Limitations of Lower-Bound Methods for the Wire Complexity of Boolean Operators

Andrew Drucker[*]

Abstract: We study the circuit complexity of Boolean operators, i.e., collections of Boolean functions defined over a common input. Our focus is the well-studied model in which arbitrary Boolean functions are allowed as gates, and in which a circuit's complexity is measured by its depth and number of wires. We show sharp limitations of several existing lower-bound methods for this model.

First, we study an information-theoretic lower-bound method due to Cherukhin, that yields bounds of form $\Omega_d(n \cdot \lambda_{d-1}(n))$ on the number of wires needed to compute cyclic convolutions in depth $d \geq 2$. This was the first improvement over the lower bounds provided by the well-known superconcentrator technique (for $d = 2, 3$ and for even $d \geq 4$). Cherukhin's method was formalized by Jukna as a general lower-bound criterion for Boolean operators, the "Strong Multiscale Entropy" (SME) property. It seemed plausible that this property could imply significantly better lower bounds by an improved analysis. However, we show that this is not the case, by exhibiting an explicit operator with the SME property that is computable in depth d with $\mathcal{O}(n \cdot \lambda_{d-1}(n))$ wires, for $d = 2, 3$ and for even $d \geq 6$.

Next, we show limitations of two simpler lower-bound criteria given by Jukna: the "entropy method" for general operators, and the "pairwise-distance method" for linear operators. We show that neither method gives super-linear lower bounds for depth 3. In the process, we obtain the first known polynomial separation between the depth-2 and depth-3 wire complexities for an explicit operator. We also continue the study (initiated by Jukna) of the complexity of "representing" a linear operator by bounded-depth circuits, a weaker notion than computing the operator.

[*] MIT CSAIL. Email: adrucker@mit.edu. This work was supported by a DARPA YFA grant of Scott Aaronson. Part of this work was done while visiting the Computer Science Division at UC Berkeley.

Contents

1 Introduction 3
1.1 Circuits with arbitrary gates . . . 3
1.2 The Strong Multiscale Entropy Method . . . 4
1.3 Two simpler lower-bound methods . . . 5
1.4 Our contributions . . . 6
1.4.1 Limitations of entropy-based methods . . . 6
1.4.2 Results on linear transformations . . . 7

2 Preliminaries 8
2.1 Wire complexity of operators . . . 8
2.2 Representing linear operators relative to different bases . . . 9
2.3 A hashing lemma . . . 9

3 Entropy and circuit lower bounds 10
3.1 Entropy of operators . . . 10
3.2 Strong Multiscale Entropy . . . 11

4 Limitations of the SME lower-bound criterion 12
4.1 The DIR operator . . . 13
4.2 Establishing the SME property for DIR . . . 14
4.3 Efficient bounded-depth circuits for DIR . . . 15

5 Limits of Jukna's entropy method, and a separation of depths 2 and 3 22

6 Representing random linear operators 24

7 Tightness of Jukna's pairwise-distance lower bound for depth 2 26

8 The pairwise-distance method fails for depth 3 29

9 Easy bases for representing linear operators 30

1 Introduction

1.1 Circuits with arbitrary gates

This paper continues the study of the circuit complexity of Boolean *operators*, that is, functions $F : \{0,1\}^n \to \{0,1\}^m$. For ease of discussion, we will focus on the common setting $m = n$. Typically we regard $F = (f_1, \ldots, f_n)$ as a collection of n Boolean functions. By comparing the circuit complexity of F to the individual complexities of the f_i's, we are asking: how much easier is it to compute the f_i's together than to compute them separately?

A great deal of work, e.g., in [Val76, Val77, DDPW83, CFL83, CFL85, Pud94, PR94, RS03, Che08a, Juk10a, Juk10b, JS10], has studied the circuit model in which unbounded fanin is allowed, and in which circuit gates can apply *arbitrary* Boolean functions to their inputs. In this model, we study the number of *wires* required in such a circuit to compute F, a quantity we denote as $s(F)$.

While allowing gates to compute arbitrary Boolean functions is physically unrealistic, there are a number of motivations to study this model. First, it arguably provides a natural measure of the "information complexity" of Boolean operators. Second, lower bounds in this strong circuit model are highly desirable, since they also apply to a variety of more realistic models. Third, several natural circuit lower-bound criteria apply even to circuits with arbitrary gates, and it seems worthwhile to understand how far techniques of this kind can carry us. Finally, for at least one important class of Boolean operators—the $\mathbb{F}_2$-linear operators, naturally computable by $\mathbb{F}_2$-linear circuits—it remains unclear whether allowing arbitrary gates in our circuits even confers additional power.

Any individual Boolean function can by trivially computed with n wires in the arbitrary-gates model, so n^2 wires always suffice to compute an operator F. In general, this is not far from optimal: random (non-linear) operators require $\Omega(n^2)$ wires to compute [JS10]. Thus random collections of Boolean functions are, in a sense, "computationally orthogonal" to one another. It would be extremely interesting to identify an *explicit* function collection with this property; however, proving a super-linear lower bound $s(F) = \omega(n)$ for an explicit operator F is a long-standing open problem.

This has led researchers to consider circuits with arbitrary gates but restricted *depth*. Even depth-2 circuits in this model are powerful, and their study was strongly motivated by work of Valiant [Val77] (see [Vio09]), who showed that any operator with depth-2 wire complexity $\omega(n^2/\ln\ln n)$ also cannot be computed by linear-size, logarithmic-depth Boolean circuits (of fanin 2). However, the known lower bounds for depth 2 are too weak to apply Valiant's results. For depth-2 circuits, the best bounds for explicit operators are of form $\Omega(n^{3/2})$ [Che08a, Juk10a]. For depths 3 and 4, the best bounds are $\Omega(n \ln n)$ and $\Omega(n \ln\ln n)$ respectively [Che08a]; for higher constant depths the known bounds (described in Section 1.2) are barely super-linear [DDPW83, Pud94, Che08a].

One might suspect that the difficulty of proving strong lower bounds stems from the unrealistic strength of the circuit model being studied. A seemingly much more modest aim is to prove lower bounds in the *linear circuit* model over $\mathbb{F}_2$. In this model, we require the circuit gates to compute $\mathbb{F}_2$-linear functions; we again allow unbounded fanin. Given some linear operator $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$, we let $s^{\oplus}(L)$ denote the number of wires needed to compute L with a linear circuit. Lupanov [Lup56] (and later Bublitz [Bub86]) showed that $s^{\oplus}(L) = \mathcal{O}(n^2/\ln n)$, and that this bound is tight if L is chosen randomly.

Unfortunately, the known lower bounds for *explicit* linear operators in the linear circuit model are just as discouragingly weak as for operators in the arbitrary-gates model. Moreover, since the lower bounds quoted earlier were shown for non-linear operators, the situation is actually slightly *worse* in the linear case: for example, for depth-2 circuits, the best known lower bound for an explicit linear operator is $\Omega\left(n\left(\frac{\ln n}{\ln\ln n}\right)^2\right)$, proved very recently [GHK+11].

Thus, it is a major unmet challenge to develop lower-bound techniques that effectively exploit the specific behavior of linear circuits.[1] In fact, it is an open question whether $s^{\oplus}(L)$ can be noticeably larger than $s(L)$, that is, whether non-linear gates can ever help us compute linear operators more efficiently. However, we also cannot rule out the possibility that *all* linear operators L are computable by depth-2, non-linear circuits of size $\mathcal{O}(n \cdot \mathrm{polylog}(n))$; see [JS10]. (We will at least prove, in Section 6, that $s(L) = \Omega(n \ln n)$ for random L.)

Since there are relatively few lower-bound methods for circuits with arbitrary gates, it is important to understand the power and limitations of existing methods. In this paper we focus on three such methods.

1.2 The Strong Multiscale Entropy Method

The first method we study was developed by Cherukhin [Che08a], and used to obtain the best known explicit lower bounds on bounded-depth wire complexity. The bounds apply to the cyclic convolution operator over $\mathbb{F}_2^n$, and are of form $\Omega_d\left(n \cdot \lambda_{d-1}(n)\right)$ for depth $d > 1$.[2] Here, $\lambda_d(n)$ is an unbounded function in n, which grows ever-more-slowly as d increases; its growth is extremely slow even for modest values of d. We have[3]

$$\lambda_1(n) \ = \ \Theta(\sqrt{n}) \ , \quad \lambda_2(n) \ = \ \Theta(\ln n) \ , \quad \lambda_3(n) \ = \ \Theta(\ln\ln n) \ ,$$

and for higher d, $\lambda_d(n) = \lambda^*_{d-2}(n)$. The precise definition is in Section 3.2.

The longstanding previous best lower bounds for explicit operators (including cyclic convolution) were of form $\Omega\left(\frac{n\ln^2 n}{\ln\ln n}\right)$ for depth 2 [RTS00] and $\Omega_d\left(\lambda_d(n)\right)$ for $d \geq 3$ [DDPW83, Pud94, AP94], and were based on the *superconcentrator technique* [Val76, Val77]. For depths 2, 3 and for even depths $d \geq 4$, Cherukhin's work gives asymptotic improvements on these older bounds; for odd depths $d \geq 5$, his bounds match the best previous ones from [Pud94]. Cherukhin's lower-bound method does not apply to linear operators. (For $d \geq 3$, the best known lower bounds for computing an explicit linear operator are of form $\Omega_d\left(n \cdot \lambda_d(n)\right)$ [Pud94, p. 215], [GHK+11]. These bounds, along with the $\Omega\left(n\left(\frac{\ln n}{\ln\ln n}\right)^2\right)$ bound for depth 2 from [GHK+11], are valid against circuits with arbitrary gates.)

[1] A lower-bound criterion specific to linear circuits, based on *matrix rigidity*, has been given by Valiant [Val77]. In principle this method is capable of showing strong lower bounds. However, except for some limited success in depth 2 [Pud94], no one has proved sufficiently-strong rigidity lower bounds on explicit $\mathbb{F}_2$-matrices to imply circuit lower bounds in this way. See [Lok09] for a survey of this line of work.

[2] Cherukhin proved his result for depths 2 and 3 earlier in [Che08b]. The paper [Che08a] contains a unified proof for all constant depths.

[3] The $\lambda_d(\cdot)$ functions are defined differently in [Pud94, GHK+11]. We follow [RS03, Che08a, Juk12] instead, and we have converted the bounds quoted from other papers to match our convention.

Cherukhin's method, developed specifically for the convolution operator, was later formulated by Jukna [Juk12, Chap. 13] as a general property of operators that yields a lower bound of form $\Omega_d(n \cdot \lambda_{d-1}(n))$. This operator property is called the *Strong Multiscale Entropy (SME)* property. Very roughly speaking, the SME property states that there is a large "information flow" between many subsets of the input and output coordinates of an operator. The precise definition has two noteworthy aspects. First, the SME property requires for this information flow to be large when measured with respect to many different partitions of the input and output coordinates, at many different "scales" (i.e., varying the size of the input and output blocks). Second, the measure of information flow between an input and output block is defined with respect to a well-chosen set of restrictions of the original operator. The SME property will be defined in Section 3.2.

The earlier superconcentrator technique works by showing (also using "information flow"-type arguments) that for certain operators F, any circuit to compute F must have a strong connectivity property: it must be a so-called superconcentrator graph. This allows one to apply known lower bounds on the number of edges in bounded-depth superconcentrators (on n input and output vertices). The power of this method is inherently limited, since for $d \geq 3$, the smallest depth-d superconcentrators have $\Theta_d(n \cdot \lambda_d(n))$ edges [DDPW83, Pud94, AP94]. Also, there exist superconcentrators with $\mathcal{O}(n)$ wires [Val76, Val77]; such graphs cannot have constant depth, but may have depth that grows extremely slowly in n [DDPW83]. In contrast with the superconcentrator technique, the SME property has an inherently information-theoretic definition, and the associated lower bounds are proved by a combination of graph-theoretic techniques from earlier work [Pud94, RS03] with novel information-theoretic techniques. For constant-depth circuits, no limitations on the method were known prior to our work, and it seemed plausible that the SME property might imply significantly stronger lower bounds by an improved analysis.[4]

1.3 Two simpler lower-bound methods

We also study two other lower bound methods, both due to Jukna. These methods are simpler than the SME method, and have only been shown to imply lower bounds for depth 2. However, we feel they are still of interest due to their elegance, and due to the fact that the important depth-2 case is still not well-understood.

The first of these methods is the so-called "entropy method" of Jukna [Juk10a]. Like the SME method, this method is a complexity measure of Boolean operators whose definition is information-theoretic: the method identifies information that passes between certain subsets of inputs and outputs, and argues that there must be many wires to carry this information. (In fact, the property of operators used by Jukna's entropy method can be viewed as a relaxation of the SME property, as will be apparent from the definitions.) Using this method, Jukna proved bounds of form $\Omega(n^{3/2})$ for the number of wires required in depth-2 circuits for

[4] For larger depths, some limitations of the SME criterion follow from previous work. In particular, the cyclic convolution operator over $\mathbb{F}_2$, which satisfies the SME property, can be computed in depth polylog(n) using $\mathcal{O}(n \log n \log \log n)$ wires. To see this, we first note that cyclic convolution of length n in $\mathbb{F}_2$ easily reduces to multiplying two polynomials in $\mathbb{F}_2[x]$, each of degree at most $2n-1$. For the latter task, we can use an algorithm of Schönhage [Sch77] (see [Pos11]).

multiplication of two $\sqrt{n}$-by-$\sqrt{n}$ matrices over $\mathbb{F}_2$. Like the SME method, Jukna's entropy method does not yield super-linear lower bounds for computing *linear* operators.

The next lower-bound method we study, also due to Jukna [Juk10b] (building on work of Alon, Karchmer, and Wigderson [AKW90]), does apply to linear operators, and indeed is specific to these operators. Jukna showed that if the columns of a matrix $A \in \mathbb{F}_2^{n \times n}$ have pairwise Hamming distance $\Omega(n)$, then any depth-2 circuit (with arbitrary gates) computing the linear transformation $x \to Ax$ must have $\Omega\left(\frac{n \ln n}{\ln \ln n}\right)$ wires [Juk10b]. This lower-bound criterion applies to a wide range of transformations, including random ones. We will refer to this technique as the "method of pairwise distances."

Jukna's result is actually stronger: the $\Omega\left(\frac{n \ln n}{\ln \ln n}\right)$ lower bound applies to any depth-2 circuit that merely computes Ax correctly when x is a standard basis vector e_i, for $i = 1, \ldots, n$. Such a circuit is said to "represent" the transformation Ax (relative to the standard basis); this is a weaker notion than computing the transformation if we allow non-linear gates. It seems worthwhile to understand how much of the difficulty of computing a linear transformation is "already present" in the simpler task of representing it relative to some basis. In this paper, we will be broadly interested in the complexity of representing linear transformations relative to various bases; we regard the method of pairwise distances as one particular lower-bound technique within this framework.

