Tight Bounds on Computing Error-Correcting Codes by Bounded-Depth Circuits with Arbitrary Gates*

Anna Gál† Kristoffer Arnsfelt Hansen‡ Michal Koucký§
Pavel Pudlák¶ Emanuele Violaǁ

February 21, 2013

Abstract

We bound the minimum number w of wires needed to compute any (asymptotically good) error-correcting code $C : \{0,1\}^{\Omega(n)} \to \{0,1\}^n$ with minimum distance $\Omega(n)$, using unbounded fan-in circuits of depth d with arbitrary gates. Our main results are:

(1) If $d = 2$ then $w = \Theta(n(\lg n/\lg\lg n)^2)$.

(2) If $d = 3$ then $w = \Theta(n \lg\lg n)$.

(3) If $d = 2k$ or $d = 2k+1$ for some integer $k \geq 2$ then $w = \Theta(n\lambda_k(n))$, where $\lambda_1(n) = \lceil \lg n \rceil$, $\lambda_{i+1}(n) = \lambda_i^*(n)$, and the $*$ operation gives how many times one has to iterate the function λ_i to reach a value at most 1 from the argument n.

(4) If $d = \lg^* n$ then $w = O(n)$.

For depth $d = 2$, our $\Omega(n(\lg n/\lg\lg n)^2)$ lower bound gives the largest known lower bound for computing any linear map. The upper bounds imply that a (necessarily

*A preliminary version of this paper appeared in: A. Gál, K. A. Hansen, M. Koucký, P. Pudlák, E. Viola. Tight bounds on computing error-correcting codes by bounded-depth circuits with arbitrary gates. In *Proceedings of the 44rd Annual ACM Symposium on Theory of Computing (STOC)*, 2012, pp. 479–494.

†University of Texas at Austin, email: panni@cs.utexas.edu. Supported in part by NSF Grant CCF-1018060.

‡Aarhus University, email: arnsfelt@cs.au.dk. Supported by the Sino-Danish Center for the Theory of Interactive Computation, funded by the Danish National Research Foundation and the National Science Foundation of China (under the grant 61061130540).

§Institute of Mathematics, Academy of Sciences, Prague, email: koucky@math.cas.cz. Part of the work done while visiting the University of Toronto, partially supported by NSERC, and Aarhus University, partially supported by the Sino-Danish Center CTIC (funded under the grant 61061130540). Partially supported by GA ČR P202/10/0854, grant IAA100190902 of GA AV ČR, project No. 1M0021620808 of MŠMT ČR and RVO: 67985840.

¶Institute of Mathematics, Academy of Sciences, Prague, email: pudlak@math.cas.cz. Partially supported by the grant IAA100190902 of GA AV ČR, by the Center of Excellence CE-ITI under the grant P202/12/G061 of GA ČR and RVO: 67985840.

ǁNortheastern University, email: viola@ccs.neu.edu. Supported by NSF grant CCF-0845003.

dense) generator matrix for our code can be written as the product of two sparse matrices.

Using known techniques we also obtain similar (but not tight) bounds for computing pairwise-independent hash functions.

Our lower bounds are based on a superconcentrator-like condition that the graphs of circuits computing good codes must satisfy. This condition is provably intermediate between superconcentrators and their weakenings considered before.

Keywords: Error-correcting codes, hashing, bounded-depth circuits, superconcentrators, lower bounds.

1 Introduction

Error-correcting codes are fundamental objects with a myriad of applications, for example in information theory, cryptography, and combinatorial constructions. Of particular importance are codes over the binary alphabet $\{0, 1\}$ that are *asymptotically good*: have constant rate (encode m bits into $n = O(m)$ bits) and can correct a constant fraction of errors. We refer to them simply as *good* codes.

In this paper we study the complexity of *encoding* good codes. Although the complexity of *de*coding is sometimes the bottleneck in applications, other times it is the complexity of encoding that matters the most. For example, jumping ahead, "efficient" encoding translates to efficient hashing thanks to a recent result by Ishai, Kushilevitz, Ostrovsky, and Sahai [17].

The complexity of encoding good codes has been studied before. It was shown that some popular encoding methods (e.g. concatenated convolutional codes, or repeat-convolute codes) cannot yield good codes under some assumptions on the complexity of the encoder [4, 5]. For example, Bazzi and Mitter [5] prove that if the encoder can be represented as a binary branching program (or equivalently a random access machine) that uses linear time and sublinear space, then the code computed cannot be good. This result should be contrasted with the existence of good codes that can be encoded in linear time [14]. Explicit constructions of good codes encodable (and decodable) in linear time and linear space were given by Spielman [29, 30]. It follows immediately from bounds on the noise-sensitivity of small AC^0 circuits [20, 6] that small AC^0 circuits cannot compute good codes. This result was generalized in [32, 21] (cf. [5]).

In this paper we consider unbounded fan-in circuits with *arbitrary* gates. In particular, we allow the use of parity gates. Giving lower bounds for arbitrary gates, that is, regardless of the operations computed by the individual gates, makes our lower bounds stronger. Our upper bounds hold for circuits that consist of parity gates only. Our focus is the tradeoffs between the number of wires and the depth of the circuits.

