, (3) follows by

$$H(1/q) - H(1/q-\delta) \leq \delta \cdot \log_2(q-1) + \delta^2 \cdot \frac{q^2}{(q-1) \log_2(q)} + O(\delta^3).$$

□

2.2 Basic Tools for Rigidity and Kronecker Products

We now give a number of basic tools which will be of use throughout our proofs.

Proposition 2.3 (The mixed-product property). Let $\mathbb{F}$ be any field, and let $A \in \mathbb{F}^{a_1 \times a_2}, B \in \mathbb{F}^{b_1 \times b_2}, C \in \mathbb{F}^{c_1 \times c_2}, D \in \mathbb{F}^{d_1 \times d_2}$ be any matrices over $\mathbb{F}$ with $a_2 = c_1$ and $b_2 = d_1$. Then, $(A \otimes B) \times (C \otimes D) = (A \times C) \otimes (B \times D)$.

Proposition 2.4. For any field $\mathbb{F}$, any positive integers $a$, $b$, and any matrices $A \in \mathbb{F}^{a \times d}$ and $B \in \mathbb{F}^{b \times d}$, we have $\text{rank}(A \otimes B) = \text{rank}(A) \cdot \text{rank}(B)$.

Proposition 2.5. For any field $\mathbb{F}$, integers $d_1, d_2, d_3, d_4$ and matrices $X_1 \in \mathbb{F}^{d_1 \times d_2}, X_2 \in \mathbb{F}^{d_2 \times d_3}, X_3 \in \mathbb{F}^{d_3 \times d_4}$, and $X_4 \in \mathbb{F}^{d_4 \times d_1}$, we have

$$X_1 \times X_2 + X_3 \times X_4 = (X_1 | X_3) \times \left( \frac{X_2}{X_4} \right).$$

where we are writing $\cdot$ to denote matrix concatenation.

Lemma 2.6. For any field $\mathbb{F}$, positive integers $q, n$, and matrices $M_1, \ldots, M_n \in \mathbb{F}^{q \times q}$, we have

$$\bigotimes_{i=1}^n M_i = \prod_{i=1}^n M_i \otimes (I_q \otimes [n]) \otimes [i].$$

(2)

Proof. We proceed by induction on $n$. The base case $n = 1$ is true since then the right-hand side of Equation (2) is simply equal
Lemma 2.8. For any field $\mathbb{F}$, positive integer $q$, and matrix $M \in \mathbb{F}^{n \times q}$, we say $M$ is an outer-$1$ matrix if, for all $i,j \in \{0,1,\ldots,q-1\}$ with $i = 0$ or $j = 0$ (or both) we have $M[i,j] = 1$. We similarly say $M$ is an outer-$0$ matrix if we have $M[i,j] = 0$ for all such $i,j$, and an outer-nonzero matrix if we have $M[i,j] \neq 0$ for all such $i,j$.

Lemma 2.9. For any field $\mathbb{F}$, positive integers $n,q$, and outer-nonzero matrices $M_1,\ldots,M_n \in \mathbb{F}^{n \times q}$, there are

- outer-$1$ matrices $M'_1,\ldots,M'_n \in \mathbb{F}^{n \times q}$, and
- two invertible diagonal matrices $D,D' \in \mathbb{F}^{n \times q}$, such that $M = D \times M' \times D'$.

Proof. We first define the diagonal matrices $G,G' \in \mathbb{F}^{q \times q}$ by: For $i \in \{0,1,\ldots,q-1\}$, set $G[i,i] = 1/M[i,0]$ and $G'[i,i] = M[0,0]/M[i,0]$. These are well-defined and invertible since $M$ is an outer-nonzero matrix. Let $M' = G \times M \times G'$; we can see that for any $i \in \{0,1,\ldots,q-1\}$ we have $M'[i,0] = M[i,0] \cdot G[i,i] \cdot G'[0,0] \cdot M[i,0]$ and for any $j \in \{0,1,\ldots,q-1\}$ we have $M'[0,j] = M[0,j] \cdot G[0,0] \cdot G'[j,j] = M[0,j] \cdot (1/M[0,0]) \cdot M[0,j]$. Finally, we can pick $D = G^{-1}$ and $D' = G'^{-1}$ so that $M = D \times M' \times D'$.

Lemma 2.10. For any field $\mathbb{F}$, positive integers $q,r$, and matrices $A,B,D,D' \in \mathbb{F}^{q \times q}$ such that $D$ and $D'$ are invertible diagonal matrices with $A = D \times B 	imes D'$, we have that $R_A(r) = R_B(r)$.

Proof. By definition of $R_B(r)$, there are matrices $L,S \in \mathbb{F}^{q \times q}$ such that rank($L$) $\leq r$, $\text{nnz}(S) \leq R_B(r)$, and $B = L + S$. It follows that $A = D \times L \times D' + D \times S \times D'$. Since multiplying on the left or right by a full-rank diagonal matrix does not change the rank or sparsity of a matrix, this expression shows that $R_A(r) \leq R_B(r)$. A symmetric argument also shows that $R_A(r) \geq R_B(r)$ as desired.