1.4 Our contributions

1.4.1 Limitations of entropy-based methods

As our most significant (and most technically involved) result, we show that Cherukhin's lower-bound method, formalized by Jukna as the SME property, is inherently limited as a lower-bound criterion for the wire complexity: there is an explicit operator with the SME property that is computable with $\mathcal{O}(n \cdot \lambda_{d-1}(n))$ wires, when $d = 2, 3$, or when $d \geq 6$ is even. For other $d > 1$, this gives an upper bound of $\mathcal{O}(n \cdot \lambda_{d-2}(n))$ wires. Thus, the Cherukhin-Jukna analysis of the SME lower-bound criterion is essentially tight.

The operator we exhibit, called the "Dyadic Interval Replication" (DIR) operator, is fairly natural, and can be roughly described as follows. Let $n := 2^k$. The input is a string $x \in \{0,1\}^n$, viewed as a labeling of the leafs of $\mathcal{T}_k$, the complete binary tree of depth k, along with a specified subtree $\mathcal{T}'$ of $\mathcal{T}_k$. The desired output is the labeling $z \in \{0,1\}^n$ in which the leaf labels of $\mathcal{T}'$ in x have been "copied" to all other subtrees of the same height. This operator is designed to create significant information flow between all parts of the input and output; the subtree $\mathcal{T}'$ will be encoded in the input in a way that is chosen to help ensure the SME property.

Our efficient bounded-depth circuits for the DIR operator are built by an induction on the depth d.[5] The basic idea is that, when the subtree $\mathcal{T}'$ to be copied is small, we can "shrink" the input x, discarding most of the labelings outside of $\mathcal{T}'$. We then either perform the replication task in a direct fashion, or, if the input has been shrunk substantially enough, we inductively apply our circuits for lower depths. By carefully optimizing the set of sizes

[5] Technically, our induction gives circuits to compute a simplified variant, which we then apply to compute the original operator.

to which we attempt to shrink the input, we obtain the upper bounds quoted above. This approach also shows that the DIR operator has *linear*-sized circuits of depth $d = \alpha(n) + 2$, where $\alpha(n) := \min\{d : \lambda_d(n) \le 1\}$ is an extremely slowly-growing function. The idea of attacking a problem at different "scales," and applying induction, has appeared earlier in efficient constructions of bounded-depth superconcentrators [DDPW83] and bounded-depth circuits to compute good error-correcting codes [GHK$^+$11], although the details are different in each case.

We share with earlier authors the belief that, for the cyclic convolution operator, it should be possible to prove significantly better lower bounds for bounded depth—say, bounds of form $\Omega(n^{1+\varepsilon_d})$ for any constant $d > 0$. Our work's message is simply that such lower bounds will have to exploit more of the specific structure of this operator. It seems likely that this will require powerful new ideas. We do hope, however, that our DIR example may be a useful reference point for future work in this area.

Next, we turn to study the limits of Jukna's entropy method. In Section 5, we give a simple example of an operator from $2n$ input bits to n output bits, which is computable by depth-3 circuits with $\mathcal{O}(n)$ wires but requires $\Omega(n^{3/2})$ wires to compute in depth 2. The operator is a simplified variant of matrix multiplication over $\mathbb{F}_2$, in which one of the two matrices is required to contain exactly one 1-entry. The lower bound follows by the same analysis used in [Juk10a] to prove the same lower bound for ordinary matrix multiplication over $\mathbb{F}_2$. Our example shows that the entropy method as formalized in [Juk10a] does not provide a nontrivial lower-bound criterion for depth-3 circuits.

As super-linear lower bounds are already known for the depth-3 wire complexity of certain operators, our negative result on Jukna's entropy method should be interpreted as a note of caution, rather than as a strong barrier to progress in circuit complexity. However, the operator we define to prove our result is also the first known example of a polynomial separation between depth-2 and depth-3 wire complexities—a finding of independent interest. (A *polylogarithmic* complexity separation between depths 2 and 3 is shown in [GHK$^+$11], for the task of computing the encoding function of certain non-explicit linear codes.)

1.4.2 Results on linear transformations

In the rest of the paper, we study the complexity of representing linear transformations over $\mathbb{F}_2^n$. While Lupanov [Lup56] showed that random linear transformations require $\Omega(n^2/\ln n)$ wires to compute by linear circuits, Jukna [Juk10b] showed that, if we allow non-linear gates, $\mathcal{O}(n \ln n)$ wires suffice to *represent* any linear transformation. (He showed this for the standard basis, but his method extends easily to all other bases.) In Section 6, we show that relative to any fixed basis B, most linear transformations require $\Omega(n \ln n)$ wires to represent relative to B. Our result shows that Jukna's upper bound is in general optimal. For our proof, we use a simple trick (similar to a technique in [JS10]) to reduce arbitrary circuits to a special, restricted class; we then apply a standard counting argument.

Recall that Jukna's method of pairwise distances [Juk10b] implies a lower bound of $\Omega\left(\frac{n \ln n}{\ln \ln n}\right)$ on the number of wires needed to represent a large class of linear transformations by depth-2 circuits. Jukna asked whether the "annoying" $(\ln \ln n)^{-1}$ factor in his result could be removed, to match the upper bound he proved for arbitrary matrices. In Section 7, we show that in fact it cannot: there is a matrix family $\{A_n \in \mathbb{F}_2^{n\times n}\}$ whose columns have pairwise distance $\Omega(n)$, for which we can compute the transformation $x \to A_n x$ using a depth-2, $\mathbb{F}_2$-linear circuit with $\mathcal{O}\left(\frac{n \ln n}{\ln\ln n}\right)$ wires. Our construction involves an application of combinatorial designs defined by polynomials over finite fields.

ln Section 8, we show that, for depth-3 circuits, the pairwise-distance method fails completely: there is a matrix family $\{A_n \in \mathbb{F}_2^{n\times n}\}$, whose columns have pairwise distance $\Omega(n)$, and for which we can compute $x \to A_n x$ using a depth-3 linear circuit with $\mathcal{O}(n)$ wires. Recently, Gál et al. [GHK+11] proved a related result: there is a linear error-correcting code $L : \{0,1\}^{\Omega(n)} \to \{0,1\}^n$ with minimum distance $\Omega(n)$, whose encoding function is computable by depth-3 linear circuits with $\mathcal{O}(n \ln\ln n)$ wires. They also show this is optimal for any such code, even if arbitrary gates are allowed. In fact, they determine fairly precisely the minimal wire complexity of computing a good error-correcting code for all depths $d \geq 2$: for depth 2, the answer is $\Theta\left(n\left(\frac{\ln n}{\ln\ln n}\right)^2\right)$, and for depth $d \geq 3$, the answer is $\Theta_d(n \cdot \lambda_d(n))$. As a corollary, this implies that the pairwise-distance method cannot give bounds better than $\Omega(n \ln\ln n)$ for depth 3; our result sharpens this by removing the $(\ln\ln n)$ factor. Comparing our work with [GHK+11] also shows that, while the generator matrices of good linear codes do have columns with high pairwise distance, the property of being a good code is an inherently stronger lower-bound criterion than the pairwise-distance method.

Finally, in Section 9, we show another potential pitfall of circuit-size lower bounds based on hardness of representing linear transformations. We show that for *invertible* linear transformations L, there is always a basis B and a depth-3 circuit C of size $\mathcal{O}(n)$ such that C represents L relative to B. (Non-linear gates are provably necessary in this construction.) Thus in attempts to prove new circuit lower bounds for depths greater than 2, we must at least take care in choosing which basis we use to analyze our linear transformation.

2 Preliminaries

Throughout the paper we use $e_1, \ldots, e_n$ to denote the standard basis vectors in $\mathbb{F}_2^n$. We freely identify $\{0,1\}$ with $\mathbb{F}_2$ when it is convenient. We use $||y||$ to denote the Hamming weight of $y \in \{0,1\}^n$.

Given a gate g in a circuit C, the *depth* of g is defined as the maximal number of edges (i.e., wires) in any directed path from an input gate to g, where each step in the path follows a wire in C in its direction of information-flow. The depth of C is defined as the maximum depth of any of its gates. When we construct circuits, we will refer to the depth-d gates as being at "Level d." Generally these circuits will not be layered; that is, wires may pass from Level d to any Level $d' > d$.

2.1 Wire complexity of operators

A *(total) operator* (or *mapping*) is any function $F : \{0,1\}^n \to \{0,1\}^m$. A case of special interest is when $F = L$ is an $\mathbb{F}_2$-*linear* operator; we will also refer to linear operators as *linear transformations.*

A *partial operator* is a function $F : D \to \{0,1\}^m$, where $D \subseteq \{0,1\}^n$. For $D' \subseteq D$, let $F|_{D'} : D' \to \{0,1\}^m$ be the restriction of F to D.

For a total or partial operator F, define $s(F)$ as the minimum number of wires in any circuit (using arbitrary Boolean functions at gates) which computes F. For $d \geq 0$, define $s_d(F)$ as the minimum number of wires in any circuit which computes F and has depth at most d. For linear operators we also study the quantity $s^{\oplus}(L)$, defined as the minimum number of wires in any $\mathbb{F}_2$-linear circuit that computes L. Similarly, define $s^{\oplus}_d(L)$ as the minimum number of wires in a $\mathbb{F}_2$-linear circuit of depth at most d that computes L.

2.2 Representing linear operators relative to different bases

Fix a basis B for $\mathbb{F}_2^n$. Say that a linear operator $L : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is *represented relative to* B by the circuit C (with n input and m output gates) if $C(x) = L(x)$ for all $x \in B$. (Definitions in [Juk10b, JS10] applied to the standard basis; we consider more general bases.) Note that if C is a *linear* circuit that represents L relative to some basis B, then in fact C *computes* L.

Let $R_d(L; B)$ be defined as the minimum number of wires in any circuit of depth at most d that represents L relative to B. We let $R(L; B) := \min_{d>0} R_d(L; B)$.

2.3 A hashing lemma

The following lemma allows us to "compress" the information in an input string in a wire-efficient way, provided the input is promised to come from a restricted subset. Item 1 of the lemma, which is an especially simple special case, will be used in several sections, while the slightly more technical item 2 will only be used in Section 9.

Lemma 1.

1. *There is a $\mathbb{F}_2$-linear operator $H_{sta} : \mathbb{F}_2^n \to \mathbb{F}_2^{2\lceil\sqrt{n}\rceil}$, computable by a depth-1 circuit with $2n$ wires, and such that for any two distinct standard basis vectors $e_i, e_j \in \mathbb{F}_2^n$, the image vectors $H_{sta}(e_i), H_{sta}(e_j)$ are distinct and each of Hamming weight 2. (We call H_{sta} a "hash mapping" for $\{e_1, \ldots, e_n\}$.)*

2. *Let $D \subset \{0,1\}^n$ be of size n. There is an $\mathbb{F}_2$-linear operator $H : \mathbb{F}_2^n \to \mathbb{F}_2^{\lceil\sqrt{n}\rceil}$, computable by a depth-1 linear circuit with $\mathcal{O}(n)$ wires, that satisfies $H(u) \neq H(v)$ for any two distinct $u, v \in D$.*

Proof. **(1.)** For $n \geq 1$, the number of size-2 subsets of $[2\lceil\sqrt{n}\rceil]$ is $\binom{2\lceil\sqrt{n}\rceil}{2} \geq n$. To each $i \in [n]$, we arbitrarily assign a distinct $S_i \subseteq [2\lceil\sqrt{n}\rceil]$ of size 2. Let the input variable x_i be wired to the two gates $h_t, h_{t'}$ where $S_i = \{t, t'\}$, and let each h_t compute the sum mod 2 of its inputs. Then letting $H_{sta}(x) := (h_1(x), \ldots, h_{2\lceil\sqrt{n}\rceil}(x))$, we have

$$H_{sta}(e_i) \;=\; \mathbf{1}_{S_i} \;,$$

where $\mathbf{1}_{S_i} \in \mathbb{F}_2^{2\lceil\sqrt{n}\rceil}$ is the characteristic function of S_i. Our circuit is depth-1, contains $2n$ wires, and computes a mapping with the desired property.

(2.) We will give a construction that works for sufficiently large n; this is enough to prove the statement. Let $h_1, \ldots, h_{\lceil\sqrt{n}\rceil}$ denote the outputs of H. We define H by building the circuit C_H that computes it. Our construction is probabilistic: each input gate is connected to 14 output gates chosen uniformly and independently at random, and each h_t computes the sum (over $\mathbb{F}_2$) of its inputs. (If multiple wires connect the input x_i and output h_t, each wire contributes to the sum. The constant 14 is simply chosen large enough to make the analysis work.) The total number of wires is $14n = \mathcal{O}(n)$, as required.

We claim that, with probability $1 - o(1)$ over the randomness in our construction, H is injective on D. To see this, first fix attention to any pair $u, v \in D$ with $u \neq v$. For clarity, reorder the input coordinates so that $u_n \neq v_n$. Condition on any wiring of the outgoing wires from input gates $x_1, \ldots, x_{n-1}$, and consider x_n to have not yet received its assignment of outgoing wires. This incomplete circuit defines a linear transformation $\widetilde{H} : \{0,1\}^n \to \{0,1\}^{\lceil\sqrt{n}\rceil}$ from the inputs to the outputs.

Let $\bar{\mathbf{t}} = (\mathbf{t}(1), \ldots, \mathbf{t}(14)) \in [\lceil\sqrt{n}\rceil]^{14}$ be the 14 uniformly chosen indices of gates to which x_n is to be connected. Assume without loss of generality that $u_n = 0, v_n = 1$. Then $H(u) = \widetilde{H}(u)$, while

$$H(v) \;=\; \widetilde{H}(v) \oplus e_{\mathbf{t}(1)} \oplus \ldots \oplus e_{\mathbf{t}(14)}$$

(here e_t denotes the t-th standard basis vector in $\mathbb{F}_2^{\lceil\sqrt{n}\rceil}$). Let $w := \widetilde{H}(u) \oplus \widetilde{H}(v)$; it follows that

$$H(u) \;=\; H(v) \quad\iff\quad e_{\mathbf{t}(1)} \oplus \ldots \oplus e_{\mathbf{t}(14)} = w\ . \tag{1}$$

We will show that, regardless of the value w, the probability $p_w := \Pr[e_{\mathbf{t}(1)} \oplus \ldots \oplus e_{\mathbf{t}(14)} = w]$ satisfies $p_w = o(n^{-2})$. First, $p_w = 0$ if $||w|| > 14$. There are $\binom{\lceil\sqrt{n}\rceil}{k}$ strings of Hamming weight k, and each occurs as $e_{\mathbf{t}(1)} \oplus \ldots \oplus e_{\mathbf{t}}(14)$ with equal probability, so $p_w = \mathcal{O}(n^{-3})$ if $||w|| \in [6, 14]$.