Previously, two main tradeoffs were known. Any code can be computed in depth 1 with $O(n^2)$ wires (since we allow arbitrary gates), and it is also easy to show that $\Omega(n^2)$ wires are necessary to compute any good code in depth 1. We can generalize the upper bound to $O(n^{1+1/d})$ wires and depth d, for any fixed d, by viewing the message as a d-dimensional cube and encoding one by one along each dimension. On the other hand, Spielman [30] gives

explicit good codes that can be encoded by bounded fan-in circuits with $O(n)$ wires and depth $O(\lg n)$ (and also decoded in linear time). Thus, optimal number of wires for encoding good codes can be achieved using depth $O(\lg n)$. It is easy to see that with bounded fan-in circuits, linear number of wires for encoding good codes cannot be achieved in smaller than $\Omega(\lg n)$ depth. Can we achieve closer to optimal bounds on the number of wires in small depth with unbounded fan-in circuits?

1.1 Our results

We show that already depth 2 is sufficient to have a quasi-linear $O(n(\lg n/\lg\lg n)^2)$ number of wires for computing asymptotically good codes. In fact, for any depth we obtain matching upper and lower bounds that reveal that the optimal number of wires is n times an inverse Ackermann-like function $\lambda(n)$ that is parameterized by the depth.

Notation. Let $\lambda_1(n) = \lceil \lg n \rceil$ and $\lambda_{i+1}(n) = \lambda_i^*(n)$, where the $*$ operation gives how many times one has to iterate the function λ_i to reach a value at most 1 from the argument n.

We say that an injective function $C : \{0,1\}^m \to \{0,1\}^n$ is a (ρ, δ)-*good code* if $m \geq \rho n$ and any two distinct codewords have hamming distance at least δn. It is well known that there are constants $0 < \rho_0, \delta_0 < 1$ so that (ρ_0, δ_0)-good codes exist for all m. Indeed one may pick $1/32 \leq \rho_0$ and $1/8 \leq \delta_0$.

Let $w_d(C)$ be the wire complexity of encoding the code C by depth-d unbounded-fan-in circuits with arbitrary gates, that is the minimum number of wires over all circuits of depth d computing C. Let $w_{d,\rho,\delta}(n) = \min_C w_d(C)$, over all binary (ρ, δ)-good codes C with block-length n (and thus message length $m = \rho n$, and relative distance δn). An XOR circuit is a circuit consisting of input gates and XOR gates (i.e., gates that compute parity of their inputs).

Our results can be summarized in the following main theorem.

Theorem 1 (Complexity of encoding) *For any constants $0 < \rho \leq 1/32$ and $0 < \delta < 1/8$, we have*

$w_{2,\rho,\delta}(n) = \Theta(n(\lg n/\lg\lg n)^2)$,

$w_{3,\rho,\delta}(n) = \Theta(n \lg\lg n)$,

if $d = 2k$ or $d = 2k+1$ for some integer $k \geq 2$ then $w_{d,\rho,\delta}(n) = \Theta(n\lambda_k(n))$.

The upper bounds are witnessed by (non-explicit) circuits using only XOR gates (hence computing good codes that are linear). The lower bounds hold for any $\rho, \delta > 0$, for circuits with arbitrary gates, and computing arbitrary (possibly nonlinear) (ρ, δ)-good codes.

Each bound is obtained for the first time in this paper. For depth 2, our $\Omega(n(\lg n/\lg\lg n)^2)$ lower bound gives the largest known lower bound for computing any linear map, improving on the $\Omega(n \lg^{3/2} n)$ bound by Pudlák and Rödl in [27]. Moreover their lower bound only held against circuits consisting exclusively of XOR gates, as opposed to arbitrary gates in our result. For an excellent survey of lower bounds for circuits with arbitrary gates, see the recent book [18] by Jukna. As we discuss later, our lower bounds also apply to multiplication by

an n-bit integer, by an n-bit field element, by an $n \times n$ Toeplitz matrix, and to computation of hash functions.

The lower bounds in Theorem 1 show that to compute good codes with a linear number of wires, the depth cannot be constant. However, the depth can grow very slowly as the following theorem indicates.

Theorem 2 *For any constants* $0 < \rho \leq 1/32$, $0 < \delta < 1/8$ *and* $d \geq 1$, *we have* $w_{\lambda_d(n),\rho,\delta}(n) = O(n)$.

Spielman [29] achieves depth $O(\lg n)$ for linear-size circuits with *bounded fan-in* gates, and this is optimal for computing good codes with bounded fan-in circuits. Our theorem holds for circuits with unbounded fan-in gates.

Decomposing the generator matrix of good codes into sparse matrices. Since the upper bounds in Theorem 1 hold for circuits with XOR gates only, they correspond to linear transformations. So those upper bounds can be interpreted as saying that there exist $n \times \Omega(n)$ matrices G generating good codes over GF(2), that can be decomposed as $G = G_1 \cdot G_2 \cdot \cdots \cdot G_d$, where the G_i are *sparse* (have quasilinear number of ones). By contrast, note it is easy to see that any generator matrix G itself must have $\Omega(n^2)$ ones, if it generates a good code. Our lower bounds show that no such decomposition exists with $d = O(1)$ and each matrix G_i having $O(n)$ ones.