The next Lemma, which shows that the product of non-rigid matrices is also non-rigid, was also used by [13, Lemma 2.18].

Lemma 2.11. For any field $\mathbb{F}$, positive integers $q,r$, and matrices $A,B,C \in \mathbb{F}^{q \times q}$ with $D$ a diagonal matrix and $C = A \times D \times B$, we have that

$R_C(2r) \leq R_A(r) \cdot R_B(r)$.

Proof. Let $s_A := R_A(r)$ and $s_B := R_B(r)$. Write $A = L_A + S_A$ and $B = L_B + S_B$ where $L_A,L_B,S_A,S_B \in \mathbb{F}^{q \times q}$ are matrices with rank($L_A$) $\leq r$, rank($L_B$) $\leq r$, $\text{nnz}(S_A) \leq s_A$, $\text{nnz}(S_B) \leq s_B$, and $\text{nnz}(C) \leq s_A \cdot s_B$. We have that $C = (L_A + S_A) \times D \times (L_B + S_B) = L_A \times D \times (L_B + S_B) + S_A \times D \times L_B + S_A \times D \times S_B$.

The first two matrices in the right-hand-side, $L_A \times D \times (L_B + S_B)$ and $S_A \times D \times L_B$, both have rank at most $r$ since $L_A$ and $L_B$ have rank at most $r$. The third, $M := S_A \times D \times S_B$, has both $\text{nnz}(M) \leq \text{nnz}(S_A) \cdot \text{nnz}(S_B)$, $\text{nnz}(C) \leq s_A \cdot s_B$. It follows that $\text{nnz}(C) \leq s_A \cdot s_B$. This expression thus shows that $R_C(2r) \leq s_A \cdot s_B$ as desired.

3 FRAMEWORK FOR DESIGNING SMALL CIRCUITS FROM NON-RIGIDITY

We first note that an upper bound for a fixed matrix in a family of Kronecker products leads to one for the entire family.

Lemma 3.1. For any field $\mathbb{F}$, fixed positive integers $q,t,d$, and matrix $M \in \mathbb{F}^{q \times q}$, suppose $M^{\otimes t} = \prod_{j=1}^{d} B_j$ for matrices $B_j$ for all $i \in [d]$ with $\text{nnz}(B_j) = b_j$. Then, for all positive integers $n$ and $j \in [d]$ there are matrices $A_{n,j}$ with $\text{nnz}(A_{n,j}) < b_j^{1/n^t}$ and $M^{\otimes t} = \prod_{j=1}^{d} A_{n,j}$. If $d$ divides $n$, the upper bound can be further reduced to $\text{nnz}(A_{n,j}) < b_j^{1/n}$. 

780
Proof. Assuming $t$ divides $n$, we will show there are matrices $A_{n,j}$ with $\text{nnz}(A_{n,j}) = b_j^{n/t}$ and $M^{\otimes n} = \prod_{j=1}^d A_{n,j}$. If $t$ does not divide $n$, we can instead apply this construction for the next multiple $n’ > n$ of $t$, and then pick the appropriate submatrix of $M^{\otimes n’}$, to get $M^{\otimes n}$, we will thus have $\text{nnz}(A_{n,j}) = b_j^{n’/t} < b_j^{1+n/t}$.

Now, assuming $t$ divides $n$, then we can simply write $M^{\otimes n} = (\prod_{j=1}^d B_j^{\otimes n/t}) = \prod_{j=1}^d B_j^{\otimes n/t}$, and pick $A_{n,j} = B_j^{\otimes n’/t}$, which has $\text{nnz}(A_{n,j}) = \text{nnz}(B_j^{\otimes n/t}) = b_j^{n/t}$, as desired. □

Next, we observe that rigidity upper bounds can be used to give depth-2 synchronous circuit upper bounds.

**Lemma 3.2.** For any field $\mathbb{F}$, fixed positive integers $r, q$, and matrix $M \in \mathbb{F}^{q \times d}$, there are matrices $B \in \mathbb{F}^{q \times (qr)}$ and $C \in \mathbb{F}^{q \times (qr)q}$ such that $M = B \times C$, $\text{nnz}(B) = q \cdot r + R_M(r)$, and $\text{nnz}(C) = q \cdot (r + 1)$.