Now say $||w|| = k \leq 5$. Given an outcome of $\bar{\mathbf{t}}$ satisfying $e_{\mathbf{t}(1)} \oplus \ldots \oplus e_{\mathbf{t}(14)} = w$, the cancellations which occur imply that we can find $\ell := (14 - k)/2$ pairs of indices $\{i_1, j_1, \ldots, i_\ell, j_\ell\} \subseteq [14]$, with no two indices appearing twice, such that

$$\mathbf{t}(i_r) \;=\; \mathbf{t}(j_r),\ r = 1, 2, \ldots, \ell\ . \tag{2}$$

Each event $[\mathbf{t}(i) = \mathbf{t}(j)]$ occurs with probability $\lceil\sqrt{n}\rceil^{-1}$ if $i \neq j$. The variables $\mathbf{t}(1), \ldots, \mathbf{t}(14)$ are independent, so the probability that Eq. (2) holds is $\mathcal{O}(n^{-\ell/2}) = \mathcal{O}(n^{-9/4})$, using $k \leq 5$.

In each case we find $p_w = o(n^{-2})$, so by Eq. (1), we conclude $\Pr[H(u) = H(v)] = o(n^{-2})$. By a union bound over all pairs $u, v \in D$, the probability that H fails to be injective is $\binom{n}{2} \cdot o(n^2) = o(1)$. So our construction of H has the desired property on some setting to the randomness. □

3 Entropy and circuit lower bounds

3.1 Entropy of operators

Given an operator $F = (f_1, \ldots, f_m) : \{0,1\}^n \to \{0,1\}^m$, define the *entropy*

$$\mathrm{Ent}(F) \;:=\; \log_2\left(|\mathrm{range}(F)|\right)$$

as the logarithm of the number of distinct outputs of F. We have two easy facts, both from [Juk10a]:

Fact 2. *Suppose we fix some assignment to a subset $I \subseteq [n]$ of the inputs to F, and let $F' : \{0,1\}^{n-|I|} \to \{0,1\}^m$ be the resulting operator. Then $\mathrm{Ent}(F') \leq \mathrm{Ent}(F)$.*

Fact 3. *Suppose that there is a subset $S \subseteq [n]$, such that from the value $F(x)$ one can always infer the values of all input bits x_i with $i \in S$. Then, $\mathrm{Ent}(F) \geq |S|$.*

Say we are given an $x \in \{0,1\}^n$, a nonempty set $I \subseteq [n]$, and an $i \in I$. Let $x[I;i]$ denote the vector obtained from x by setting the i^{th} bit to 1, setting the $(i')^{th}$ bit to 0 for each $i' \in I \setminus \{i\}$, and leaving all other bits unchanged.

Letting $F(x)$ be as above, and fixing some output coordinate $j \in [m]$, define the function

$$f_{I,i,j}(x) \ := \ f_j(x[I;i]) \ .$$

Now for $J \subseteq [m]$, define a mapping $F_{I,J} : \{0,1\}^{n-|I|} \to \{0,1\}^{|I|\cdot|J|}$ by

$$F_{I,J} \ := \ (f_{I,i,j})_{i \in I, j \in J} \ .$$

Note, $F_{I,J}$ has as its domain the bits $\{x_\ell : \ell \notin I\}$. (We will still write $F_{I,J} = F_{I,J}(x)$, however.) We can now state Jukna's entropy-based lower-bound criterion:

Theorem 4. *[Juk10a] Let $F : \{0,1\}^n \to \{0,1\}^m$. Let $I_1, \ldots, I_p$ be a partition of $[n]$, and let $J_1, \ldots, J_p$ be a partition of $[m]$ with the same number of parts. Then,*

$$s_2(F) \ \geq \ \sum_{t=1}^{p} \mathrm{Ent}(F_{I_t,J_t}) \ .$$

3.2 Strong Multiscale Entropy

Next we define the Strong Multiscale Entropy property, which is a generalization due to Jukna [Juk12, Chap. 13] of a lower-bound method of Cherukhin [Che08a].

For a pair of integers $N, m \geq n_0$, we consider pairs $(\mathcal{I}, \mathcal{J})$ where $\mathcal{I}$ is a collection of subsets of $[N]$ and $\mathcal{J}$ is a collection of subsets of $[m]$. For an integer $p \leq n_0$, we say that $(\mathcal{I}, \mathcal{J})$ form an *n_0-partition at scale p* if:

1. $\mathcal{I}$ consists of p disjoint sets $I_t \subseteq [N]$, with $|I_t| = \lfloor n_0/p \rfloor$;

2. $\mathcal{J}$ consists of $\lfloor n_0/p \rfloor$ disjoint sets $J_{t'} \subseteq [m]$, with $|J_{t'}| = p$.

Say that a family $\{F_N : \{0,1\}^N \to \{0,1\}^m\}_{N>0}$ has the *Strong Multiscale Entropy (SME) property*, if there exists a parameter $n_0 = n_0(N) = \Omega(N)$ along with constants $C, \gamma > 0$ such that, for every N and every $p \in [C\sqrt{n_0}, n_0]$, there exists a pair $(\mathcal{I}, \mathcal{J})$ that form an n_0-partition at scale p, satisfying

$$\mathrm{Ent}(F_{I_t,J_{t'}}) \geq \gamma \cdot n_0 \ , \quad \forall I_t \in \mathcal{I}, \ J_{t'} \in \mathcal{J}. \tag{3}$$

We also define the *enhanced SME property* similarly to the above, except that we ask for a pair $(\mathcal{I}, \mathcal{J})$ satisfying Eq. (3) for all $p \in [C, n_0]$.

To state the lower bounds for operators with the SME property, we need some definitions. We let $g^{(i)}$ denote the i-fold composition of a function $g : \mathbb{Z} \to \mathbb{Z}$. Suppose g satisfies $1 \leq g(n) < n$ for all $n > 1$; we then define $g^* : \{1, 2, 3, \ldots\} \to \{0, 1, 2, \ldots\}$ by

$$g^*(n) \ := \ \min\{i : g^{(i)}(n) \leq 1\} \ .$$

Following conventions in [RS03, Che08a], define a family of slowly-growing functions $\lambda_d(n)$ as follows: let

$$\lambda_1(n) \ := \ \lfloor \sqrt{n} \rfloor, \quad \lambda_2(n) := \lceil \log_2 n \rceil \ ,$$

and for $d > 2$, let

$$\lambda_d(n) \ := \ \lambda^*_{d-2}(n) \ .$$

(Note that $\lambda_3(n) = \Theta(\ln \ln n)$.)

Applying the technique of Cherukhin [Che08a], Jukna proved:

Theorem 5. *[Juk12, Chap. 13] Suppose the operator family $\{F_N : \{0,1\}^N \to \{0,1\}^m\}$ has the Strong Multiscale Entropy property. Then for any constant $d \geq 2$, any depth-d circuit to compute F_N has $\Omega_d(N \cdot \lambda_{d-1}(N))$ wires.*

4 Limitations of the SME lower-bound criterion

In this section we introduce an explicit Boolean operator called the "Dyadic Interval Replication" (DIR) operator, and use it to show that the Strong Multiscale Entropy property does not imply wire complexity lower bounds substantially better than those given by Theorem 5. We prove:

Theorem 6. *There is an operator family $\{\mathrm{DIR}_N : \{0,1\}^N \to \{0,1\}^{\Omega(N)}\}$, with the enhanced Strong Multiscale Entropy property, for which we have:*

$$s_2\left(\mathrm{DIR}_N\right) \ = \ \Theta(N^{3/2}) \ = \ \Theta\left(N \cdot \lambda_1(n)\right) \ ;$$

$$s_3\left(\mathrm{DIR}_N\right) \ = \ \Theta\left(N \ln N\right) = \Theta\left(N \cdot \lambda_2(n)\right) \ ;$$

$$s_5\left(\mathrm{DIR}_N\right) \ = \ \mathcal{O}\left(N \ln \ln N\right) = \mathcal{O}(N \cdot \lambda_3(n)) \ ;$$

For even $d = d(N) \geq 6$,

$$s_d\left(\mathrm{DIR}_N\right) \ = \ \mathcal{O}\left(N \cdot \lambda_{d-2}(N)\right) = \mathcal{O}\left(N \cdot \lambda_{d-1}(N)\right) \ ,$$

and so for constant, even values $d \geq 6$,

$$s_d\left(\mathrm{DIR}_N\right) \ = \ \Theta_d\left(N \cdot \lambda_{d-1}(N)\right) \ .$$

For odd values $d = d(N) \geq 7$, we have

$$s_d\left(\mathrm{DIR}_N\right) \ \leq \ s_{d-1}\left(\mathrm{DIR}_N\right) = \mathcal{O}(N \cdot \lambda_{d-2}(N)) \ .$$

The lower bounds come from Theorem 5. In the statements above, we are using the fact that $\lambda_d(N) = \Theta\left(\lambda_{d+1}(N)\right)$ for even values $d = d(N) \geq 4$.

The hidden constants in the $\mathcal{O}(\cdot)$ notation above are *independent* of d. Thus, DIR_N is computable by a circuit with $\mathcal{O}(N)$ wires, of depth $\alpha(N) + 2$, where $\alpha(N) := \min\{d : \lambda_d(N) \leq 1\}$ is an extremely slowly-growing function. On the other hand, the lower bounds from Theorem 5 hide a multiplicative constant that goes to 0 as $d \to \infty$. So there may be room for some further tightening of the upper or lower bounds for all values of d.

In Theorem 6, we show that DIR_N satisfies not only the SME property, but also the *enhanced* SME property. We do so to clarify that even this stronger property does not yield significantly better lower bounds than those given by Theorem 5. We emphasize that our upper bounds for the specific operator DIR_N are also upper limits on the lower bounds that follow in general from the SME property.

4.1 The DIR operator

Now we define DIR_N and show it has the SME property. In our work in this section, it will be convenient to index vectors in $\{0,1\}^n$ as $x = (x_0, \ldots, x_{n-1})$, and regard the indices as lying in $\mathbb{Z}_n$. For $a \in \mathbb{Z}_n$, define

$$\mathrm{shift}(x; a) \;:=\; \left(x_{-a}, x_{1-a}, \ldots, x_{(n-1)-a}\right) ,$$

with index arithmetic over $\mathbb{Z}_n$. We also use set addition: for $A, B \subseteq \mathbb{Z}_n$, define $A + B := \{a + b : a \in A, b \in B\}$ (with addition over $\mathbb{Z}_n$). For $i \in \mathbb{Z}_n$, we write $A + i := A + \{i\}$.

We consider input lengths $N = 2 \cdot 2^k + \lceil \log_2 k \rceil$, for $k \geq 1$. We let $n := 2^k$, and we regard inputs of length N to have the form

$$(x, y, r) \in \{0,1\}^{n+n+\lceil \log_2 k \rceil} .$$

We will consider r as an integer in $[0, k-1]$.[6] Define the *Dyadic Interval Replication* operator $\mathrm{DIR}_N(x, y, r) : \{0,1\}^N \to \{0,1\}^n$ by the following rule:

1. If $||y|| \neq 1$, output $z := 0^n$.

2. Otherwise, let $i = i(y) \in \mathbb{Z}_n$ be the unique index for which $y_i = 1$. Output the string z given by
$$z_j := \mathrm{shift}(x; i \cdot 2^r)_{(j \bmod 2^r)} . \tag{4}$$

Let us explain this definition in words. The input vector x divides naturally into $n/2^r = 2^{k-r}$ substrings of length 2^r. The operator DIR_N chooses one of these substrings, and outputs 2^{k-r} consecutive copies of this substring.

We can extend the definition to input lengths $N \geq 6$ not of the above form, by considering the input to be padded with irrelevant bits.

[6] If k is not a power of 2, some values in $[0, k-1]$ will have more than one encoding; this technicality doesn't affect our arguments.

4.2 Establishing the SME property for DIR

Lemma 7. *The family* $\{\mathrm{DIR}_N\}$ *has the enhanced SME property.*

Proof of Lemma 7. The number of irrelevant bits in the input to DIR_N is not more than twice the number of relevant bits, so for the purposes of our asymptotic analysis, we may assume that N is of form $N = 2 \cdot 2^k + \lceil \log_2 k \rceil$ with $k \geq 1$. Let $n := 2^k$, and let $n_0 := n = \Omega(N)$.

Let $p \in [4, n]$ be given. Define collections $\mathcal{I}, \mathcal{J}$ as follows. For $t \in [p]$, let

$$I_t \ := \ \{0, 1, \ldots, \lfloor n/p \rfloor\} + (t-1)\lfloor n/p \rfloor$$

be the t^{th} consecutive interval of length $\lfloor n/p \rfloor$ in $\mathbb{Z}_n$. For $t' \in [\lfloor n/p \rfloor]$, let

$$J_{t'} \ := \ \{0, 1, \ldots, p\} + (t'-1)p$$

be the $(t')^{th}$ interval of length p in $\mathbb{Z}_n$. Note that $(\mathcal{I}, \mathcal{J})$ form an n_0-partition at scale p for the input and output lengths of DIR_N.

Say we are given any $t \in [p]$ and $t' \in [\lfloor n/p \rfloor]$; we will show that $\mathrm{Ent}(\mathrm{DIR}_{I_t, J_{t'}}) = \Omega(n) = \Omega(N)$. First, suppose that $p \in [2^\ell, 2^{\ell+1})$, where $\ell > 0$. Then, $J_{t'}$ contains an interval $\widehat{J}$ of form

$$\widehat{J} \ = \ \{0, \ldots, 2^{\ell-1} - 1\} + s \cdot 2^{\ell-1} \ ,$$

for some $s \in [0, 2^{k-\ell+1})$. We now fix assignments (y^*, r^*) to part of the input to $\mathrm{DIR}_{I_t, J_{t'}}$:

$$y^* \ := \ 0^n, \quad r^* \ := \ \ell - 1 \ .$$

Define $\mathrm{DIR}^*_{I_t, J_{t'}}(x) := \mathrm{DIR}_{I_t, J_{t'}}(x, y^*, r^*)$. Using Fact 2 applied to $\mathrm{DIR}_{I_t, J_{t'}}$, we have $\mathrm{Ent}(\mathrm{DIR}_{I_t, J_{t'}}) \geq \mathrm{Ent}(\mathrm{DIR}^*_{I_t, J_{t'}})$. So it will be enough to lower-bound $\mathrm{Ent}(\mathrm{DIR}^*_{I_t, J_{t'}})$.