Coding vs. hashing. Our results on the complexity of encoding imply new upper and lower bounds on the complexity of computing pairwise independent hash functions. This follows from the close relationships between the complexity of encoding and that of hashing, the more surprising direction of which (i.e., codes imply hash functions) we extrapolate from an exciting result by Ishai, Kushilevitz, Ostrovsky, and Sahai [17]. The relationships show that depth-d circuits for codes imply depth-$2d$ circuits for hashing, while depth-d circuits for hashing imply depth-d circuits for codes; and the wire complexity only multiplies by a constant. Using this in conjunction with Theorem 1 immediately gives a superlinear lower bound on the wire-complexity of constant-depth hashing circuits, as well as upper bounds.

It is instructive to compare our results to those of the aforementioned paper [17]. [17] obtain circuits with $w = O(m)$ wires (and unspecified depth $d \geq \Omega(\lg m)$) for computing pairwise independent hash functions mapping m bits to m bits. This disproves a conjecture by Mansour, Nisan, and Tiwari [22] that $\Omega(m \lg m)$ wires were needed.

Our results show that one can go below $w = \Omega(m \lg m)$ wires (e.g. $w = O(m \lg^* m)$) even with constant-depth circuits. However, to get $w = O(m)$ super-constant depth is necessary.

1.2 Techniques

Lower bounds. Slowly-growing bounds similar to those in our Theorem 1 are known to hold for the number of edges in *superconcentrator* graphs. Indeed, to establish our lower

bounds we establish and exploit a relationship between superconcentrators and circuits computing good codes. However, despite this relationship, circuits computing good codes behave differently from superconcentrator graphs. In fact, jumping ahead, our bounds for depth 2 imply a gap between the wire complexity of circuits computing good codes and the wire complexity of superconcentrator graphs.

We now elaborate on the connection with superconcentrators. We state the following properties for circuits with m input nodes and n output nodes. There are various connectivity requirements one could ask for in a circuit. Consider a circuit with m inputs and $n \geq m$ outputs. Let X be a subset of input bits and Y be a subset of output bits of the circuit. Denote by $f(X, Y)$ the maximum number of vertex disjoint paths from X to Y.

For a fixed constant $0 < \delta \leq 1$ one can require:

1. For each $k \in \{1, \ldots, m\}$, for each X and Y of size k, $f(X, Y) \geq \delta k$.
2. For each $k \in \{1, \ldots, m\}$, for each X and a random Y of size k, $\mathrm{E}_Y[f(X, Y)] \geq \delta k$.
3. For each $k \in \{1, \ldots, m\}$, for random X and Y of size k, $\mathrm{E}_{X,Y}[f(X, Y)] \geq \delta k$.

These are various versions of the *superconcentrator* property ordered by strength. The classical definition of superconcentrators [31] requires Property 1 to hold for $\delta = 1$ (and $n = m$). Property 3 corresponds to the so-called relaxed superconcentrators which were analyzed in [11, 26].

In this work we show that Property 2 is the property satisfied by circuits computing good codes. A connection between superconcentrators and circuits computing error-correcting codes was already observed by Spielman [30]. His construction of linear-size encoding circuits was inspired by known constructions of linear-size superconcetrators. Spielman also observed that some similarity to superconcentrators is necessary. He proved that circuits (with m inputs and n outputs) computing codes with minimum distance δn have δm vertex disjoint paths from any set of δm inputs to any set of $(1 - \delta)n$ outputs.

We revisit the latter connection between circuits encoding error-correcting codes and superconcentrators, and establish the following lemma.

Lemma 3 *Let $\delta > 0$ be a constant, $C : \{0,1\}^m \to \{0,1\}^n$ be a code with minimum distance δn and G be a circuit computing C. For any $0 < k \leq m$, and for any k-element subset X of inputs of G, if we take uniformly at random a k-element subset Y of outputs of G, then the expected number of vertex disjoint paths from X to Y in G is at least δk.*

We prove this lemma using a classical result in matroid theory. We give a self-contained proof of the necessary claim in the appendix.

Having established the above lemma, we proceed to prove lower bounds on the number of wires of circuits satisfying Property 2. The quality of the bounds we get depends on the rate $\rho = m/n$. For simplicity, we state our bounds for codes with constant rate ρ, that is our circuits have $m = \Omega(n)$ input nodes. The precise dependence of the bounds on $\rho = m/n$ is hidden in the Ω notation.

For the case of depth 2, we show that a depth-2 circuit satisfying Property 2 has at least $\Omega(n(\lg n/\lg\lg n)^2)$ wires. Moreover, this is optimal due to our constructions.

We remark that the bounds for depth 2 circuits satisfying Property 1, 2 and 3 are different. Specifically, by results of Radhakrishnan and Ta-Shma [28], for any fixed $0 < \delta \leq 1$, a depth-2 circuit satisfying Property 1 has at least $\Omega(n \lg^2 n/\lg\lg n)$ wires which is known to be optimal. ([28] state the claim only for $\delta = 1$ but their proof works for any constant $\delta > 0$.) For Property 3, [11, 26] show that a depth-2 circuit satisfying it has at least $\Omega(n \lg n)$ wires, which is again known to be optimal. An example of a circuit satisfying this property is a circuit computing the Prefix-XOR function [8, 9].