Proof. By definition of rigidity, we can write $M = L + S$ for matrices $L, S \in \mathbb{F}^{q \times q}$ with $\text{rank}(L) = r$ and $\text{nnz}(S) = R_M(r)$. In particular, there are matrices $B’ \in \mathbb{F}^{q \times q}$ and $C’ \in \mathbb{F}^{q \times q}$ such that $L = B’ \times C’$. By Proposition 2.5, our desired matrix decomposition is thus

$$M = (S)B’ \times \left(\frac{I_q}{C’}\right).$$

We have $\text{nnz}(B) = \text{nnz}(S) + \text{nnz}(B’) \leq R_M(r) + q \cdot r$, and $\text{nnz}(C) = \text{nnz}(I_q) + \text{nnz}(C’) \leq q + q \cdot r$. □

**Remark 3.3.** Applying Lemma 3.2 to $M^T$ instead of $M$, we can alternatively obtain $B \in \mathbb{F}^{q \times (qr)}$ and $C \in \mathbb{F}^{q \times (qr)q}$ such that $M = B \times C$, $\text{nnz}(B) = q \cdot (r + 1)$, and $\text{nnz}(C) = q \cdot r + R_M(r)$. In other words, we can choose either $B$ or $C$ to have higher sparsity.

Finally, we show how to ‘symmetrize’ the construction of Lemma 3.2 to extend it to small circuits of any depth $d \geq 2$.

**Theorem 3.4.** For any field $\mathbb{F}$, positive integers $r, q$, and matrix $M \in \mathbb{F}^{q \times d}$, let

$$c := \log_q((r + 1) \cdot (r + R_M(r)/q)).$$

Then, for every positive integers $n, d$, setting $N = q^n$, the matrix $M^{\otimes n} \in \mathbb{F}^{N \times N}$ can be written as $M^{\otimes n} = \prod_{j=1}^d A_{n,j}$ for matrices $A_{n,j}$ with $\text{nnz}(A_{n,j}) < q^{1-c} \cdot N^{1+c/d}$. If $d$ divides $n$, the upper bound can be further reduced to $\text{nnz}(A_{n,j}) \leq N^{1+c/d}$.

Proof. Using Lemma 3.2 and Remark 3.3, there are matrices $B, B’, C, C’$ such that $M = B \times C = C’ \times B’$, $\text{nnz}(B) = q \cdot r + R_M(r)$, and $\text{nnz}(C) = \text{nnz}(C’) = q \cdot (r + 1)$. We thus have the following $d$ ways to write $M$ as a product of $d$ matrices:

$$M = B \times C \times I_q \times I_q \times \cdots \times I_q \times I_q \times I_q,$$

$$M = I_q \times B \times C \times I_q \times \cdots \times I_q \times I_q \times I_q,$$

$$M = I_q \times I_q \times B \times C \times \cdots \times I_q \times I_q \times I_q,$$

$$M = I_q \times I_q \times I_q \times I_q \times \cdots \times B \times C,$$

$$M = I_q \times I_q \times I_q \times I_q \times \cdots \times I_q \times I_q \times I_q \times I_q \times I_q.$$

Applying Proposition 2.3, there are thus permutation matrices $P_j, P_j’$ for each $j \in [d]$ such that we can write $M^{\otimes d}$ as:

$$M^{\otimes d} = \left(\prod_{j=1}^d P_j \times (B \otimes C’ \otimes I_{q^{d-2}}) \times P_j’\right)$$

$$\times \left(\prod_{j=2}^{d-1} P_j \times (B \otimes C \otimes I_{q^{d-2}}) \times P_j’\right)$$

$$\times \left(P_1 \times (B’ \otimes C’ \otimes I_{q^{d-2}}) \times P_1’\right).$$

Since $\text{nnz}(B) = \text{nnz}(B’)$ and $\text{nnz}(C) = \text{nnz}(C’)$, this is expressing $M^{\otimes d}$ as a product of $d$ matrices, each of which has sparsity

$$\text{nnz}(B \otimes C \otimes I_{q^{d-2}}) = \text{nnz}(B) \cdot \text{nnz}(C) \cdot \text{nnz}(I_{q^{d-2}})$$

$$= (q \cdot r + R_M(r)) \cdot (q \cdot (r + 1)) \cdot q^{d-2}.$$

Assume first that $d$ divides $n$. Applying Lemma 3.1, it follows that the matrix $M^{\otimes n}$ can be written as $M^{\otimes n} = \prod_{j=1}^d A_{n,j}$ for matrices $A_{n,j}$ with

$$\text{nnz}(A_{n,j}) \leq ((q \cdot r + R_M(r)) \cdot (q \cdot (r + 1)) \cdot q^{d-2}) \cdot n/d$$

$$= q^n \cdot (r + R_M(r)/q)^n \cdot (r + 1)^n/d$$

$$= q^n \cdot \left(1 + \log_q((r + 1) \cdot (r + R_M(r)/q))\right)^n/d$$

$$= N^{1+\frac{c}{d}},$$

where $N = q^n$ so that $M^{\otimes n} \in \mathbb{F}^{N \times N}$, and $c := \log_q((r + 1) \cdot (r + R_M(r)/q))$, as desired.