Fix any $i \in I_t$. Our assignment $y^* := 0^n$ satisfies

$$||y^*[I_t; i]|| \ = \ 1 \ .$$

Thus for any x, case 2 holds in the definition of $\mathrm{DIR}(x, y^*[I_t; i], r^*)$. Consider any $j \in \widehat{J}$; substituting values into Eq. (4), we find

$$\begin{aligned} (\mathrm{DIR}_N(x, y^*[I_t; i], r^*))_j \ &= \ \left(\mathrm{shift}(x; i \cdot 2^{\ell-1})\right)_{(j \bmod 2^{\ell-1})} \\ &= \ x_{(j \bmod 2^{\ell-1}) - i 2^{\ell-1}} \ . \end{aligned}$$

Thus, from the output of $\mathrm{DIR}^*_{I_t, J_t}(x)$ we can determine x_a, for each $a \in \widehat{J}_{(\bmod 2^{\ell-1})} - 2^{\ell-1} \cdot I_t$. Here, $\widehat{J}_{(\bmod 2^{\ell-1})} := \{j' \in [0, 2^{\ell-1} - 1] : j' = j \mod 2^{\ell-1}$ for some $j \in \widehat{J}\}$. We observe that actually $\widehat{J}_{(\bmod 2^{\ell-1})} = [0, 2^{\ell-1} - 1]$, since $\widehat{J}$ is a consecutive interval of length $2^{\ell-1}$. Fact 3 now implies that

$$\mathrm{Ent}(\mathrm{DIR}^*_{I_t, J_{t'}}) \ \geq \ \left| [0, 2^{\ell-1} - 1] - 2^{\ell-1} \cdot I_t \right| \ .$$

Recall that I_t is an interval of length $\lfloor n/p \rfloor$. It follows that, with arithmetic taken over the *integers* $\mathbb{Z}$, the set $[0, 2^{\ell-1} - 1] - 2^{\ell-1} \cdot I_t$ is an interval in $\mathbb{Z}$ of size $2^{\ell-1}\lfloor n/p \rfloor$. We conclude that, over $\mathbb{Z}_n$,

$$\begin{aligned} \left| [0, 2^{\ell-1} - 1] - 2^{\ell-1} \cdot I_t \right| \ &= \ \min\{n, 2^{\ell-1}\lfloor n/p \rfloor\} \\ &\geq \ \min\{n, (p/4) \cdot \lfloor n/p \rfloor\} \ = \ \Omega(n) \ . \end{aligned}$$

This proves Lemma 7. □

4.3 Efficient bounded-depth circuits for DIR

In this subsection, we prove the upper bounds needed to establish Theorem 6.

First we prove the upper bound for depth 2, namely $s_2\left(\mathrm{DIR}_N\right) = \mathcal{O}\left(N^{3/2}\right)$. Our circuit construction will split into two cases, handled separately as follows: first, if $2^r < \sqrt{n}$, the needed substring of x can be copied into $\sqrt{n}$ gates on Level 1 of the circuit, and then copied from this middle level by the output gates. On the other hand, if $2^r \geq \sqrt{n}$, then each output bit can depend on at most $\sqrt{n}$ possible bits of x.

Lemma 8. $s_2(\mathrm{DIR}_N) = \mathcal{O}\left(N^{3/2}\right) = \mathcal{O}(N \cdot \lambda_1(N))$.

Proof. As before, we may assume $N = 2 \cdot 2^k + \lceil \log_2 k \rceil$, with $n := 2^k$. For convenience, we will assume further that k is even, so that $\sqrt{n} = 2^{k/2}$ is an integer.

Recall that, when $||y|| = 1$, the output of $\mathrm{DIR}_N(x, y, r)$ will consist of 2^{k-r} consecutive copies of a substring of x of length 2^r. We will design two depth-2 circuits $C^{\downarrow}, C^{\uparrow}$, each with $\mathcal{O}\left(N^{3/2}\right)$ wires. $C^{\downarrow}$ will compute DIR_N under the promise that $2^r < \sqrt{n}$; $C^{\uparrow}$ will compute DIR_N provided $2^r \geq \sqrt{n}$. It is then easy to combine these two circuits to get a single circuit computing DIR_N under no assumption. (We apply each of $C^{\downarrow}, C^{\uparrow}$ to the input, merging their corresponding output gates. Each output gate is also wired to the inputs of r, to determine whether it should output the value of $C^{\downarrow}$ or of $C^{\uparrow}$; this takes $\mathcal{O}(n \cdot \log_2 k)$ additional wires.)

For $C^{\downarrow}$, the basic idea is that when $2^r < \sqrt{n}$, fewer than $\sqrt{n}$ bits of x actually "matter" for the output; we can extract these bits on Level 1 and distribute them to the appropriate outputs on Level 2. More precisely, we will have $\sqrt{n} + 1$ gates $(s, g_1, \ldots, g_{\sqrt{n}})$ on Level 1 of our circuit $C^{\downarrow}$, each wired to all of (x, y, r). We set $s = 1$ iff $||y|| = 1$. The gates $g_1, \ldots, g_{\sqrt{n}}$ will simply copy the interval of size $2^r < \sqrt{n}$ in x that must be replicated in the output of DIR_N, as determined by x, r, and $i = i(y)$. (This interval of bits from x will be padded with $\sqrt{n} - 2^r$ zeros when copied to Level 1.)

Next, each output bit z_t $(t \in \mathbb{Z}_n)$ is wired to all Level 1 gates and to r. We won't give an explicit rule, but it is clear that with these inputs, each z_t can determine its correct output to compute DIR_N (assuming here that $2^r < \sqrt{n}$). The number of wires in $C^{\downarrow}$ is $\mathcal{O}\left(n^{3/2} + n(\sqrt{n} + \log_2 k)\right) = \mathcal{O}\left(N^{3/2}\right)$, as required.

Now we build $C^{\uparrow}$. The basic idea here is that, assuming $2^r \geq \sqrt{n} = 2^{k/2}$, each output bit z_t depends only on y, r, and on input bits $x_{t'}$ for which $t - t'$ is a multiple of $\sqrt{n}$. Thus, after "compactifying" the relevant information in y into $\sqrt{n}$ bits on Level 1, each output bit can be computed from the Level 1 gates, from r, and from $\sqrt{n}$ bits of x, using $\mathcal{O}\left(n^{3/2}\right)$ wires in total. Details follow.

Let $H(y) = H_{sta}(y) = (h_1, \ldots, h_{2\lceil\sqrt{n}\rceil}) : \mathbb{F}_2^n \to \mathbb{F}_2^{2\lceil\sqrt{n}\rceil}$ be the operator from item 1 of Lemma 1 that is injective on $\{e_1, \ldots, e_n\}$. We implement H on Level 1 of our circuit with $\mathcal{O}(n)$ wires, following the construction in Lemma 1. As in $C^{\downarrow}$, on Level 1 we also include a single gate s, wired to r, that outputs 1 iff $||y|| = 1$. Thus the total number of wires between inputs and Level 1 is $\mathcal{O}(n)$, and there are $\sqrt{n} + 1$ gates at Level 1.

Next, each output bit z_t $(t \in \mathbb{Z}_n)$ is wired to all Level 1 gates, to all of r, and to the input bits $(x_t, x_{t+\sqrt{n}}, x_{t+2\sqrt{n}}, \ldots, x_{t+(\sqrt{n}-1)\sqrt{n}})$. Thus our circuit is of depth 2, and the total number of wires to the outputs is $n \cdot ((\sqrt{n} + 1) + \lceil \log_2 k \rceil + \sqrt{n}) = \mathcal{O}(n^{3/2})$.

Rather than specifying the output rule for z_t precisely, we argue that this gate has all the information it needs to output $(\mathrm{DIR}_N(x, y, r))_t$ correctly (assuming $2^r \geq \sqrt{n}$). First, if

$||y|| \neq 1$, then z_t can learn this and output 0 by looking at s. Otherwise, z_t knows that $||y|| = 1$. In this case, z_t must output the bit $\mathrm{shift}(x; i \cdot 2^r)_{(t \bmod 2^r)} = x_{(t \bmod 2^r) - i2^r}$ (here the outer index arithmetic is over $\mathbb{Z}_n$). This desired bit lies among $(x_t, x_{t+2^r}, \ldots, x_{t+(2^{k-r}-1)2^r})$, and these are contained in the inputs to z_t since 2^r is a multiple of $\sqrt{n}$. Finally, the value $i = i(y)$ can be determined from $H(y)$, because $H(y)$ determines y when $||y|| = 1$. Thus z_t can output the correct value. □

Next, we will develop tools for building more-efficient circuits of higher depths. For depth 3, we will show $s_3(\mathrm{DIR}_N) = \mathcal{O}(N \ln N)$. The plan for depth 3 is fairly simple: First, from an input (x, y, r) satisfying $||y|| = 1$, we can extract the index $i = i(y)$ and the value $p := (i \cdot 2^r \bmod n)$ in depth 1, with $n \log_2 n$ wires. Then we show that there is a circuit to compute the appropriate output given (x, i, r, p) using $\mathcal{O}(N)$ wires in depth 2, *under the promise* that r equals some fixed value $a \in [0, k-1]$. As there are only $\log_2 n$ possible values of r, we can combine these circuits (merging their output gates) into a single circuit of total depth 3 and with $\mathcal{O}(N \ln N)$ wires overall.

To build our circuits for depths 3 and higher, it is useful to introduce some auxiliary operators, which are "easier" versions of DIR_N. The first such operator further restricts the "admissible" values of r to some interval $[a, b] \subseteq [0, k-1]$. Define $\mathrm{DIR}_N^{[a,b]} : \{0,1\}^{2n + \lceil \log_2 k \rceil}$ by

$$\mathrm{DIR}_N^{[a,b]}(x, y, r) \ := \ \begin{cases} \mathrm{DIR}_N(x, y, r) & \text{if } r \in [a, b], \\ 0^n & \text{otherwise.} \end{cases}$$

The second simplified operator makes the values i and $p := (i \cdot 2^r \bmod n)$ available in binary. Define $\mathrm{DIR}_N^{bin,[a,b]} : \{0,1\}^{n + k + \lceil \log_2 k \rceil + k}$ by

$$\mathrm{DIR}_N^{bin,[a,b]}(x, i, r, p) \ := \ \begin{cases} \mathrm{DIR}_N^{[a,b]}(x, e_i, r) & \text{if } p = i \cdot 2^r \bmod n, \\ 0^n & \text{otherwise.} \end{cases}$$

We are abusing notation slightly, since the input size to $\mathrm{DIR}_N^{bin,[a,b]}$ is actually smaller than $N = 2n + \lceil \log_2 k \rceil$.

The following lemma, which handles a fixed value $r = a$, will be useful.

Lemma 9. *For any $a \in [0, k-1]$, there is a depth-2 circuit C^a, using $\mathcal{O}(n)$ wires, that computes $\mathrm{DIR}_N^{bin,[a,a]}$.*

Proof. Let a be fixed. We include a single gate s on Level 1 that outputs 1 iff all of the following hold:

1. $||y|| = 1$;

2. $p = i \cdot 2^r \bmod n$;

3. $r = a$.

Also on Level 1 of the circuit C^a, we define gates x'_t, for $t \in \{0, 1, \ldots, 2^a - 1\}$. Each such gate is wired to the $(k-a)$ most significant bits of p, and to the inputs $(x_t, x_{t+2^a}, \ldots, x_{t+(2^{k-a}-1)2^a})$.

Let $\tilde{p} := p - (p \bmod 2^a)$ be the value obtained by assuming that the unseen bits of p are zero. We then set $x'_t := x_{t-\tilde{p}}$. Note that the needed bit of x falls within the inputs to x'_t. The number of incoming wires to this group of gates is $2^a \cdot \big(2^{k-a} + (k-a)\big) = \mathcal{O}(2^k) = \mathcal{O}(n)$.

Finally, given an output gate z_j of C^a with $j \in \mathbb{Z}_n$, we set

$$z_j \;:=\; x'_{(j \bmod 2^a)} \wedge s \;,$$

so that the output gates have $2n$ incoming wires in total, and the entire circuit C^a is depth-2 and contains $\mathcal{O}(n)$ wires.

We claim that C^a has the desired behavior. To see this, fix any $j \in \mathbb{Z}_n$. First, if $s = 0$ then $z_j = 0$ as needed. Next assume that $s = 1$, so that $\mathrm{DIR}_N^{bin,[a,a]}(x, i, r, p) = \mathrm{DIR}_N(x, e_i, a)$. We compute

$$\begin{aligned}
z_j &= x'_{(j \bmod 2^a)} \wedge 1 \\
&= x_{(j \bmod 2^a) - \tilde{p}} \\
&= x_{(j \bmod 2^a) - i2^a} \\
&\text{(since } s = 1 \text{ implies } \tilde{p} = p = i \cdot 2^a \bmod n) \\
&= (\mathrm{shift}(x; i \cdot 2^a))_{(j \bmod 2^a)} \;,
\end{aligned}$$

as needed. This proves the correctness of C^a. □

Lemma 10. *For any* $0 \leq b < k$, $s_2\left(\mathrm{DIR}_N^{bin,[0,b]}\right) = \mathcal{O}\left(N \ln N\right)$. *Also,* $s_3\left(\mathrm{DIR}_N\right) = \mathcal{O}\left(N \ln N\right) = \mathcal{O}(N \cdot \lambda_2(N))$.

Proof. Again assume that $N = 2 \cdot 2^k + \lceil \log_2 k \rceil$, with $n := 2^k$.

First we show $s_2\left(\mathrm{DIR}_N^{bin,[0,k-1]}\right) = \mathcal{O}\left(N \ln N\right)$. Let (x, i, r, p) be the inputs. We apply the circuits $C^0, C^1, \ldots, C^b$ from Lemma 9 to (x, i, r, p). Each such circuit C^a has n outputs, call them $z_{0,a}, \ldots, z_{n-1,a}$. For $t \in \mathbb{Z}_n$, we "collapse" $z_{t,0}, \ldots, z_{t,b}$ into the single output gate z_t (which takes all the inputs of $z_{t,0}, \ldots, z_{t,b}$ as its inputs). This gate is also wired to the input r, and it outputs $z_t := z_{t,r}$.

Let C denote the circuit we have constructed. That C computes $\mathrm{DIR}_N^{bin,[0,b]}$ is immediate. C is of depth 2 since each C^a is of depth 2, and C has $\mathcal{O}(N) \cdot (b+1) + n \cdot \lceil \log_2 k \rceil = \mathcal{O}(N \ln N)$ wires, since each C^a has $\mathcal{O}(N)$ wires and $b < k = \log_2 n$.

Next we show $s_3\left(\mathrm{DIR}_N\right) = \mathcal{O}\left(N \ln N\right)$. In our circuit C' for DIR_N, we will assume that the input satisfies $||y|| = 1$. As usual, it is easy to modify this circuit to handle the case where $||y|| \neq 1$.

On Level 1 of our circuit, we compute $i = i(y)$ and $p := i \cdot 2^r \bmod n$. This takes $\mathcal{O}(n \ln n)$ wires since i, p are k bits each. Next, we set $b := k - 1$ and apply our previously constructed circuit C for $\mathrm{DIR}_N^{bin,[0,k-1]}$ to (x, i, r, p). By definition, the resulting output is $\mathrm{DIR}_N(x, y, r)$. Our construction of C' is of depth $1 + 2 = 3$ and contains $\mathcal{O}(N \ln N)$ wires. □

To work with depths larger than 3, we will give a technique that allows us to "shrink" the size of an instance of the Dyadic Interval Replication problem, discarding most of the irrelevant bits of x, when the value r is not too large. The next lemma collects two variants of this process.