The lower bound for depth 2. Our $\Omega(n(\lg n/\lg\lg n)^2)$ lower bound for depth-2 is obtained by modifying the clever lower bound by Radhakrishnan and Ta-Shma [28], which was tailored for Property 1 and gives a stronger bound. As in [28], we classify the vertices in the intermediate level according to their degree. We consider $\Omega(\lg n/\lg\lg n)$ disjoint classes and prove a lower bound $\Omega(n \lg n/\lg\lg n)$ on the number of edges incident with these vertices for each of them, unless the number of edges non-incident with these vertices is already $\epsilon n(\lg n/\lg\lg n)^2$ for some absolute constant $\epsilon > 0$.

Each of the classes on the intermediate level is associated with a number k and we use the condition of Lemma 3 for this k to prove the lower bound on the number of incident edges. The class associated with k is denoted by V_m^k. The assumption that the number of edges non-incident with the vertices of the given class V_m^k is less than $\epsilon n(\lg n/\lg\lg n)^2$ implies that for every set of k input vertices and randomly chosen set of k output vertices, there are on average $\delta' k$ vertex-disjoint paths connecting them through V_m^k, for some positive constant δ'.

To prove the lower bound on the number of edges incident with V_m^k, we argue by contradiction. We assume that the number of these edges is small, and find a large subset of output vertices W^0 and a subset of input vertices U^0, $|U^0| \geq k$ such that there is no path going through V_m^k that connects U^0 with W^0. We show that with our setting of the parameters this is a contradiction.

To construct the sets U^0 and W^0 we use a technique based on the work of Hansel [16]. For every vertex $v \in V_m^k$, we randomly either delete all edges connecting v with output vertices or all edges connecting v with input vertices. We do the former with small probability, the latter with high. We put U^0 and W^0 to be the set of input (respectively, output) vertices that are not affected by this process. Since all paths through V_m^k have been disconnected, but no path has been removed between U^0 and W^0, the sets U^0 and W^0 are not connected through V_m^k in the original graph. This concludes the overview of the proof.

One novelty in our proof is that we use different probabilities for the two possible choices to obtain W^0 of linear size. In fact, this reflects a qualitative difference between superconcentrators and circuits computing good codes. Optimal depth-2 superconcentrators must have the same average degree for input and output nodes, while our optimal construction of circuits computing good codes has different average degrees. More detailed discussion of the difference between our proof and the proof of [28] is provided after the proof of Theorem 8, and Section 4.1 discusses the properties of depth-2 superconcentrators.

Lower bounds for depth > 2. For depth > 2, our lower bounds follow from combining our Lemma 3 with off-the-shelf lower bounds on superconcentrators. Specifically, we use bounds by Pudlák [26] that improve on those by Dolev et al. [11]. We note that these lower bounds are for circuits satisfying (essentially) Property 3 above. Hence, for depth at least 3, Property 3 is sufficient to obtain a result tight up to constant factors.

Upper bounds. Our constructions of circuits computing good codes are probabilistic, and we leave it as an open problem to obtain explicit constructions. (Jumping ahead, partial progress towards explicit constructions is discussed in Section 6.)

Basic building blocks of our constructions are circuits that we call *range detectors*. A (m, n, ℓ, k, r, s)-range detector is a circuit built from XOR gates that has m inputs, n outputs and on any input of Hamming weight between ℓ and k it outputs a string with Hamming weight between r and s. We will omit the last parameter if $s = n$. Clearly, an $(m, O(m), 1, m, \Omega(m))$-range detector is a circuit computing a good linear code as for linear codes one has to worry only about the number of ones produced on non-zero inputs. When $\ell \in \Theta(k)$ and $n \in \Theta\left(\lg \binom{m}{k}\right)$ then one can easily construct range detectors of depth one with $r \in \Theta(n)$ using $O(m \lg m)$ wires, by an application of the probabilistic method. If we take such range detectors for $\ell = m/2^i$ and $k = m/2^{i-1}$, for $i = 1, \ldots, \lg m$, in parallel then on any non-zero input at least one of them will output a constant fraction of ones (and say also a constant fraction of zeros). Hence, an XOR gate sampling one output from each of these range detectors will evaluate to one on non-zero input with constant probability. If we take $O(m)$ of such XOR gates chosen independently at random, then on any non-zero input a constant fraction of them will evaluate to one with overwhelming probability. Hence, a particular choice of these XOR gates will have such a property on any non-zero input. That in essence gives a depth-2 XOR circuit with $O(m \lg^2 m)$ wires computing good codes. By modifying the parameters slightly one can achieve $O(m \lg^2 m / \lg^2 \lg m)$-size depth-2 XOR circuits for good codes.

To obtain asymptotically better size in higher depth we use a similar recursive approach. We present the main idea of the construction here. First, one can *condense* the input using various depth-1 range detectors and then one can apply on the output of each of the range detectors a depth-d circuit computing a good code. Adding a layer of XOR gates that sample outputs of these depth-d range detectors concludes the construction of a depth-$d+2$ circuit for good codes. A careful choice of parameters leads to the overall reduction in size; the actual construction is slightly more involved.