Next, consider when $d$ does not divide $n$. Let $n’$ be the largest integer less than $n$ such that $d$ divides $n’$. By the above argument, there are matrices $A_{n’,1}, \ldots, A_{n’,d}$ such that $M^{\otimes n’} = \prod_{j=1}^d A_{n’,j}$ and $\text{nnz}(A_{n’,j}) \leq q^{n’} \cdot (1+c/d)$. For each $1 \leq \ell \leq k$ we can also write $M = \prod_{j=1}^d (I_{q^{k-1}} \times M : I_q)$. Combining these $k + 1$ expressions together, again using Proposition 2.3, it follows that there are permutation matrices $P_j, P_j’$ for each $j \in [d]$ such that

$$M^{\otimes n} = \left(\prod_{j=1}^k P_j \times (A_{n’,j} \otimes M \otimes I_{q^{k-1}}) \times P_j’\right)$$

$$\times \left(\prod_{j=k+1}^d P_j \times (A_{n’,j} \otimes I_q) \times P_j’\right).$$
We can calculate that \( \text{nnz}(A_{n',j} \otimes M \otimes I_{q^{k-1}}) \leq q^{n'-(1+c/d)k+1} < q^{(1-c)n(1+c/d)} \), and similarly \( \text{nnz}(A_{n',j} \otimes I_{q^k}) < q^{(1-c)n(1+c/d)} \), which concludes the proof like before.

In the proof of Theorem 3.4, we made use of Remark 3.3 that our fixed upper bound from non-rigidity can be made symmetric. For fixed upper bounds designed in other ways, this may not be the case. In the full version of the paper, we will nonetheless show that any nontrivial fixed upper bound can be used to prove a result similar to Theorem 3.4. For now, in this section and the next, we will focus specifically on our upper bounds from non-rigidity.

**Corollary 3.5.** For any field \( \mathbb{F} \), positive integers \( r, q \), and matrix \( M \in \mathbb{F}^{q \times d} \), let

\[
\begin{equation}
\begin{split}
c := \log_q((r + 1) \cdot (r + R_M(r)/q)).
\end{split}
\end{equation}
\]

Then, for every positive integers \( n, d \), with \( d < q(n) \), setting \( N = q^n \), the matrix \( M \otimes I_{q^n} \in \mathbb{F}^{N \times N} \) has a synchronous linear circuit of size \((1 + o(1)) \cdot d \cdot q^{n(1+c/d)}\).

**Proof.** Let \( n' \) be the integer in the range \( n \geq n' > n - d \) such that \( d \) divides \( n' \), and let \( k = n - n' \). Applying Theorem 3.4 to \( M^{\otimes n'} \), we see that it has a synchronous circuit of size \( d \cdot q^{n'-(1+c/d)} \cdot q^k = d \cdot q^{n(1+c/d)}/q^{k-c/d} \). Next, again by applying Theorem 3.4, but this time for depth \( k \), we see that \( M^{\otimes k} \) has a synchronous circuit of size \( k \cdot q^{k+c} \), and so \( M^{\otimes n'} \otimes M^{\otimes k} \) has a synchronous circuit of size \( q^k \cdot k \cdot q^{n+c} = k \cdot q^{n+c} \). Hence, since \( M^{\otimes n} = M^{\otimes n'} \otimes M^{\otimes k} = (M^{\otimes n'} \otimes I_{q^k}) \otimes (I_{q^k} \otimes M^{\otimes k}) \), it follows that \( M^{\otimes n} \) has a synchronous circuit of size

\[
\begin{equation}
\begin{split}
d \cdot q^{n(1+c/d)}/q^{k-c/d} + k \cdot q^{n+c} = q^{n(1+c/d)} \cdot \left( \frac{d}{q^{k-c/d}} + \frac{k}{q^{c+n(d-1)}} \right) \leq (1 + o(1)) \cdot d \cdot q^{n(1+c/d)}.
\end{split}
\end{equation}
\]

**Corollary 3.6.** For any field \( \mathbb{F} \), positive integers \( r, q \), and matrix \( M \in \mathbb{F}^{q \times d} \), let

\[
\begin{equation}
\begin{split}
c := \log_q((r + 1) \cdot (r + R_M(r)/q)).
\end{split}
\end{equation}
\]

Then, for every positive integer \( n \), setting \( N = q^n \), the matrix \( M^{\otimes n} \in \mathbb{F}^{N \times N} \) has a synchronous linear circuit of size \((c \cdot e \cdot \log_e(2) + o(1)) \cdot N \cdot \log_2(N)\).

**Theorem 4.3.** For any field \( \mathbb{F} \), matrix \( M \in \mathbb{F}^{2^d \times 2^d} \), and positive integers \( d, n > 1 \), the matrix \( M^{\otimes n} \in \mathbb{F}^{N \times N} \) for \( N = 2^n \) has a depth-\( d \) linear circuit of size \( 2^e \cdot N^{1+(1-c)/d} \) for some constant \( e > 0.01526 \).