Lemma 11. *Let $N = 2 \cdot 2^k + \lceil \log_2 k \rceil$. Let $0 \leq a \leq b \leq k-1$ be given, and let $d = d(N) \geq 1$. Let $N' := 2 \cdot 2^{b-a+1} + \lceil \log_2 (b-a+1) \rceil$.*

1. *There is a depth-$(d+2)$ circuit C that computes $\mathrm{DIR}_N^{bin,[a,b]}$; the number of wires in C is*
$$2^{a+1} \cdot s_d \left(\mathrm{DIR}_{N'}^{bin,[0,b-a]} \right) + \mathcal{O}(N) \ .$$
2. *There is a depth-$(d+3)$ circuit C' that computes $\mathrm{DIR}_N^{[a,b]}$, and has*
$$2^{a+1} \cdot s_d \left(\mathrm{DIR}_{N'}^{bin,[0,b-a]} \right) + \mathcal{O}(N(k-b))$$
wires.

In each case the $\mathcal{O}(\cdot)$ is independent of a, b, d.

Proof. **(1.)** We split into two cases according to whether the input p satisfies $p = 0 \bmod 2^{b+1}$, designing a different depth-$(d+2)$ circuit for each case. It is easy to combine the two circuits using $\mathcal{O}(N)$ additional wires. We assume in the following construction that $p \neq 0 \bmod 2^{b+1}$, and then sketch the other, quite similar case; this will double the number of wires, giving the quoted bound.

On Level 1 of our circuit C (for the case $p \neq 0 \bmod 2^{b+1}$), we include gates $x' = (x'_0, \ldots, x'_{2^{b+1}-1})$, where x'_t is wired to $(x_t, x_{t+2^{b+1}}, \ldots, x_{t+(2^{k-b-1}-1)2^{b+1}})$, and also to the $k-b-1$ most significant bits of p, that is, to $p_{b+1}, \ldots, p_{k-1}$. We set
$$x'_t \ := \ x_{t-\tilde{p}-2^{b+1}} \ , \qquad \text{where} \quad \tilde{p} \ := \ \textstyle\sum_{\ell=b+1}^{k-1} p_\ell 2^\ell \ = \ p - (p \bmod 2^{b+1}) \ .$$

$x_{t-\tilde{p}-2^{b+1}}$ lies among the inputs to x'_t as needed. Computing x' uses $2^{b+1} \cdot (2^{k-b-1} + (k-b-1)) = \mathcal{O}(N)$ wires. Also on Level 1 of C, we include a gate s, wired to (i, r, p). We set $s := 1$ iff the following conditions hold: (1) $p = i \cdot 2^r \bmod n$; (2) $r \in [a, b]$. Computing s requires $o(N)$ wires. Define the quantities $i' := i \bmod 2^{b-a+1}$, $r' := \min\{r-a, b-a\}$, $p' := i' \cdot r' \bmod 2^{b-a+1}$, and note that (i', r', p') can all be determined from (i, r, p). On Level 1 of C we also include gates computing (i', r', p'); this takes $\mathcal{O}(\ln^2 N) = o(N)$ wires. For $u \in [0, 2^a - 1]$, define $x'(u) = (x'(u)_0, \ldots, x'(u)_{2^{b-a+1}-1})$ by letting
$$x'(u)_\ell \ := \ x'_{\ell \cdot 2^a + u} \ .$$

Here we are just introducing new notation that "divides up" x' into the subsequences $x'(0), \ldots, x'(2^a - 1)$.

Next, on Levels 2 through $(d+1)$ of C, for each $u \in [0, 2^a - 1]$ we place a copy of an optimal (wire-minimizing) depth-d circuit computing $\mathrm{DIR}_{N'}^{bin,[0,b-a]}$, to which we provide the values $(x'(u), i', r', p')$ as inputs. Let $z'(u) = (z'(u)_0, z'(u)_1, \ldots, z'(u)_{2^{b-a+1}-1})$ denote the output gates of this circuit.

Finally, for $t \in \mathbb{Z}_n$, we may uniquely write $t = \ell \cdot 2^a + u$, for some $\ell \in [0, 2^{k-a} - 1]$ and $u \in [0, 2^a - 1]$. Then the output gate z_t is defined by
$$z_t \ := \ z'(u)_{\ell \bmod 2^{b-a+1}} \wedge s \ .$$

The total number of wires in our circuit C is $\mathcal{O}(N) + 2^a \cdot s_d\left(\mathrm{DIR}_{N'}^{bin,[0,b-a]}\right)$, and the depth of C is $(d+2)$. Next we prove correctness. First, if $s = 0$ then C outputs 0^n as needed, so assume $s = 1$ (which implies $r' = r - a$). Fix $t \in \mathbb{Z}_{n=2^k}$, and write $t = \ell \cdot 2^a + u$ with ℓ, u as above. We have

$$\begin{aligned}
z_t &= z'(u)_{\ell \bmod 2^{b-a+1}} \wedge s \\
&= \mathrm{shift}(x'(u); i' \cdot 2^{r'})_{(\ell \bmod 2^{r'})} \\
&\quad \text{(using that } 2^{r'} \text{ divides } 2^{b-a+1}\text{)} \\
&= x'_{([(\ell \bmod 2^{r'}) - i'2^{r'}] \bmod 2^{b-a+1}) \cdot 2^a + u} \\
&= x'_{([(\ell \bmod 2^{r'}) - i2^{r'}] \bmod 2^{b-a+1}) \cdot 2^a + u} \\
&= x'_{((\ell \cdot 2^a \bmod 2^r) - i2^r) \bmod 2^{b+1} + u} \\
&\quad \text{(using } (c \bmod m) \cdot w = cw \bmod (mw)\text{)} \\
&= x'_{2^{b+1} + (\ell \cdot 2^a \bmod 2^r) - (i2^r \bmod 2^{b+1}) + u} \\
&\quad \text{(since } p\text{, a multiple of } 2^r\text{, is } \neq 0 \bmod 2^{b+1}\text{, and } s = 1\text{)} \\
&= x_{[2^{b+1} + ((\ell \cdot 2^a + u) \bmod 2^r) - (i2^r \bmod 2^{b+1})] - \tilde{p} - 2^{b+1}} \\
&= x_{(t \bmod 2^r) - (\tilde{p} + (p \bmod 2^{b+1}))} \\
&= x_{(t \bmod 2^r) - p} \ ,
\end{aligned}$$

as needed. Finally, the case $p = 0 \bmod 2^{b+1}$ is handled identically except that we let $x'_t := x_{t-\tilde{p}}$. The analysis is very similar. Combining these two circuits adds a factor of 2 to our bound on the wires, which gives the result stated in item 1 above.

(2.) As earlier, we may assume the input to our circuit satisfies $||y|| = 1$, since the other case is easily handled using $\mathcal{O}(N)$ additional wires in the circuit. On Level 1 of C', we place gates $p_{k-g}, \ldots, p_{k-1}$, each wired to all the bits of y and to r; these gates output the g most significant bits of $p := (i \cdot 2^r \bmod n)$, where $i = i(y)$ is the unique index for which $y_i = 1$. This takes $g \cdot (n + \lceil \log_2 k \rceil) = \mathcal{O}(gN)$ wires. Also on Level 1, we include gates $h_1, \ldots, h_{2\lceil\sqrt{n}\rceil}$ computing the operator $H(y) = H_{sta}(y) : \{0,1\}^n \to \{0,1\}^{2\lceil\sqrt{n}\rceil}$ from Lemma 1, item 1, that is injective on $\{e_1, \ldots, e_n\}$ and computable in depth 1 with $2n$ wires.

On Level 2 of C', we include gates which compute the quantities (i', r', p') as defined in part 1 of Lemma 11, relative to $i = i(y)$, r, and $p := i \cdot 2^r \bmod n$. These quantities can be computed with $\mathcal{O}(\ln n)$ gates, each wired to r and to $h_1, \ldots, h_{2\lceil\sqrt{n}\rceil}$ (since the value of $H(y)$ determines $i(y)$). This group of gates requires $\mathcal{O}(\sqrt{n} \ln n) = o(N)$ input wires overall.

Also on Level 2 of C', we include gates $x'_0, \ldots, x'_{2^{k-g}-1}$, defined just as in part 1 of the Lemma, in terms of x and p. Note that we can compute these values with $\mathcal{O}(N)$ wires just as in part 1, since we have computed the g most significant bits of p on Level 1.

Levels 3 through $d+3$ of C' are identical to Levels 2 through $d+2$ of our circuit from part 1 (i.e., the full circuit, combining the two cases we considered). Correctness is proved exactly as before, and the number of gates is $2^{a+1} s_d\left(\mathrm{DIR}_{N'}^{bin,[0,b-a]}\right) + \mathcal{O}(gN)$, as required. □

Lemma 12. *$s_5(\mathrm{DIR}_N) = \mathcal{O}(N \ln\ln N) = \mathcal{O}(N \cdot \lambda_3(N))$.*

Proof. As usual we may assume $||y|| = 1$, solving the other case with $\mathcal{O}(N)$ additional wires. The idea for our construction is that we will handle the case when $2^r \leq n/\log_2 n$ by "shrinking" the input with Lemma 11, then applying our depth-2 construction from Lemma 10. We can handle the case $2^r > n/\log_2 n$ by a more straightforward approach since there are only $\approx \log_2 \log_2 n$ possible values of r in this range.

For any choice of $b < k$, it follows from the definition of DIR_N that we can write

$$(\mathrm{DIR}_N)_j = \left(\mathrm{DIR}_N^{[0,b]}\right)_j \vee \bigvee_{b<a<k} \left(\mathrm{DIR}_N^{[a,a]}\right)_j , \quad \forall j \in \mathbb{Z}_n \quad . \tag{5}$$

Set b as the largest value for which $2^b \leq n/\log_2 n$. By part 2 of Lemma 11 with $a := 0$, $\mathrm{DIR}_N^{[0,b]}$ can be computed in depth $5 = 2 + 3$ with $2 \cdot s_2\left(\mathrm{DIR}_{N'}^{bin,[0,b]}\right) + \mathcal{O}(N(k-b))$ wires, where $N' = 2 \cdot 2^{b+1} + \lceil \log_2(b+1) \rceil$. By Lemma 10, $s_2\left(\mathrm{DIR}_{N'}^{bin,[0,b]}\right) = \mathcal{O}(N' \ln N') = \mathcal{O}(2^{b+1}(b+1)) = \mathcal{O}((n/\log_2 n) \cdot \log_2 n) = \mathcal{O}(n)$. Also, $k - b \leq \log_2 \log_2 n + \mathcal{O}(1)$. Thus the total cost to compute $\mathrm{DIR}_N^{[0,b]}$ in depth 5 is $\mathcal{O}(N \ln \ln N)$.

To compute each of $\mathrm{DIR}_N^{[b+1,b+1]}, \ldots, \mathrm{DIR}_N^{[k-1,k-1]}$, we first compute $H(y)$, where $H = H_{sta}$ is the mapping defined within part 1 of Lemma 1; H is injective on $\{e_1, \ldots, e_n\}$ and computable in $2n$ wires. On Level 2, we use $(H(y), r)$ to compute $i = i(y)$ and $p := i \cdot 2^r \bmod n$; this takes $\mathcal{O}(\sqrt{n} \ln n) = o(n)$ wires. Then we use the depth-2 circuits C^a from Lemma 9 to compute $\mathrm{DIR}_N^{bin,[a,a]}(x, i, r, p)$ for $a = \{b+1, \ldots, k-1\}$, which give the outputs of $\mathrm{DIR}_N^{[b+1,b+1]}, \ldots, \mathrm{DIR}_N^{[k-1,k-1]}$ we need. Each C^a has $\mathcal{O}(n)$ wires, so the total cost of computing $\mathrm{DIR}_N^{[b+1,b+1]}, \ldots, \mathrm{DIR}_N^{[k-1,k-1]}$ is $\mathcal{O}(n(k-b)) = \mathcal{O}(n \ln \ln n)$.

At Level 5 of our circuit, we combine the outputs of all of our subcircuits: we "merge" the gates giving the values $\left(\mathrm{DIR}_N^{[0,b]}\right)_j, \left(\mathrm{DIR}_N^{[b+1,b+1]}\right)_j, \ldots, \left(\mathrm{DIR}_N^{[k-1,k-1]}\right)_j$ into a single output gate z_j computing the OR of these values. By Eq. (5), this circuit computes DIR_N; it is of depth 5 and contains $\mathcal{O}(N \ln \ln N)$ wires. This proves the Lemma. □

The next lemma, our key algorithmic tool for depths $d > 5$, gives an inductive construction of ever-more-efficient circuits for $\mathrm{DIR}_N^{bin,[0,k-1]}$ at the cost of increasing the circuit depth.

Lemma 13. *For even values $d = d(N) \geq 2$, we have $s_d\left(\mathrm{DIR}_N^{bin,[0,k-1]}\right) = \mathcal{O}\left(N \cdot \lambda_d(N)\right)$. The $\mathcal{O}(\cdot)$ is independent of d.*

Proof. Let $C > 0$ be chosen larger than the implicit constants in the $\mathcal{O}(\cdot)$-notation used in all of our previous results, when the bounds are, for convenience, *re-expressed* in terms of the parameter $n = \Theta(N)$; recall that in each case the bound was independent of d and the other parameters. We claim, and prove by induction on even $d \geq 2$, that $s_d\left(\mathrm{DIR}_N^{bin,[0,k-1]}\right) < 40Cn \cdot \lambda_d(n)$. We may assume in what follows that $k > 20$, setting C large enough that the claim is trivially true for $k \leq 20$.

For $d = 2$, Lemma 10 gives $s_2\left(\mathrm{DIR}_N^{bin,[0,k-1]}\right) < Cn \cdot \lambda_2(n)$, as needed. Now let $d \geq 4$ be even, and consider the statement proved for $d' = d - 2$. First, if $\lambda_{d-2}(n) = 1$, the result is trivial; so assume from now on that $\lambda_{d-2}(n) \geq 2$. Define a nondecreasing integer sequence $a_1, a_2, \ldots, a_T$, where

$$a_t \;:=\; \lfloor \log_2(n/\lambda_{d-2}^{(t)}(n)) - 20 \rfloor$$

(recalling that $g^{(t)}$ denotes the t-fold composition of g). We let $T := \min\{t : \lambda_{d-2}^{(t)}(n) = 1\}$; thus $T = \lambda_{d-2}^{*}(n) = \lambda_d(n)$ by the definitions. It is immediate that $\lambda_{d-2}(m) \geq 1$ whenever $m > 1$, so in fact $\lambda_{d-2}^{(T)}(n) = 1$ and all the a_t's are well-defined, with $a_T = k - 20$. Also, $T > 1$ by our assumption $\lambda_{d-2}(n) \geq 2$.