This recursive approach is similar to the construction of superconcentrators [11]. However, there is a notable difference between the construction for codes and for superconcentrators. In known superconcentrator constructions, the bottom and top layers are symmetric so the whole circuit can be made symmetric with respect to the middle layer. In our construction, the bottom and top layers are different and although on intuitive level they fulfill symmetric functionality they cannot be interchanged with each other in general. Indeed, as observed before, for depth 2, the size bounds on codes are different than the size bounds on superconcentrators.

Bounds on hashing. Our results on the complexity of encoding imply corresponding results on the complexity of computing pairwise independent hash functions $f : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^n$, i.e. functions such that $\forall x \neq y$, the joint distribution $(f(x,S), f(y,S))$ is uniform over $(\{0,1\}^n)^2$.

This implication relies on the fact that the complexity of hashing and coding are closely related. First, an application of the Chernoff bound shows that any pairwise independent hash function contains a good code, which was explicitly pointed out by Miltersen. This implication has no over-head. Hence our lower bounds for codes apply to hashing as well.

Fact 4 (Proposition 7 in [23]) *Let*

$$f : \{0,1\}^{\rho n} \times \{0,1\}^r \to \{0,1\}^n$$

be a pairwise independent hash function. For every $\delta < 1/2$ such that $2\rho < 1 - H(\delta)$, there is a fixed R such that the map $f(\cdot, R)$ is the encoding function of an error correcting code with relative distance $\geq \delta$ (and message length ρn and block length n).

For the reverse direction, we rely on a recent result by Ishai, Kushilevitz, Ostrovsky, and Sahai [17], who give a construction of pairwise independent hash functions using circuits with a linear number of wires. We observe that their construction can be seen as a reduction to encoding, and that the reduction blows up the wires and the depth by only a constant factor. So plugging our efficient encoding circuits we get correspondingly efficient hashing circuits.

Theorem 5 (Implicit in [17]) *Suppose there are constants ρ, δ, d, and an increasing function $w(n) \geq n$ such that*

1. *for any n there is a linear function $C : \{0,1\}^{\rho n} \to \{0,1\}^n$ that is a (ρ, δ)-good code, and that can be computed by a depth-d XOR circuit with $w(n)$ wires.*

Then there is a constant c such that

2. *for every m we can compute a pairwise independent hash function $h : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^m$ with $r \leq cm$, by a depth-$2d$ circuit with $\leq cw(cm)$ wires (using arbitrary gates).*

Moreover, if the codes are explicit then the hash functions are too.

There is a factor-2 gap between the upper and lower bounds for our hashing circuits. Resolving this is an interesting problem. It is also not hard to show that hashing circuits satisfy the superconcentrator Property 1 mentioned earlier, and so, as also explained earlier, depth-2 hashing circuits are larger than depth-2 encoding circuits.

Organization. In Section 2 we prove that circuits computing good codes must satisfy connectivity Property 2. In Section 3 we prove the lower bounds in Theorem 1. In Section 4 we prove the upper bounds in Theorem 1 and also Theorem 2. The constructions of hash functions are in Section 5. In this section we also deduce lower bounds for various types of multiplication. Finally, we make some remarks on making our upper bounds explicit in Section 6.

2 Connectivity of circuits computing good codes

In this section we provide a proof of Lemma 3 which establishes that circuits computing good codes satisfy connectivity Property 2. We restate that lemma next.

Lemma 3 *Let $\delta > 0$ be a constant, $C : \{0,1\}^m \to \{0,1\}^n$ be a code with minimum distance δn and G be a circuit computing C. For any $0 < k \leq m$, and for any k-element subset X of inputs of G, if we take uniformly at random a k-element subset Y of outputs of G, then the expected number of vertex disjoint paths from X to Y in G is at least δk.*

To prove the lemma we need the following definition and lemma. Let X be the set of inputs and let W be the set of outputs of a directed graph. Given a subset of outputs $Y \subseteq W$, we say that a vertex $v \in W \setminus Y$ is *bad* for Y if the largest number of vertex disjoint paths from X to $Y \cup \{v\}$ is not larger than the largest number of vertex disjoint paths from X to Y.

Lemma 6 *Let $V \subseteq W \setminus Y$ be a set of bad vertices for Y. Then the largest number of vertex disjoint paths from X to $Y \cup V$ is not larger than the largest number of vertex disjoint paths from X to Y.*

First note that this statement would easily follow, if *any* collection of vertex disjoint paths could be extended to a collection of vertex disjoint paths with the largest possible cardinality in any graph. Such a property would be similar to having a matroid, where the sets of vertex disjoint paths form the independent sets. However this property does not hold, and one cannot define a matroid with vertex disjoint paths forming the independent sets. Consider the simple example of a graph that consists of a matching between vertices $x_1, \ldots x_k$ and $y_1, \ldots, y_k$, with edges from x_i to y_i, and an extra edge from x_1 to y_k. Then the largest number of vertex disjoint paths is k, but there is no collection of k vertex disjoint paths containing the path from x_1 to y_k. This example also illustrates why some seemingly simple approaches to prove Lemma 6 fail.

Despite the fact that the collections of vertex disjoint paths do not form a matroid, the collections of their endpoints do. The statement of Lemma 6 is equivalent to the following statement: the subsets of W formed by the sets of endpoints of vertex disjoint paths from X to W are independent sets of a matroid over W. This is a well known theorem of matroid theory, see e.g. Chapter 13 in [33], Chapter 2.4 in [24], or [25]. We include a self contained and direct proof of Lemma 6 in the Appendix.