**Proof.** Applying Theorem 3.4 with \( M^{\otimes q} \), \( q = 8 \), and \( r = 1 \), combined with the rigidity bound of Lemma 4.2, shows that \( M^{\otimes n} \) has a depth-\( d \) linear circuit of size \( 2^{c' \cdot N^{1+c/d}} \) for

\[
\begin{equation}
\begin{split}
c = \log_q \left( (r + 1) \cdot (r + R_M(r)/q) \right)
\end{split}
\end{equation}
\]

\[
\begin{equation}
\begin{split}
\leq \log_8 \left( 2 \cdot \left( 1 + \frac{23}{8} \right) \right) = 0.98474 = 1 - e.
\end{split}
\end{equation}
\]
Corollary 4.4. For any field $F$, matrix $M \in \mathbb{F}^{2 \times 2}$, and positive integer $n > 1$, the matrix $M^{\otimes n} \in \mathbb{F}^{N \times N}$ for $N = 2^n$ has a synchronous linear circuit of size $(1 - \epsilon) \cdot \epsilon \log_2(2) + o(1)) \cdot N \log_N$ for some constant $\epsilon > 0.01526$.

Proof. Apply Corollary 3.6 with the same rigidity bound of Lemma 4.2. □

4.2 Walsh-Hadamard Transform

Lemma 4.5. Over any field $F$ with $\text{ch}(F) \neq 2$, we have $\mathcal{R}_{2 \times 2}(1) = 4$.

Proof. First, to see that $\mathcal{R}_{2 \times 2}(1) \leq 4$, we write $H_2$ as

$$
H_2 = \begin{bmatrix}
H_1 & H_1 \\
-H_1 & H_1
\end{bmatrix}.
$$

Each copy of $H_1$ has rank 2, so we must change at least one entry in each $H_1$ to drop the rank of the whole matrix to 1. Since there are four disjoint copies, we must change at least four entries. □

Lemma 4.6. Over any field $F$, we have $\mathcal{R}_{2 \times 2}(1) \leq 22$.

Proof. We use the same construction as in Lemma 4.2, with $\omega = -1$ so that $M^{\otimes 3} = H_3$. In this case, there is one more correct entry than in the general case, since when $x = y = (1, 1, 1)$, we have $M^{\otimes 3}[x, y] = \omega$ and $L[x, y] = \omega$, but these are equal when $\omega = -1$, so the number of errors is only $23 - 1 = 22$. □

Lemma 4.7. Over any field $F$, we have $\mathcal{R}_{4 \times 4}(1) \leq 96$.

Proof. In the proof of Lemma 4.5, we showed there is a matrix $A \in \{-1, 1\}^{4 \times 4}$ which differs from $H_2$ in 4 entries, and which has rank 1 over any field. Let $B = A^{\otimes 2} \in \{-1, 1\}^{16 \times 16}$. We have that $\text{rank}(B) = \text{rank}(A^2) = 1$. Indexing the rows and columns of $H_2$ by $\{0, 1, 2, 3\}$, and the rows and columns of $H_4$ by $\{0, 1, 2, 3\}^2$, we see that for $a, b, c, d \in \{0, 1, 2, 3\}$ we have

$$
\frac{B[\{a, b\}, \{c, d\}]}{H_4[\{a, b\}, \{c, d\}]} = \frac{A[a, c] \cdot A[b, d]}{H_2[a, c] \cdot H_2[b, d]}.
$$

This will equal 1 (and hence the $(a, b, c, d)$ entries of $B$ and $H_4$ will be equal) whenever either:

- $A[a, c] = H_2[a, c]$ and $A[b, d] = H_2[b, d]$, which happens for $(16 - 4^2) = 144$ values of $a, b, c, d \in \{0, 1, 2, 3\}$, or
- $A[a, c] \neq H_2[a, c]$ and $A[b, d] \neq H_2[b, d]$ (since all these values are in $[-1, 1]$), which happens for $4^2 = 16$ values of $a, b, c, d \in \{0, 1, 2, 3\}$.

Thus, $B$ only differs from $H_4$ in $16^2 - 144 - 16 = 96$ entries, as desired. □

Remark 4.8. I verified using a brute-force search that Lemma 4.6 and Lemma 4.7 are tight over any field $F$ with $\text{ch}(F) \neq 2$. I unfortunately haven’t found more enlightening proofs of these facts.

Theorem 4.9. For any field $F$ and positive integers $d, n > 1$, the matrix $H_n \in \mathbb{F}^{N \times N}$ for $N = 2^n$ has a depth-$d$ linear circuit of size $\leq 2^n \cdot N^{1+((1-\epsilon) \cdot \epsilon \log_2(2) + o(1))} \cdot N \log_N$ for some constant $\epsilon > 0.04816$.