Let $t^* := \min\{t \in [T] : a_t > 0\}$. As $a_T = k - 20$, we can express the interval $[0, k-1]$ as

$$[0, k-1] \;=\; [0, a_{t^*}] \cup [a_{t^*}, a_{t^*+1}] \cup \ldots \cup [a_{T-1}, a_T] \cup [k-19, k-1] \;,$$

and for $j \in \mathbb{Z}_n$ we can write

$$\left(\mathrm{DIR}_N^{bin,[0,k-1]}\right)_j \;=\; \left(\mathrm{DIR}_N^{bin,[0,a_{t^*}]}\right)_j \vee \left(\mathrm{DIR}_N^{bin,[k-19,k-1]}\right)_j \vee \bigvee_{t=t^*+1}^{T} \left(\mathrm{DIR}_N^{bin,[a_{t-1},a_t]}\right)_j \;. \quad (6)$$

By the same technique used in Lemma 12, one can "merge" the outputs of depth-d circuits for the operators $\mathrm{DIR}_N^{bin,[0,a_{t^*}]}$, $\mathrm{DIR}_N^{bin,[a_{t^*+1},a_{t^*+2}]}$, $\ldots$, $\mathrm{DIR}_N^{bin,[a_{T-1},a_T]}$, and $\mathrm{DIR}_N^{bin,[k-19,k-1]}$ to get a depth-d circuit for $\mathrm{DIR}_N^{bin,[0,k-1]}$.

Let $\tilde{n} := 2^{a_{t^*}+1}$, $\tilde{N} := 2 \cdot 2^{a_{t^*}+1} + \lceil \log_2(a_{t^*} + 1) \rceil$. Applying Lemma 11, part 1 (with $a := 0, b := a_{t^*}$), we find that

$$s_d\left(\mathrm{DIR}_N^{bin,[0,a_{t^*}]}\right) \;\leq\; 2 \cdot s_{d-2}\left(\mathrm{DIR}_{\tilde{N}}^{bin,[0,a_{t^*}]}\right) + Cn \;.$$

If $t^* = 1$, then $2^{a_{t^*}+1} \leq 2^{-19} \cdot (n/\lambda_{d-2}(n))$, and, using the inductive hypothesis,

$$s_{d-2}\left(\mathrm{DIR}_{\tilde{N}}^{bin,[0,a_{t^*}]}\right) \;\leq\; .005C \cdot (n/\lambda_{d-2}(n)) \cdot \lambda_{d-2}(n) \;,$$

so that $s_d(\mathrm{DIR}_N^{bin,[0,a_{t^*}]}) < 1.01Cn$. If $t^* > 1$, then $a_{t^*-1} \leq 0$, so $2^{-a_{t^*-1}} \geq 1$ and

$$\tilde{n} \;=\; 2^{a_{t^*}+1} \;\leq\; 2 \cdot 2^{a_{t^*} - a_{t^*-1}} \;\leq\; \lceil 4 \cdot \lambda_{d-2}^{(t^*-1)}(n)/\lambda_{d-2}^{(t^*)}(n) \rceil \;,$$

and

$$\begin{aligned} s_{d-2}\left(\mathrm{DIR}_{\tilde{N}}^{bin,[0,a_{t^*}]}\right) \;&\leq\; 40C \cdot \lceil 4 \cdot \lambda_{d-2}^{(t^*-1)}(n)/\lambda_{d-2}^{(t^*)}(n) \rceil \cdot 4\lambda_{d-2}(\lambda_{d-2}^{(t^*-1)}(n)) \\ &\leq\; 640C\lambda_{d-2}^{(t^*-1)}(n) \\ &<\; Cn \end{aligned}$$

(here using $n = 2^k > 2^{20}$ and $t^* > 1$), so that $s_d(\mathrm{DIR}_N^{bin,[0,a_{t^*}]}) \leq 2Cn$ in this case.

Now consider $t \in [t^* + 1, T]$. By Lemma 11, part 1, we have

$$s_d(\mathrm{DIR}_N^{bin,[a_{t-1},a_t]}) \;\leq\; 2^{a_{t-1}+1} \cdot s_{d-2}(\mathrm{DIR}_{N_t}^{bin,[0,a_t - a_{t-1}]}) + Cn \;,$$

where

$$N_t \ := \ 2 \cdot 2^{a_t - a_{t-1}+1} + \lfloor \log_2(a_t - a_{t-1} + 1) \rfloor \ .$$

Now $2^{a_t - a_{t-1}+1} \leq \lceil 4 \cdot \lambda_{d-2}^{(t-1)}(n) / \lambda_{d-2}^{(t)}(n) \rceil$, so, using the inductive hypothesis, $s_{d-2}(\mathrm{DIR}_{N_t}^{bin,[0,a_t-a_{t-1}]})$ is at most

$$40C \cdot \lceil 4 \cdot \lambda_{d-2}^{(t-1)}(n) / \lambda_{d-2}^{(t)}(n) \rceil \cdot (4\lambda_{d-2}(\lambda_{d-2}^{(t-1)}(n)) \ = \ 640C\lambda_{d-2}^{(t-1)}(n) \ .$$

Thus, $s_d(\mathrm{DIR}_N^{bin,[a_{t-1},a_t]})$ is at most

$$2^{a_{t-1}+1}C \cdot (640\lambda_{d-2}^{(t-1)}(n)) + Cn \ < \ 1.01Cn \ ,$$

using the definition of a_{t-1}.

Finally, $\mathrm{DIR}_N^{bin,[k-19,k-1]}$ can be computed with $19Cn$ wires, using 19 applications of Lemma 9. Combining our cases and applying them to Eq. (6), we find that $s_d(\mathrm{DIR}_N^{bin,[0,k-1]})$ is less than $19Cn + 2Cn + T \cdot (1.01Cn) < 40Cn \cdot \lambda_d(n)$, since $T = \lambda_d(n)$. This extends the induction to d, completing the proof. □

Lemma 14. *For even $d \geq 6$, we have $s_d(\mathrm{DIR}_N) = \mathcal{O}(N \cdot \lambda_{d-2}(N))$; the $\mathcal{O}(\cdot)$ is independent of d.*

Proof. As usual, we may assume the input satisfies $||y|| = 1$ (handling the case $||y|| \neq 1$ separately with $\mathcal{O}(N)$ additional wires).

On Levels 1 and 2 of our circuit C for $\mathrm{DIR}_N(x, y, r)$, we compute $i = i(y)$ and $p := i \cdot 2^r \bmod n$ with $\mathcal{O}(N)$ wires, by applying the mapping $H = H_{sta}$ from Lemma 1, part 1 to y and then applying a brute-force circuit to $(H(y), r)$. Then we apply an optimal depth-$(d-2)$ circuit for $\mathrm{DIR}_N^{bin,[0,k-1]}$ to the tuple (x, i, r, p). This yields the desired output. The number of wires in our circuit is $s_{d-2}(\mathrm{DIR}_N^{bin,[0,k-1]}) + \mathcal{O}(N)$, and by Lemma 13 this is $\mathcal{O}(N \cdot \lambda_{d-2}(N))$. □

By collecting the upper bounds for DIR_N in Lemmas 8, 10, 12 and 14, along with the lower bounds we get from Theorem 5 and Lemma 7, we have proved Theorem 6.

5 Limits of Jukna's entropy method, and a separation of depths 2 and 3

In this section, we show:

Theorem 15. *There is a family of operators $MM' : \{0,1\}^{2n} \to \{0,1\}^n$ for which $s_2(MM') = \Omega(n^{3/2})$ while $s_3(MM') = \mathcal{O}(n)$.*

The notation MM' indicates that our operator is a modified (simplified) form of matrix multiplication. The lower bound on MM' for depth 2 will be proved using Jukna's entropy method, Theorem 4. This example shows that the entropy method cannot be used to prove super-linear wire lower bounds in depth 3.

Proof of Theorem 15. For any integer $n > 0$, we can find a perfect square $n' = m^2$ in the range $[n, 2n]$. Thus to prove our asymptotic statement, we may assume that $n = m^2$ is itself a perfect square.

We regard the input to MM' as two matrices $X, Y \in \mathbb{F}_2^{m\times m}$. The output is a third matrix $Z \in \mathbb{F}_2^{m\times m}$. Define

$$MM'(X,Y) := \begin{cases} X \cdot Y & \text{if } X \text{ contains exactly one 1-entry,} \\ 0 & \text{otherwise.} \end{cases}$$

Claim 16. $s_2(MM') \geq m^3 = n^{3/2}$.

Proof. The argument is basically identical to the one used in [Juk10a] to show that multiplying two m-by-m matrices in depth 2 requires m^3 wires. Letting $p := m + 1$, we set up and apply Theorem 4. For $t \in [p-1]$, let I_t be the t-th row of input matrix X, and let J_t be the t-th row of output matrix Z. (Thus $I_p = Y$, $J_p = \emptyset$, and the p-th part will contribute nothing to the lower bound from Theorem 4.)

Fix any $t \in [m]$, and values $k, \ell \in [m]^2$. Let $X^{(t,k)}$ be the matrix whose (t,k)-entry is 1 and whose other entries are 0. Note that for any Y,

$$\left(MM'\left(X^{(t,k)}, Y\right)\right)_{t,\ell} \;=\; \left(X^{(t,k)} \cdot Y\right)_{t,\ell} \;=\; Y_{k,\ell}\ .$$

Thus any desired bit of the matrix Y can be recovered from a value in the t-th row of $MM'(X,Y)$ (i.e., in the output block J_t), for some setting to X which has a single 1-entry in the t-th column. Thus Y can be determined from values of $MM'_{I_t,J_t}(0^{m\times m}, Y)$. It follows from Facts 2 and 3 that $\mathrm{Ent}(MM'_{I_t,J_t}) \geq m^2$, so by Theorem 4, $s_2(MM') \geq m \cdot m^2 = m^3$. □

Now we show that $s_3(MM') = \mathcal{O}(n)$ by giving a depth-3 circuit C for MM' with $\mathcal{O}(n)$ wires.

First, on Level 1 we define $2\lceil\sqrt{n}\rceil = 2m$ "hash gates" $h_1, \ldots, h_{2m}$, which compute the linear transformation $H_{sta}(X) = (h_1(X), \ldots, h_{2m}(X)) : \mathbb{F}_2^n \to \mathbb{F}^{2m}$ given by item 1 of Lemma 1, applied to the input matrix X. Define $\mathbf{1}_{(i,j)} \in \mathbb{F}_2^{2m}$ as the vector obtained by applying H_{sta} to the input matrix $X^{(i,j)}$ which contains a single 1-entry in its $(i,j)^{th}$ position. By Lemma 1 the vectors $\mathbf{1}_{(i,j)}$ are pairwise distinct and each of Hamming weight 2.

On Level 1 we also include a "security gate" s. This gate is connected to all the variables in X, and outputs 1 if X has exactly one 1-entry, or 0 otherwise.

Next, on Level 2 we will have a set of "row gates" $r_1, \ldots, r_m$, and "column gates" $c_1, \ldots, c_m$. The row gate r_k takes $h_1, \ldots, h_{2m}$ and s as inputs. We define

$$r_k := \begin{cases} 1 & \text{if } s = 1 \text{ and } (h_1, \ldots, h_{2m}) = \mathbf{1}_{(k,j)} \text{ for some } j \in [m], \\ 0 & \text{otherwise.} \end{cases}$$

The column gate c_ℓ takes $h_1, \ldots, h_{2m}$, and the ℓ-th column of Y as inputs. We define

$$c_\ell := \begin{cases} Y_{j,\ell} & \text{if } (h_1, \ldots, h_{2m}) = \mathbf{1}_{(k,j)} \text{ for some } k \in [m], \\ 0 & \text{otherwise.} \end{cases}$$

Finally, for $k, \ell \in [m]$, on Level 3 we let $Z_{k,\ell}$ be the AND of r_k and c_ℓ.

We argue that C computes MM'. First suppose that X does not have exactly one 1-entry. Then $s = 0$, so all row gates are 0 and $Z = 0^{m \times m}$ as required. Next, suppose X has a single 1-entry in the (i, j) position. Then we have $(h_1, \ldots, h_{2m}) = \mathbf{1}_{(i,j)}$, and $s = 1$. It follows that for $k \in [m]$, we have $r_k = [k = i]$. Also, $(c_1, \ldots, c_m) = (Y_{j,1}, \ldots, Y_{j,m})$. Thus for $\ell \in [m]$, we have $Z_{k,\ell} = [k = i] \wedge Y_{j,\ell}$. This is precisely the (k, ℓ)-entry of $MM'(X, Y)$. Thus C computes MM'.

Finally, we count the wires in C. The subcircuit computing $H_{sta}(X)$ has $\mathcal{O}(n)$ gates, by Lemma 1. The security gate has $m^2 = n$ inputs. Each row and column gate has at most $2m$ inputs, for a total of $\leq (2m)^2$ wires as input to a row or column gate. Each output $Z_{k,\ell}$ has 2 inputs, so the total number of wires is $\mathcal{O}(n)$ as desired. This completes the proof of Theorem 15. □

6 Representing random linear operators

In the rest of the paper, we study the wire complexity of computing and representing linear transformations. (Recall the notion of representing a linear operator L relative to a basis B, and the quantities $R(L; B)$ and $R_d(L; B)$, from Section 2.2.)

Jukna [Juk10b] showed:

Theorem 17. *[Juk10b] Every linear operator $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$ can be represented by a depth-2 circuit of $\mathcal{O}(n \ln n)$ wires relative to the standard basis.*

An easy modification of his proof shows that for any linear operator $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and any basis B for $\mathbb{F}_2^n$, $R_2(L; B) = \mathcal{O}(n \ln n)$. We show that Jukna's upper bound is optimal up to constant factors, by proving the following lower bound on the wire complexity of representing random linear operators:

Theorem 18. *Fix any basis B for $\mathbb{F}_2^n$. Suppose a random linear operator $L = L_A : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is defined by uniformly selecting its defining matrix $A \in \mathbb{F}_2^{n \times n}$. With probability $1 - o(1)$, we have*

$$R(L; B) \; > \; \frac{n \log_2 n}{5} \; .$$

Theorem 18 is implied by the following more general result about random *partial* operators (not necessarily linear):

Theorem 19. *Let $D \subseteq \{0, 1\}^n$ be of size $r = r(n) > 1$, and assume $r/\log_2 r \geq \log_2 n$. Then a $(1 - o_n(1))$ fraction of all partial operators $F : D \to \{0, 1\}^n$ satisfy*

$$s(F) \; > \; \frac{n \log_2 r}{5} \; .$$

The constant 1/5 is not optimal, and we do not attempt to optimize it here.

Proof of Theorem 18. L is distributed as a uniformly random partial operator from B to $\{0, 1\}^n$ when we consider its restriction to inputs from a linearly independent set B. Thus the result follows immediately from Theorem 19. □

Proof of Theorem 19. We first define an augmented circuit model. Fix a canonical ordering $D = \{x^1, \ldots, x^r\}$ of the possible input strings. In a *free-ID circuit*, the input $x = x^i \in D$ is given along with inputs $z_1, \ldots, z_{\lceil \log_2 r \rceil}$ which give the binary encoding of $i \in [r]$. That is, the circuit is provided with this unique identifier of x "for free," and these bits can be used as inputs to any gate. As before, we can use any function at the circuit gates.