In the special case of linear codes a version of Lemma 6 with a different definition of "bad" vertices would be substantially easier to prove. One could consider the rank of the submatrices $X \times Y$ of the generator matrix describing the code. The bad vertices are those outputs v that do not increase the rank, that is the rank of the submatrix $X \times Y \cup \{v\}$ is the same as the rank of the submatrix $X \times Y$. Then, adding the union of all bad vertices cannot increase the rank either: the bad vertices correspond to columns of the generator matrix that are in the linear span of the submatrix $X \times Y$. The rank of the submatrix $X \times Y$ of the generator matrix describing the code is at most the number of nodes in the smallest set S of vertices in the graph of the circuit such that every path from X to Y contains at least one vertex of the set S, since varying the inputs in X the number of different outputs over Y cannot be larger than $2^{|S|}$. The statement then follows by Menger's Theorem.

We use the following version of Menger's Theorem (see e.g. Theorem 4.2 in [13]).

Theorem 7 (Menger's Theorem) *Let G be a directed graph, X its set of input (indegree 0) nodes, and Y its set of output (outdegree 0) nodes. The largest number of pairwise vertex disjoint paths from X to Y equals the smallest number of vertices in a set S such that every path from X to Y contains a vertex from S.*

We refer to a set of vertices S such that every path from X to Y contains a vertex from S as a *(vertex) cut* separating X and Y.

We are ready to prove Lemma 3.

Proof of Lemma 3. For a fixed $k \in \{1, \ldots, m\}$ and given k-element set X of inputs, we prove that the expected number of vertex disjoint paths from X to a randomly chosen k-element subset Y of outputs is at least δk. We will pick Y one element at a time by choosing from the remaining outputs at random.

We claim that as long as $|Y| < k$ (so that there are less than k vertex disjoint paths from X to Y) with probability at least δ the next randomly chosen element of Y will increase the number of vertex disjoint paths from X to the current Y by one.

As above, call an output vertex v bad for Y, if adding it to Y does not increase the number of vertex disjoint paths from X to Y. We claim that for any Y with $|Y| < k$, at least δn output vertices are not bad. This suffices to prove the claim as with probability at least δ we will sample such a vertex at each step and then by linearity of expectation we get at least δk vertex disjoint paths on average.

Let $|Y| < k$. Denote the set of output vertices bad for Y by B. Let the number of vertex disjoint paths from X to the current Y be $\ell < k$. By Lemma 6, if we take the union of Y with all the bad vertices then there are still no more than ℓ vertex disjoint paths from X to the union $Y \cup B$. By Menger's theorem (Theorem 7), the smallest cut separating X and $Y \cup B$, is of size ℓ. If we set all input bits except for X to 0, then by varying inputs to X we have $2^{|X|} = 2^k$ different inputs. However, over these 2^k inputs, since the cut separating X and $Y \cup B$ is of size ℓ, outputs belonging to $Y \cup B$ will see at most 2^ℓ different settings as these output bits will be determined by the values at the gates of the cut. Thus there exist two different inputs of the form $0x_1$ and $0x_2$ (they are both 0 outside X, but differ on X), such that the outputs of our circuit on these two inputs agree on the $Y \cup B$ part. So the

Hamming distance between the outputs of the circuit on $0x_1$ and $0x_2$ is at most the number of output vertices outside of $Y \cup B$.

However, since G computes a code with minimum distance δn, the Hamming distance between the encodings of any two different inputs has to be at least δn. Thus, the number of output vertices outside of $Y \cup B$ is at least δn. □

3 Lower bounds on computing good codes

In this section we prove lower bounds on the number of wires of circuits computing good codes. We start with the case of depth 2, then we discuss depth bigger than 2.

3.1 Lower bound for depth two

In this section we prove our lower bound for depth-2 circuits computing good codes:

Theorem 8 $w_{\rho,\delta,2}(n) \in \Omega\left(n\left(\frac{\lg n}{\lg\lg n}\right)^2\right)$.

We need the following lemma, an easy corollary of Chebyshev's Inequality. Its proof is included in Appendix for completeness.

Lemma 9 *Let $X_1, \ldots, X_k$ be 0-1 random variables and $C, \alpha > 0$ be reals. Suppose that for every $i \in \{1, \ldots, k\}$, there are at most C indices $j \in \{1, \ldots, k\}$ such that X_i and X_j are not independent. Let $\mu = \mathrm{E}\left(\sum_{i=1}^k X_i\right)$. Then*

$$\Pr\left[\left|\sum_{i=1}^k X_i - \mu\right| \geq \alpha\mu\right] \leq \frac{C}{\alpha^2\mu}.$$

Proof of Theorem 8. Fix a large enough n and consider the directed graph G that corresponds to a circuit computing a (ρ, δ)-good code on inputs of size $m = \rho n$. Let U be the set of vertices of G corresponding to the inputs of the circuit, V be the set of vertices of G that correspond to the middle layer of the circuit, and W be the vertices corresponding to the output gates of the circuit. Hence, $|U| = m$ and $|W| = n$. All the edges in G go either from U to V or from V to W.