Proof. Applying Theorem 3.4 with $H_4$, $q = 16$, and $r = 1$, combined with the rigidity bound of Lemma 4.7, shows that $H_n = H_n^{\otimes n}$ has a depth-$d$ linear circuit of size $2^n \cdot N^{1+\epsilon/c} \cdot N^{1+\epsilon/d}$ for

$$
c = \log_q \left( (r + 1) \cdot \frac{\mathcal{R}_{d}(r)}{q} \right)
\leq \log_{16} \left( 2 \cdot \left( 1 + \frac{96}{16} \right) \right)
< 0.95184 = 1 - \epsilon.
$$

□

Corollary 4.10. For any field $F$ and positive integer $n > 1$, the matrix $H_n \in \mathbb{F}^{N \times N}$ for $N = 2^n$ has a synchronous linear circuit of size $(1 - \epsilon) \cdot \epsilon \log_2(2) + o(1)) \cdot N \log_N$ for some constant $\epsilon > 0.04816$.

Proof. Apply Corollary 3.6 with the same rigidity bound of Lemma 4.7. □

4.3 Fourier Transform

In order to use the approach of Theorem 3.4 to prove that the $N \times N$ Fourier transform matrix $F_N$ has depth-$d$ circuits of size $O(N^{1+\epsilon}/d)$ for some $c < 1$, we would need it to be the case that, for some positive integers $N > r > 0$, we have

$$\log_N((r + 1) \cdot (r + \mathcal{R}_{F_N}(r)/N)) < 1.$$

We next remark that known rigidity lower bounds for $F_N$ show that this is never the case. In fact, the proof extends to any Vandermonde matrix.

Proposition 4.11. For any positive integers $N > r > 0$, the $N \times N$ Fourier transform matrix $F_N$ has

$$(r + 1) \cdot (r + \mathcal{R}_{F_N}(r)/N) \geq N.$$

Proof. Shparlinski [29] shows that $\mathcal{R}_{F_N}(r) \geq (N - r)^2/r(r + 1)$; for completeness, we prove this below in Lemma 4.13. It then follows that

$$\frac{(r + 1) \cdot (r + \mathcal{R}_{F_N}(r)/N)}{N} \geq \frac{(N - r)^2}{(r + 1) \cdot N},$$

$$= \frac{1}{N} \left( N^2 + r^2 + Nr(r - 1) \right),$$

$$\geq \frac{1}{N} \left( N^2 \right),$$

$$= N.$$

We next prove a Lemma which we will need in the proof of Shparlinski’s rigidity lower bound.

Lemma 4.12. For any positive integers $N > r > 0$, any integer $0 \leq k < n - r$, and any $S \subseteq \{0\}^k$ of size $|S| = r$, let $M_{k,S}$ be the $r \times r$ submatrix of $F_N$ consisting of the rows of $\{k, k+1, k+2, \ldots, k+r-1\}$ and the columns of $S$. Then, $M_{k,S}$ has full rank.
Proof. Indexing the rows of $M_{k,s}$ by $[r]_0$ and the columns by $S$, we have for $j \in [r]_0$ and $s \in S$ that $M_{k,s}[j,s] = (\omega_N^s)^j$, where $\omega_N = e^{2\pi i/N} \in \mathbb{C}$ is a primitive $N$th root of unity. Assume to the contrary that $M_{k,s}$ does not have full rank. Thus, there is a nontrivial linear combination of its rows summing to zero. This means that there are $a_0, a_1, \ldots, a_{r-1} \in \mathbb{C}$, which are not all 0, such that, for each $s \in S$, we have

$$\sum_{j=0}^{r-1} a_j \cdot (\omega_N^s)^j = 0.$$ 

In other words, the $r$ different values $(\omega_N^s \mid s \in S)$ are all roots of the polynomial $p(z) = \sum_{j=0}^{r-1} a_j \cdot z^j$. However, $p$ is a nonzero polynomial of degree at most $r-1$, so it cannot have $r$ roots, a contradiction. \qed

Lemma 4.13 ([29]). For any positive integers $N > r \geq 0$, we have $\text{rank}(F_N) \geq (N-r)^2/(r+1)$.

Proof. Suppose that one can change $t$ entries of $F_N$ to make its rank at most $r$. For $k \in [N-r]_0$, let $t_k$ be the number of changes which are in rows $\{k,k+1,k+2,\ldots,k+r\}$. Since each change contributes to at most $r + 1$ of the $t_k$ values, we have that $\sum_{k=0}^{N-r-1} t_k \leq (r+1) \cdot t$. Thus, by the pigeonhole principle, there must be a $k' \in [N-r]_0$ such that $t_{k'} \leq (r+1) \cdot t/(N-r)$. Let $S \subseteq [N]_0$ be the columns of $F_N$ such that none of the changes in rows $\{k',k'+1,k'+2,\ldots,k'+r\}$ is in a column of $S$. It must be that $|S| \leq r$, since otherwise, by Lemma 4.12, the matrix $M_{k',S}$ has rank $r+1$ and we did not make any changes to it. On the other hand, by definition, $|S| \leq N - t_{k'} \leq N - (r+1) \cdot t/(N-r)$. It follows that $r \geq N - (r+1) \cdot t/(N-r)$, which rearranges to the desired $t \geq (N-r)^2/(r+1)$. \qed