Let $s_{free-ID}(F)$ denote the minimal number of wires in any free-ID circuit which computes F. It is clear that $s_{free-ID}(F) \le s(F)$, so to prove the Theorem it is enough to prove that $s_{free-ID}(F) > n \log_2 r/5$ holds for a $(1 - o(1))$ fraction of all $F : D \to \{0,1\}^n$.

Suppose $F : D \to \{0,1\}^n$ satisfies $s_{free-ID}(F) \le n \log_2 r/5$, and let C be an optimal (wire-minimizing) circuit with at most $L := \lfloor n \log_2 r/5 \rfloor$ wires computing F. Besides the input and free-ID gates, all gates with fanin zero are constant (0 or 1), so we may assume C contains at most two such gates. Each wire is input to just one gate, so we can assume the total number of gates of C (inclusive of inputs and free-ID gates) is at most $L+n+\lceil \log_2 r \rceil+2$. We can then reintroduce useless (fanin-zero) gates as necessary to get exactly this many gates.

Next we make a simple, key observation: optimality of C implies that all gates of C have fanin at most $\lceil \log_2 r \rceil$. To see this, suppose that some gate g of C has fanin greater than $\lceil \log_2 r \rceil$. The value of g on any input $x = x^i$ is determined by i, and hence by the ID variables $z_1, \ldots, z_{\lceil \log_2 r \rceil}$. Thus we can rewire g to have the inputs $z_1, \ldots, z_{\lceil \log_2 r \rceil}$ and output the same result. This modified circuit still computes F but has fewer wires than C, contradicting the minimality of C.

Now we upper-bound the number (call it N_L) of free-ID circuits with at most L wires, exactly $m := L + n + \lceil \log_2 r \rceil + 2$ gates (including the n input gates and $\lceil \log_2 r \rceil$ free-ID gates), and maximum fanin $\lceil \log_2 r \rceil$. By our reasoning above, this will bound the number of operators $F : D \to \{0,1\}^n$ for which $s_{free-ID}(F) \le n \log_2 r/5$. Our calculations will follow similar ones in [JS10] with minor modifications.

There are at most $(\log_2 r + 2)^m$ sequences of fanins $(d_1, \ldots, d_m)$ we may choose for our gates, where $0 \le d_i \le \lceil \log r \rceil$ and $\sum_{i \in [m]} d_i \le L$. For each such sequence and for $i \in [m]$, we can choose the inputs to the i-th gate in at most $\binom{m}{d_i} \le m^{d_i}$ ways, and there are at most $2^{2^{d_i}}$ Boolean functions to assign to this gate. Thus,

$$\begin{aligned} N_L &\le (\log_2 r + 2)^m \prod_{i \in [m]} m^{d_i} \prod_{j \in [m]} 2^{2^{d_j}} \\ &= (\log_2 r + 2)^m \, m^{\sum_{i \in [m]} d_i} 2^{\sum_{j \in [m]} 2^{d_j}} \\ &\le (\log_2 r + 2)^m \, m^L 2^{\sum_{j \in [m]} 2^{d_j}} . \end{aligned}$$

Taking logs,

$$\begin{aligned} \log_2(N_L) &\le m\,(\log_2 r + 2) + L \log_2 m + \sum_{j \in [m]} 2^{d_j} \\ &\le 2L \log_2 L + \sum_{j \in [m]} 2^{d_j} + o(L \log_2 L) \ , \end{aligned}$$

by our settings of m, L.

In a circuit of L wires, counting tells us that fewer than $n/4$ gates have fanin larger than $4L/n$. Since no gate has fanin larger than $\lceil \log_2 r \rceil$, we have

$$\begin{aligned}\sum_{j\in[m]} 2^{d_j} &\leq m2^{4L/n} + (n/4)2^{\lceil \log_2 r\rceil} \\ &\leq m2^{4\log_2 r/5} + nr/2 \\ &= nr/2 + Lr^{4/5} + o(Lr^{4/5}) \ .\end{aligned}$$

Thus,

$$\log_2(N_L) \leq 2L\log_2 L + \left(nr/2 + Lr^{4/5}\right) + o\left(L\log_2 L + Lr^{4/5}\right) . \tag{7}$$

We have $Lr^{4/5} \leq nr^{4/5}\log_2 r/5 = o(nr)$. Also,

$$\begin{aligned}L\log_2 L &\leq (1/5)n\log_2 r\log_2(n\log_2 r/5) \\ &< (1/5)n\left[(\log_2 r)(\log_2 n) + \log_2^2 r\right] \\ &\leq (1/5)n\left[r + \log_2^2 r\right] \ ,\end{aligned}$$

using our initial assumption $r/\log_2 r \geq \log_2 n$. Plugging into Eq. (7), we find $\log_2(N_L) < nr/2 + 2\cdot nr/5 + o(nr)$, so for sufficiently large n, $\log_2(N_L) < .95nr$ and $N_L < 2^{.95nr}$.

Finally we compare this to the number of partial operators $F : D \to \{0,1\}^n$. For each of the r inputs in D, there are 2^n possible outputs, so we have $(2^n)^r = 2^{nr}$ many partial operators in total. Thus less than a $2^{-.05nr} = o(1)$ fraction of these satisfy $s_{free-ID}(F) \leq n\log_2 r/5$. □

7 Tightness of Jukna's pairwise-distance lower bound for depth 2

Given $A \in \mathbb{F}_2^{n\times m}$, let $\mathrm{Dist}(A) \in \{0, 1\ldots, n\}$ denote the minimal Hamming distance between any two columns of A. Building on [AKW90], Jukna gave a lower-bound criterion for representing linear operators (proved for the case $n = m$, although a similar result can be given for other cases as well):

Theorem 20. *[Juk10b] For $A \in \mathbb{F}_2^{n\times n}$, every depth-2 circuit representing the linear transformation $x \to Ax$ relative to the standard basis must have at least $\Omega\left(\mathrm{Dist}(A)\cdot \frac{\ln n}{\ln\ln n}\right)$ wires.*

For random matrices and for some explicit examples, we have $\mathrm{Dist}(A) = \Omega(n)$. In this case, Theorem 20 gives a lower bound of $\Omega\left(\frac{n\ln n}{\ln\ln n}\right)$, which nearly matches the upper bound from Theorem 17. It was left open in [Juk10b] whether the $(\ln\ln n)^{-1}$ factor in the bound from Theorem 20 could be removed, leading to matching upper and lower bounds for a large class of matrices. In this section we show that this cannot be done: there are, in fact, linear transformations with $\mathrm{Dist}(A) = \Omega(n)$, for which the lower bound from Theorem 20 is tight.

Theorem 21. *There exists a family of matrices $\{A_n \in \mathbb{F}_2^{n\times n}\}_{n>0}$ for which $\mathrm{Dist}(A_n) = \Omega(n)$, and for which the linear transformation $x \to A_n x$ over $\mathbb{F}_2^n$ can be computed by a depth-2 circuit with $\mathcal{O}\left(\frac{n\ln n}{\ln\ln n}\right)$ wires.*

Note that the linear transformations we define can be *computed*, not just represented, using $\mathcal{O}\left(\frac{n \ln n}{\ln \ln n}\right)$ wires. Now, Theorem 18 states that random matrices A require $\Omega(n \ln n)$ wires even to *represent* by a circuit of any depth. Thus, the difficulty of representing random matrices is not "fully captured" by the property that their columns have pairwise distance $\Omega(n)$.

Our main tool to prove Theorem 21 is a *combinatorial design*, or set family, defined by low-degree polynomials. Set families of this kind have seen several applications in complexity theory, e.g., in [NW94]. The form we use is given in the following Claim. For a set X, $\mathcal{P}(X)$ denotes the collection of all subsets of X.

Claim 22. *For any integer $D > 1$, there is an prime $q = \mathcal{O}(D)$ and a set family $\mathcal{S} \subset \mathcal{P}(\mathbb{F}_q^2)$, which contains at least D^D sets and satisfies:*

(i) $|S_i| = q$ for all $S_i \in \mathcal{S}$, and for each $a \in \mathbb{F}_q$ we have $|S_i \cap (a \times \mathbb{F}_q)| = 1$;

(ii) $|S_i \cap S_j| \leq q/2$ for all $i \neq j$.

Proof. Let q be a prime number in the range $[2D, 4D]$, as guaranteed to exist by Bertrand's postulate. For each nonzero polynomial $P(x) \in \mathbb{F}_q[x]$ of degree at most D, define $S_P \in \mathcal{P}(\mathbb{F}_q^2)$ as the "graph of P,"

$$S_P \ := \ \{(a, b) \in \mathbb{F}_q^2 : P(a) = b\} \ .$$

Let $\mathcal{S}$ contain the sets S_P for each such polynomial. These polynomials are in 1-to-1 correspondence with $\mathbb{F}_q^{D+1} \setminus \{\mathbf{0}\}$, by the mapping which sends a nonzero polynomial to its vector of coefficients. Thus the number of such polynomials is $q^{D+1} - 1 \geq D^D$ and $|\mathcal{S}| \geq D^D$.

It is clear that condition (i) holds, since each S_P is a graph. For condition (ii), note that $(a, b) \in S_P \cap S_Q$ exactly when $P(a) = Q(a) = b$, and distinct polynomials of degree at most D agree on at most $D \leq q/2$ values. □

A *sunflower of size k* is a collection of distinct sets $A_1, \ldots, A_k$ (called *petals*), such that the pairwise intersections $A_i \cap A_j$, for $i \neq j$, are all equal to some fixed set C (called the *core*). The lower bound technique of Theorem 20 works by finding a large sunflower in the incidence pattern of wires in a circuit, and using this sunflower to identify an information bottleneck. The set family $\mathcal{S}$ in Lemma 22 is a prototypical example of a set family which does not contain sunflowers of too-large size: all sunflowers in $\mathcal{S}$ have size at most q. Thus it is natural to try to use such a set family to show the tightness of Theorem 20. These were the considerations that led us to the proof of Theorem 21 given below.

Proof of Theorem 21. We are going to define the matrix $A_n \in \mathbb{F}_2^{n \times n}$, and its associated transformation L_{A_n}, by defining a depth-2 linear circuit C_n that computes L_{A_n}. Our construction will work for sufficiently large n.

For $y \geq 1$, define $\beta(y) \in \mathbb{R}^+$ as the unique positive solution to $x^x = y$. Direct computation shows that for large n we have $\frac{\ln n}{\ln \ln n} < \beta(n) \leq \frac{(1+o(1)) \ln n}{\ln \ln n}$.

Let $\mathcal{S} \subset \mathcal{P}(\mathbb{F}_q^2)$ be the set family given by Lemma 22, with the setting $D := \lceil \beta(n) \rceil$. We have $|\mathcal{S}| \geq D^D \geq n$ (by definition of $\beta(n)$), so we can assign a distinct set $S_i \in \mathcal{S}$ to each input coordinate $i \in [n]$.

Now we describe the middle level (Level 1) of our depth-2 circuit C_n. Let $q = \mathcal{O}(D)$ be the prime number used in defining $\mathcal{S}$. Level 1 of C_n consists of q^2 gates $g_{(a,b)}$ identified with the elements of $\mathbb{F}_q^2$. For $i \in [n]$, the i-th input gate is connected to the middle gate $g_{(a,b)}$ for each $(a, b) \in S_i$. Each $g_{(a,b)}$ is the sum mod 2 of its inputs:

$$g_{(a,b)} := \bigoplus_{i:(a,b)\in S_i} x_i \ .$$

For the output level, we divide the n output bits into q contiguous blocks $B_1, \ldots, B_q$, each of size $|B_a| = \lfloor n/q \rfloor$; the remaining output bits will be identically zero. We re-index the output gates in B_a as

$$B_a = (z_{a,1}, \ldots, z_{a,\lfloor n/q \rfloor}) \ .$$

We fix a collection $V = \{v^0, \ldots, v^{q-1}\} \subseteq \mathbb{F}_2^{\lfloor n/q \rfloor}$, such that any pair of vectors from V disagree on at least a 1/3 fraction of coordinates. This can clearly be achieved for large n, since $\lfloor n/q \rfloor = \omega(q)$. We think of v^a as an "error-correcting encoding" of a. We determine the output bits as

$$z_{a,\ell} := \bigoplus_{0\le b<q} v_\ell^b \cdot g_{a,b} = \bigoplus_{0\le b<q:\ v_\ell^b=1} g_{a,b} \ ,$$

where v_ℓ^b is the ℓ-th bit of v^b. This completes the description of C_n.

Note that each input gate and output gate in C_n is connected to at most $q = \mathcal{O}(\beta(n))$ middle gates, so the total number of wires is $\mathcal{O}(n \cdot \beta(n)) = \mathcal{O}\left(\frac{n \ln n}{\ln \ln n}\right)$, as desired. Also, C_n is $\mathbb{F}_2$-linear as promised, so it defines a matrix $A_n \in \mathbb{F}_2^{n\times n}$ by the relation $C_n(x) \equiv A_n x$. We now argue that $\mathrm{Dist}(A_n) = \Omega(n)$. It is equivalent to show that for each pair $i, j \in [n]$ of distinct indices, $C_n(e_i)$ and $C_n(e_j)$ disagree on $\Omega(n)$ positions.

Fix any $i \in [n]$. For any $a \in \mathbb{F}_q$, condition (i) of Lemma 22 tells us that the intersection $S_i \cap (a \times \mathbb{F}_q)$ consists of a single element of $\mathbb{F}_q$; call this element $b_i(a)$. We verify that on input vector e_i we have $g_{a,c}(e_i) = 1$ iff $c = b_i(a)$. Thus, for $0 \le \ell < q$ we have $z_{a,\ell} = v_\ell^{b_i(a)}$, so that the restriction of $C_n(e_i)$ to the output block B_a equals $v^{b_i(a)}$.

If $j \in [n] \setminus \{i\}$, then condition (ii) of Lemma 22 tells us that for at least $q/2$ choices of a we have $b_i(a) \neq b_j(a)$. For such a, the restrictions $C_n(e_i)|_{B_a} = v^{b_i(a)}$, $C_n(e_j)|_{B_a} = v^{b_j(a)}$ disagree on at least 1/3 of their positions. So the total number of disagreements between $C_n(e_i), C_n(e_j)$ is at least

$$\frac{q}{2} \cdot \frac{\lfloor n/q \rfloor}{3} = \Omega(n) \ .$$

This shows $\mathrm{Dist}(A_n) = \Omega(n)$, completing the proof. □

By an easy refinement of our argument, for any $\delta > 0$ we can modify the matrices A_n in Theorem 21 to satisfy $\mathrm{Dist}(A_n) \ge (1/2 - \delta)n$ for sufficiently large n, while the resulting transformation $x \to A_n x$ is still computable in depth 2 with $\mathcal{O}_\delta\left(\frac{n \ln n}{\ln \ln n}\right)$ wires. A remaining question is whether we can have $\mathrm{Dist}(A_n) \ge n/2 - o(n)$, with $s_2(A_n) = \mathcal{O}\left(\frac{n \ln n}{\ln \ln n}\right)$. To achieve this we would need to change our approach, due to limits on the achievable parameters of combinatorial designs (see [RRV02, Prop. 14]).