Our goal is to show that for n sufficiently large, the number of edges of G is at least

$$\frac{\delta \cdot \min(\rho, \delta)}{1600} \cdot n \cdot \left(\frac{\lg n}{\lg\lg n}\right)^2. \tag{1}$$

We will prove this by contradiction so we will assume that G has fewer edges than (1).

By the degree $\deg(v)$ of a vertex v in G we understand the number of incident edges (incoming and outgoing) in G, and for a set of vertices A of G, $\deg_A(v)$ denotes the number of edges between A and v.

For every integer k such that $n^{1/4} \leq k \leq n^{1/2}$ we define

$$V_h^k = \{v \in V;\ \deg(v) \geq \frac{n}{k}\lg^2 n\} \qquad \text{(high degree vertices)}$$

$$V_m^k = \{v \in V;\ \frac{n}{k}\lg^2 n > \deg(v) \geq \frac{n}{k\lg^2 n}\} \qquad \text{(medium degree vertices)}$$

$$V_l^k = \{v \in V;\ \frac{n}{k\lg^2 n} > \deg(v)\} \qquad \text{(low degree vertices)}$$

If for all k of the form $(\lg n)^{4i}$, where i is an integer between $\frac{1}{16}\cdot\frac{\lg n}{\lg\lg n}$ and $\frac{1}{8}\cdot\frac{\lg n}{\lg\lg n}$, V_m^k is incident with at least

$$\frac{\delta\cdot\min(\rho,\delta)}{100}\cdot n\cdot\frac{\lg n}{\lg\lg n} \tag{2}$$

edges, then we immediately obtain a contradiction with the number of edges in G. So for the rest of the proof we fix some k such that $n^{1/4} \leq k \leq n^{1/2}$ where the number of edges incident with V_m^k is less than (2). Since k is fixed, we will omit the upper index k in V_h^k, V_m^k and V_l^k.

Given the bound (2) on the number of edges incident with V_m, for at most $\frac{\delta}{10}n$ vertices w from W, $\deg_{V_m}(w) > \frac{\delta}{10}\cdot\frac{\lg n}{\lg\lg n}$. Call these vertices W_{bad}, and set

$$W_{\text{good}} = W \setminus W_{\text{bad}}.$$

Furthermore, for at most $\frac{\rho}{5}n = \frac{m}{5}$ vertices u from U,

$$\deg_{V_m}(u) > \frac{\delta}{10}\cdot\frac{\lg n}{\lg\lg n},\ \text{or}\ \deg(u) > \frac{\delta}{160}\cdot\left(\frac{\lg n}{\lg\lg n}\right)^2.$$

Call these vertices U_{bad}, and set $U_{\text{good}} = U \setminus U_{\text{bad}}$.

Let $p = 1/\lg n$. Consider the following random process: for each vertex $v \in V_m$, with probability p remove all the edges between v and W and with the remaining probability $1-p$ remove all the edges between v and U. Let

$W^0 = \{w \in W_{\text{good}};\ \text{no edge from } w \text{ to } V_m \text{ was removed by the random process}\}$,

$U^0 = \{u \in U_{\text{good}};\ \text{no edge from } u \text{ to } V_m \text{ was removed by the random process}\}$.

Notice, the random process cuts all paths from U_{good} to W_{good} going through V_m. Since vertices U^0 and W^0 retain all their edges during the process, there is no path from U^0 to W^0 going through V_m in the original graph G.

Claim 10 *For large enough n, with probability at least $1/3$, the following conditions are true simultaneously:*

$$\begin{aligned} |W^0| &\geq (1-\frac{\delta}{5})n, \\ |U^0| &\geq \frac{2\rho}{5}m^{9/10}. \end{aligned}$$

In particular, $|U^0| \geq k$ if n is sufficiently large.

Proof of the claim. First we show that the first condition is satisfied with probability at least 2/3. Each vertex $w \in W_{\text{good}}$ is originally connected to at most $\frac{\delta}{10} \cdot \frac{\lg n}{\lg\lg n}$ vertices in V_m so the probability of w being in W^0 is

$$\begin{aligned}(1-p)^{\deg_{V_m}(w)} &\geq (1 - p\deg_{V_m}(w)) \\ &\geq \left(1 - p\frac{\delta}{10} \cdot \frac{\lg n}{\lg\lg n}\right) \\ &\geq \left(1 - \frac{\delta}{10\lg\lg n}\right).\end{aligned}$$

Consequently, the expected number of vertices $w \in W_{\text{good}}$ that are not in W^0 is at most $\frac{\delta}{10\lg\lg n}|W_{\text{good}}|$. Hence by the Markov Inequality, with probability at least 2/3, $|W^0| \geq |W_{\text{good}}| - \frac{3\delta}{10\lg\lg n}|W_{\text{good}}|$. Since $|W_{\text{good}}| \geq (1 - \frac{\delta}{10})n$, for n large enough, $|W^0| \geq (1 - \frac{\delta}{5})n$.