4.4 Disjointness

Recall the Disjointness matrix $R_{N} \in \mathbb{F}^{N \times N}$ from Section 2.1.4. The approach of Theorem 3.4 can be used to prove that $R_{N}$ has depth-$d$ linear circuits of size $N^{1+O(1/c^d)}$. However, since $R_{N}$ is very sparse (it has $\text{nnz}(R_{N}) = 3^d \leq N^{1+O(1)}$) immediately that it has depth-$d$ circuits of size $O(N^{1+c/d})$ for $c = \log_{2}(1+\sqrt{2}) < 0.585$. In fact, using a construction of Jukna and Sergeev [16], we can do even better than this, improving to $c < 0.5432$. We give the construction in the remainder of this section.

Lemma 4.14 ([16, Lemma 4.2]). Let $t = \log_{2}(1 + \sqrt{2}) < 1.28$. For any field $\mathbb{F}$ and positive integer $n$, there are matrices $A_n, B_n \in \mathbb{F}^{2^n \times 2^n}$ with $\text{nnz}(A_n), \text{nnz}(B_n) \leq O(2^{t \cdot n})$ such that $R_n = A_n \times B_n$.

Proof. We show how to partition the 1s of $R_n$ into squares (all-1s combinatorial rectangles with the same number of rows and columns) and rectangles (all-1s combinatorial rectangles with twice as many rows as columns). Our partition is defined recursively. Let $s_n$ be the sum of the side-lengths of the squares in the partition of $R_n$, and let $r_n$ be the sum of the shorter side-lengths of the rectangles. For

\[
R_1 := \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},
\]

we can see that $s_1 = r_1 = 1$. Next, from the recursive definition

\[
R_n := \begin{bmatrix} R_{n-1} & R_{n-1} \\ R_{n-1} & 0 \end{bmatrix},
\]

we see that the three copies of any $s \times s$ square in $R_{n-1}$ can be partitioned into a $s \times s$ square and a $2s \times s$ rectangle in $R_n$, and the three copies of any $2s \times s$ rectangle in $R_{n-1}$ can be partitioned into a $2s \times s$ rectangle and a $2s \times 2s$ square in $R_n$. It follows that we get the recurrence

\[
\begin{bmatrix} s_n \\ r_n \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} s_{n-1} \\ r_{n-1} \end{bmatrix}.
\]

Since the matrix $\begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$ has eigenvalues $1 \pm \sqrt{2}$, it follows that $s_n, r_n \leq O((1 + \sqrt{2})^n)$. We have thus written the 1s of $R_n$ as a disjoint sum of combinatorial rectangles whose side-lengths sum to $O((1 + \sqrt{2})^n) = O(2^t \cdot n)$, from which the result follows. \qed

Following the same construction as Theorem 3.4, we get: Proposition 4.15. For any field $\mathbb{F}$ and any positive integers $n,d$, let $N = 2^n$ and let $c = 2(\log_{2}(1+\sqrt{2}) - 1) < 0.5432$. There are $d$ matrices $A_{n,1},\ldots,A_{n,d}$ such that $R_n = \prod_{j=1}^{d} A_{n,j}$ and $\text{nnz}(A_{n,j}) \leq O(N^{1+c/d})$ for all $j \in [d]$.

ACKNOWLEDGMENTS

I would like to thank Amol Aggarwal, Chi-Ning Chou, Ben Edelman, Alexander Golovnev, DD Liu, Jon Schneider, Leslie Valiant, Virginia Vassilevska Williams, and Ryan Williams for helpful discussions throughout this project, and anonymous reviewers for many helpful comments.

This work is supported by a Michael O. Rabin postdoctoral fellowship.