It is also natural to wonder whether better lower bounds would be implied by a very strict column-distance condition on $A_n \in \mathbb{F}_2^{n\times n}$, namely $\mathrm{Dist}(A_n) = n/2$. This may be so;

however, in [AKW90] it was shown that the *Sylvester* matrices, which satisfy this condition, can be computed using $\mathcal{O}\left(n \cdot \lambda_k(n)\right)$ wires in depth $d = \mathcal{O}(k)$.

8 The pairwise-distance method fails for depth 3

In this section we show that Jukna's complexity measure $\mathrm{Dist}(A)$ (defined in Section 7) does not yield super-linear lower bounds for circuits of depths 3 and higher:

Theorem 23. *There exists a family of matrices $\{A_n \in \mathbb{F}_2^{n\times n}\}_{n>0}$ for which $\mathrm{Dist}(A_n) \geq n/2 - o(n)$, and such that the linear transformation $x \to A_n x$ over $\mathbb{F}_2^n$ can be computed by an $\mathbb{F}_2$-linear depth-3 circuit with $\mathcal{O}\left(n\right)$ wires.*

The work of Gál et al. [GHK+11] already implied the existence of a family $\{A_n \in \mathbb{F}_2^{n\times \Omega(n)}\}$ with $\mathrm{Dist}(A_n) = \Omega(n)$, whose associated linear transformations are computable by depth-3 linear circuits with $\mathcal{O}(n \ln\ln n)$ wires.

Proof of Theorem 23. We may assume, by padding if necessary, that the input length n is a perfect square, $n = m^2$. We will define A_n by defining the circuit C_n that computes it. Let $H = H_{sta} : \mathbb{F}_2^n \to \mathbb{F}_2^{2m}$ be the mapping given by item 1 of Lemma 1, with associated circuit C_H. Let $m' := 2m$. We let the outputs $h_1, \ldots, h_{m'}$ of H occupy Level 1 of C_n, and connect them to the inputs according to C_H. Thus for $e_i \in E$, the gates of C_n's first level compute $H(e_i)$, and the number of wires used for the first level is $\mathcal{O}(n)$.

Level 2 will also consist of m' gates, call them $F = (f_1, \ldots, f_{m'})$. Each f_i will be connected to a uniformly random subset of $\{h_1, \ldots, h_{m'}\}$, and will compute the sum over $\mathbb{F}_2$ of its inputs. This requires at most $\mathcal{O}(m^2) = \mathcal{O}(n)$ wires.

Finally, Level 3 consists of m' blocks of outputs, with each i-th block B_i of size $m/2$. For $i \in [m']$, each gate in B_i simply outputs f_i. Thus Level 3 requires $\mathcal{O}(n)$ wires, and the circuit uses $\mathcal{O}(n)$ wires in total. Also, C_n is an $\mathbb{F}_2$-linear circuit as promised, since C_H is linear.

We now show that C_n computes a transformation with the desired properties, with probability $1 - o(1)$ over our random choices. The m' gates on Level 2, considered as a linear transformation over the m' Level 1 gates, compute a uniformly random linear transformation from $\mathbb{F}_2^{m'}$ to itself; call this transformation $\widetilde{F}$.

Fix any pair $i, j \in [n]$ with $i \neq j$. Now, as distinct nonzero vectors in $\mathbb{F}_2^{m'}$, the pair $(H(e_i), H(e_j))$ map to uniform, independent images under $\widetilde{F}$. Letting $\Delta(\cdot,\cdot)$ denote Hamming distance, Chernoff-Hoeffding bounds imply that for $\alpha > 0$,

$$\Pr\left[\Delta\left(\widetilde{F}(H(e_i)), \widetilde{F}(H(e_j))\right) < m'/2 - \alpha\sqrt{m' \ln m'}\right] \;=\; \exp\left(-\Omega(\alpha^2 \ln m')\right) \;=\; o(n^{-2})\ ,$$

if α is a sufficiently large constant. By a union bound, with probability $1 - o(1)$ we have that $\Delta\left(\widetilde{F}(H(e_i)), \widetilde{F}(H(e_j)\right) \geq m'/2 - \mathcal{O}\left(\sqrt{m' \ln m'}\right) = m - o(m)$ for all $i, j \in [n]$, $i \neq j$. Note that by our definition of the output gates of C_n,

$$\Delta(C_n(e_i), C_n(e_j)) \;=\; (m/2) \cdot \Delta\left(\widetilde{F}(H(e_i)), \widetilde{F}(H(e_j)\right)\ ,$$

so with high probability, $\Delta(C_n(e_i), C_n(e_j)) \geq n/2 - o(n)$ for all $i, j \in [n]$, $i \neq j$. This proves Theorem 23. □

9 Easy bases for representing linear operators

For a linear transformation L, recall the quantities $R_d(L; B)$ and $R(L; B)$ from Section 2. Since computing a transformation is a stronger requirement than representing the transformation, we have

$$s_d(L) \ \geq \ \max_{B : B \text{ a basis for } \mathbb{F}_2^n} R_d(L; B) \ .$$

It is natural to wonder: how close are the left-hand and right-hand sides above? Note that for a random $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$, we have $s^{\oplus}(L) = \Omega(n^2 / \ln n)$, by a standard counting argument. Following Jukna and Schnitger [JS10], we suspect that also $s(L) = \Omega(n^2 / \ln n)$ for random L, but this is not known. It was shown by [GHK+11] that $s_2(L) = \Omega\left(n \left(\frac{\ln n}{\ln \ln n}\right)^2\right)$ if $L : \mathbb{F}_2^n \to \mathbb{F}_2^{\mathcal{O}(n)}$ is the encoding function for a good linear error-correcting code (and we have explicit examples of these). On the other hand, for any basis B, Jukna's upper bound technique from Theorem 17 shows that $R_2(L; B) = \mathcal{O}(n \ln n)$ for $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

So lower bounds for representing a random transformation $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$ are probably not even close to optimal bounds for computing L; are provably lower by nearly a $(\ln n)$ factor in some cases; and *never* give bounds of form $\omega(n \ln n)$. However, as mentioned in Section 1.2, the largest lower bounds on $s_3(L)$ for an explicit linear transformation L are of form $\Omega\left(n \cdot \lambda_3(n)\right) = \Omega\left(n \ln \ln n\right)$ [GHK+11], and for higher depths the bounds are weaker still. Thus we feel that the quantities $R_d(L; B)$ are still worth taking seriously as complexity measures. Thinking optimistically, we may ask:

Question 24. *Given $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$, suppose that $s_d(L) = \Omega(n \ln n)$. Does it follow that $R_d(L; B) = \Omega(n \ln n)$ for some basis B?*

This motivates another, more general question: how do we *find* a good basis B for L, one for which $R_d(L; B)$ is nearly maximized? We don't have an answer to this question. It should also be noted that the lower bound techniques of [Juk10b] which yield Theorem 20 are specific to the standard basis, so proving lower bounds for representing explicit linear transformations relative to other bases may well be harder.

However, in the present section we will show that, if we consider depth-3 circuits, there are at least some choices for B that definitely *fail* to yield interesting lower bounds. Namely, we will show that, if $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is invertible, then there exists a basis B such that $R_3(L; B) = \mathcal{O}(n)$. A uniformly-selected matrix $A \in \mathbb{F}_2^{n \times n}$ is invertible with $\Omega(1)$ probability [CRR90], so this phenomenon applies to many linear transformations. Compare this with Theorem 18, which tells us that for any *fixed* basis B, $R(L; B) = \Omega\left(n \ln n\right)$ for random operators L.

Theorem 25. *Let $D \subset \{0,1\}^n$ be of size n, and let $G : D \to \{0,1\}^n$ be any partial operator mapping D into the basis vectors $\{e_1, \ldots, e_n\}$. Then G is computable by a depth-3 circuit C with $\mathcal{O}(n)$ wires.*

If $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is an invertible linear transformation and we set $B := \{L^{-1}(e_1), \ldots, L^{-1}(e_n)\}$, then B is a basis, and it follows from Theorem 25 that $R_3(L; B) = \mathcal{O}(n)$. The circuits used to prove Theorem 25 involve non-linear gates. This is necessary in general: any linear circuit representing the linear transformation L relative to any basis also computes L, and most linear transformations require $\Theta\left(n^2 / \ln n\right)$ wires to compute by a linear circuit [Lup56, Bub86].

Proof of Theorem 25. Assume, by padding if needed, that n is a perfect square, $n = m^2$. Let H be the hash mapping and C_H the associated depth-1 circuit given by item 2 of Lemma 1, where in applying Lemma 1 we let D be the domain of G. We let Level 1 consist of m gates, and use a copy of C_H to connect Levels 0 and 1. Thus, on input $u \in D$, Level 1 computes $H(u)$.

Level 2 of C consists of $2m$ gates, call them $W = (w_1, \ldots, w_{2m})$. Each w_t is wired to every gate on Level 1. To define the behavior of these gates, first choose distinct sets $S_1, \ldots, S_n \subset [2m]$, each of size 2. We have $\binom{2m}{2} \geq n$, so we can do this (we used this idea earlier in the proof of Lemma 1, item 1). Let $v^i \in \{0,1\}^{2m}$ denote the characteristic vector of S_i.

Recall that $G(u)$ is a standard basis vector for any $u \in D$. Let $i(u)$ be defined by the equation

$$G(u) = e_{i(u)}. \tag{8}$$

Define the mapping $W : H(D) \to \{0,1\}^{2m}$ by the rule

$$W(H(u)) := v^{i(u)}. \tag{9}$$

This can be done consistently, since H is injective on D. We leave W undefined on other inputs. Recall that each Level 2 gate w_t is wired to see every Level 1 gate, and we have no restrictions on the functions used at gates, so we can indeed implement our choice of W.

Each output (Level 3) gate z_i (for $i \in [n]$) is connected to the two gates $g_t, g_{t'}$ whose indices satisfy $v^i_t = v^i_{t'} = 1$. We let z_i be the AND of g_t and $g_{t'}$.

Consider any input $u \in D$ to our circuit. By Eq. (9), the Level 2 gates collectively take on the value $v^{I(u)}$. Thus for $j \in [n]$, we have $z_i = 1$ iff $j = i(u)$. So, by Eq. (8) defining $i(u)$, our circuit computes G.

Finally we count the wires. There are $\mathcal{O}(n)$ wires between Levels 0 and 1, since C_H has $\mathcal{O}(n)$ wires. The number of wires between Levels 1 and 2 is $m \cdot (2m) = \mathcal{O}(n)$. Each output gate has two incoming wires, so there are $\mathcal{O}(n)$ wires in total. □

Acknowledgements

I am grateful to Stasys Jukna for many helpful comments.

References

[AKW90] Noga Alon, Mauricio Karchmer, and Avi Wigderson. Linear circuits over GF(2). *SIAM J. Comput.*, 19(6):1064–1067, 1990.

[AP94] Noga Alon and Pavel Pudlák. Superconcentrators of depths 2 and 3; odd levels help (rarely). *J. Comput. Syst. Sci.*, 48(1):194–202, 1994.

[Bub86] Siegfried Bublitz. Decomposition of graphs and monotone formula size of homogeneous functions. *Acta Informatica*, 23(6):689–696, 1986.

[CFL83] Ashok K. Chandra, Steven Fortune, and Richard J. Lipton. Lower bounds for constant depth circuits for prefix problems. In *10th ICALP*, pages 109–117, 1983.

[CFL85] Ashok K. Chandra, Steven Fortune, and Richard J. Lipton. Unbounded fan-in circuits and associative functions. *J. Comput. Syst. Sci.*, 30(2):222–234, 1985.

[Che08a] Dmitriy Yu. Cherukhin. Lower bounds for Boolean circuits with finite depth and arbitrary gates. *Electronic Colloquium on Computational Complexity (ECCC)*, TR08-032, 2008.

[Che08b] Dmitriy Yu. Cherukhin. Lower bounds for depth-2 and depth-3 Boolean circuits with arbitrary gates. In *3rd CSR*, pages 122–133, 2008.

[CRR90] Leonard S. Charlap, Howard D. Rees, and David P. Robbins. The asymptotic probability that a random biased matrix is invertible. *Discrete Mathematics*, 82(2):153–163, 1990.

[DDPW83] Danny Dolev, Cynthia Dwork, Nicholas Pippenger, and Avi Wigderson. Superconcentrators, generalizers and generalized connectors with limited depth (preliminary version). In *15th ACM STOC*, pages 42–51, 1983.

[GHK$^+$11] 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. *Electronic Colloquium on Computational Complexity (ECCC)*, TR11-150, 2011. To appear in STOC '12.

[JS10] Stasys Jukna and Georg Schnitger. Circuits with arbitrary gates for random operators. *CoRR*, abs/1004.5236, 2010.

[Juk10a] Stasys Jukna. Entropy of operators or why matrix multiplication is hard for depth-two circuits. *Theory Comput. Syst.*, 46(2):301–310, 2010.

[Juk10b] Stasys Jukna. Representing (0, 1)-matrices by Boolean circuits. *Discrete Mathematics*, 310(1):184–187, 2010.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers.* Algorithms and Combinatorics Series, #27. Springer-Verlag New York, LLC, 2012.

[Lok09] Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1-2):1–155, 2009.

[Lup56] O.B. Lupanov. On rectifier and switching-and-rectifier schemes. *Dokl. Akad. Nauk SSSR*, 111:1171–1174, 1956.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[Pos11] Alexey Pospelov. Faster polynomial multiplication via discrete Fourier transforms. In *6th CSR*, pages 91–104, 2011.

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

[Pud94] Pavel Pudlák. Communication in bounded depth circuits. *Combinatorica*, 14(2):203–216, 1994.

[RRV02] Ran Raz, Omer Reingold, and Salil P. Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *J. Comput. Syst. Sci.*, 65(1):97–128, 2002.

[RS03] Ran Raz and Amir Shpilka. Lower bounds for matrix product in bounded depth circuits with arbitrary gates. *SIAM J. Comput.*, 32(2):488–513, 2003.

[RTS00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. Discrete Math.*, 13(1):2–24, 2000.

[Sch77] A. Schönhage. Schnelle multiplikation von polynomen über körpern der charakteristic 2. *Acta Informatica*, 7:395–398, 1977.

[Val76] Leslie G. Valiant. Graph-theoretic properties in computational complexity. *Journal of Computer and System Sciences*, 13(3):278 – 285, 1976.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *6th MFCS*, pages 162–176, 1977.

[Vio09] Emanuele Viola. On the power of small-depth computation. *Foundations and Trends in Theoretical Computer Science*, 5(1):1–72, 2009.