REFERENCES

[1] Amir Abboud, Ryan Williams, and Huacheng Yu. 2014. More applications of the polynomial method to algorithm design. In Proceedings of the twenty-sixth annual ACM-SIAM Symposium on Discrete Algorithms (SODA). SIAM, 218–230.
[2] Josh Alman, Timothy M. Chan, and Ryan Williams. 2016. Polynomial representations of threshold functions and algorithmic applications. In 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 467–476.
[3] Josh Alman and Lijie Chen. 2019. Efficient construction of rigid matrices using an NP oracle. In 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 1034–1055.
[4] Josh Alman and Ryan Williams. 2015. Probabilistic polynomials and hamming nearest neighbors. In 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 136–150.
[5] Josh Alman and Ryan Williams. 2017. Probabilistic rank and matrix rigidity. In Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC) 641–652.
[6] Josh Alman and Virginia Vassilevska Williams. 2021. A refined laser method and faster matrix multiplication. In Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA). SIAM, 522–539.
[7] Peter Bürgisser and Martin Lotz. 2004. Lower bounds on the bounded coefficient complexity of bilinear maps. Journal of the ACM (JACM) 51, 3 (2004), 464–482.
[8] Bernard Chazelle. 1994. A spectral approach to lower bounds. In Proceedings 35th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 674–682.
[9] Don Coppersmith. 1982. Rapid multiplication of rectangular matrices. SIAM J. Comput. 11, 3 (1982), 467–473.
[10] Ronald De Wolf. 2006. Lower bounds on matrix rigidity via a quantum argument. In International Colloquium on Automata, Languages, and Programming (ICALP). Springer, 62–71.
[11] Zeev Dvir and Benjamin L. Edelman. 2019. Matrix Rigidity and the Croot-Lev-Pach Lemma. Theory of Computing 15, 8 (2019), 1–7.
[12] Zeev Dvir, Alexander Golovnev, and Omri Weinstein. 2019. Static data structure lower bounds imply rigidity. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC). 967–978.
[13] Zeev Dvir and Allen Liu. 2019. Fourier and Circulant Matrices Are Not Rigid. In 34th Computational Complexity Conference (CCC). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[14] Anna Gál, Kristoffer Arnsfelt Hansen, Michal Koucký, Pavel Pudlák, and Emanuele Viola. 2012. Tight bounds on computing error-correcting codes by bounded-depth circuits with arbitrary gates. In Proceedings of the forty-fourth annual ACM Symposium on Theory of Computing (STOC). 479–494.

[15] Xiaohan Huang and Victor Y Pan. 1998. Fast rectangular matrix multiplication and applications. Journal of complexity 14, 2 (1998), 257–299.

[16] Stasys Jukna and Igor Sergeev. 2013. Complexity of linear Boolean operators. Foundations and Trends® in Theoretical Computer Science 9, 1 (2013), 1–123.

[17] François Le Gall. 2014. Powers of tensors and fast matrix multiplication. In Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC). 296–303.

[18] Satyanarayana V Lokam. 2000. On the rigidity of Vandermonde matrices. Theoretical Computer Science 237, 1-2 (2000), 477–483.

[19] Satyanarayana V Lokam. 2001. Spectral methods for matrix rigidity with applications to size–depth trade-offs and communication complexity. J. Comput. System Sci. 63, 3 (2001), 449–473.

[20] Satyanarayana V Lokam. 2009. Complexity lower bounds using linear algebra. Foundations and Trends® in Theoretical Computer Science 4, 1–2 (2009), 1–155.

[21] Gauts Midrijanis. 2005. Three lines proof of the lower bound for the matrix rigidity. arXiv preprint cs/0506081 (2005).

[22] Jacques Morgenstern. 1973. Note on a lower bound on the linear complexity of the fast Fourier transform. Journal of the ACM (JACM) 20, 2 (1973), 305–306.

[23] Sivaramakrishnan Natarajan Ramamoorthy and Cyrus Rashtchian. 2020. Equivalence of Systematic Linear Data Structures and Matrix Rigidity. In 11th Innovations in Theoretical Computer Science Conference (ITCS). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[24] Noam Nisan and Avi Wigderson. 1996. Lower bounds on arithmetic circuits via partial derivatives. Computational complexity 6, 3 (1996), 217–234.

[25] Pavel Pudlák. 1994. Communication in bounded depth circuits. Combinatorica 14, 2 (1994), 203–216.

[26] Pavel Pudlák. 2000. A note on the use of determinant for proving lower bounds on the size of linear circuits. Information processing letters 74, 5-6 (2000), 197–201.

[27] Jaikumar Radhakrishnan and Amnon Ta-Shma. 2000. Bounds for dispersers, extractors, and depth-two superconcentrators. SIAM Journal on Discrete Mathematics 13, 1 (2000), 2–24.

[28] Ran Raz. 2002. On the complexity of matrix product. In Proceedings of the thirty-fourth annual ACM Symposium on Theory of Computing (STOC). 144–151.

[29] Igor E. Shparlinski. 1999. (1999). Private communication, cited in [18].

[30] Volker Strassen. 1969. Gaussian elimination is not optimal. Numerische mathematik 13, 4 (1969), 354–356.

[31] Leslie G. Valiant. 1977. Graph-theoretic arguments in low-level complexity. In International Symposium on Mathematical Foundations of Computer Science (MFCS). Springer, 162–176.

[32] Charles F Van Loan. 2000. The ubiquitous Kronecker product. Journal of computational and applied mathematics 123, 1-2 (2000), 85–100.

[33] Ben Lee Volk and Mrinal Kumar. 2021. Lower Bounds for Matrix Factorization. computational complexity 30, 1 (2021), 1–40.

[34] Ryan Williams. 2014. Faster all-pairs shortest paths via circuit complexity. In Proceedings of the forty-sixth annual ACM Symposium on Theory of Computing (STOC). 664–673.

[35] Virginia Vassilevska Williams. 2012. Multiplying matrices faster than Coppersmith-Winograd. In Proceedings of the forty-fourth annual ACM Symposium on Theory of Computing (STOC). 887–898.