Now it suffices to show that the second condition is satisfied with probability at least 2/3. First, we estimate the expected size of U^0. Each vertex $u \in U_{\text{good}}$ will end up in U^0 with probability

$$p^{\deg_{V_m}(u)} \geq \left(\frac{1}{\lg n}\right)^{\frac{\delta}{10} \cdot \frac{\lg n}{\lg\lg n}} = \frac{1}{n^{\frac{\delta}{10}}}$$

Thus, the expected size of U^0 is at least $\frac{4}{5} \cdot m \cdot n^{-\frac{\delta}{10}} \geq \frac{4\rho}{5} \cdot m^{9/10}$. We want to use Lemma 9 to show that the size of U^0 is close to its expectation with high probability.

For two distinct $u, u' \in U_{\text{good}}$, the two events whether u and u' fall in U^0 are independent if u and u' do not share a neighbor in V_m. Hence for a given u, there are at most $\deg_{V_m}(u) \cdot \frac{n}{k}\lg^2 n \leq \frac{\delta}{10} \cdot \frac{\lg n}{\lg\lg n} \cdot \frac{n}{k}\lg^2 n \leq \frac{\delta}{10} \cdot \frac{\lg^3 n}{\lg\lg n} n^{3/4}$ distinct $u' \in U_{\text{good}}$ for which the events may be dependent. In Lemma 9, set $C = m^{4/5}$, $\alpha = 1/2$, and for $u \in U_{\text{good}}$, set X_u to be the indicator variable whether $u \in U^0$. Since $\mathrm{E}[|U^0|] \geq \frac{4\rho}{5} \cdot m^{9/10}$, by Lemma 9, the probability that $|U^0| \leq \frac{1}{2} \cdot \frac{4\rho}{5} \cdot m^{9/10}$ goes to zero as n grows. So for n large enough this event happens with probability less than 1/3. □

For the rest of the proof we fix set U^0 and W^0 satisfying the bounds from the previous claim and we consider the original graph G. We know that there is no path from U^0 to W^0 going through V_m in G.

Our goal is to derive a contradiction from the fact that the number of edges incident with V_m is small, and that there is a small number of edges in the graph overall. We will consider the following quantities to derive the contradiction. Let $X \subseteq U^0$ and $Y \subseteq W$. Let $f^d(X, Y)$ denote the number of vertex disjoint paths connecting X and Y via vertices of V_l, and $f^a(X, Y)$ denote the number of all paths connecting X and Y via vertices of V_l. We will derive a contradiction by proving that $f^d(X, Y) > f^a(X, Y)$ for some X and Y.

Claim 11 *Let X be a random k-element subset of U^0 and let Y be a random k-element subset of W. Then the expected value of $f^d(X, Y)$ is at least $\frac{3}{4}\delta k$.*

Proof of the claim. Fix an arbitrary k-element subset X of U^0 and pick a random k-element subset of W. By Lemma 3, the expected number of vertex disjoint paths from X to Y is at least δk, so in expectation there are at least δk vertex disjoint paths connecting X and Y. Some of the paths may go via vertices in V_m or V_h. However, an $x \in X$ and a $y \in Y$ may be connected via V_m only if $y \notin W^0$, and at most $|V_h|$ vertex disjoint paths can go through V_h. The expected size of $Y \setminus W^0$ is at most $\delta k/5$, since $|W^0| \geq (1-\delta/5)|W|$. By the assumption on the number of edges in G and the degree of vertices in V_h

$$|V_h| \leq \frac{\delta \cdot \min(\rho, \delta)}{1600} \cdot n \cdot \left(\frac{\lg n}{\lg\lg n}\right)^2 \cdot \frac{k}{n \lg^2 n} \leq \frac{\delta}{1600 \cdot (\lg\lg n)^2} \cdot k.$$

Thus, by linearity of expectation there are at least

$$\delta k - \frac{\delta}{5}k - \frac{\delta}{1600 \cdot (\lg\lg n)^2}k \geq \frac{3}{4}\delta k$$

vertex disjoint paths connecting X and Y via V_l. □

Finally, given our assumptions we derive an upper bound on the expected size of $f^a(X, Y)$ that will contradict the previous claim. Namely, we claim:

Claim 12 *Let X be a random k-element subset of U^0 and Y be a random k-element subset of W. Then the expected value of $f^a(X, Y)$ is at most $\frac{\delta}{160(\lg\lg n)^2}k$.*

Proof of the claim. By the bounds on the degree of vertices in U^0 and V_l, the number of distinct paths from U^0 to W via V_l is at most $|U^0| \cdot \frac{\delta}{160}\left(\frac{\lg n}{\lg\lg n}\right)^2 \cdot \frac{n}{k \lg^2 n} = \frac{\delta}{160k(\lg\lg n)^2} \cdot |U^0| \cdot n$. A given path between U^0 and W will be connecting a randomly chosen X and Y with probability $\frac{k}{|U^0|} \cdot \frac{k}{|W|}$. Hence, the expected number of distinct paths between randomly chosen X and Y going through V_l is at most

$$\frac{\delta}{160k(\lg\lg n)^2} \cdot |U^0| \cdot n \cdot \frac{k^2}{|U^0| \cdot |W|} = \frac{\delta}{160(\lg\lg n)^2}k.$$

□

This proves our theorem. □

We make a few more remarks regarding the difference between the above proof and the one in [28]. We note that the sets U^0 of inputs and W^0 of outputs produced by the random process are both sub-linear in the case of [28]. This is not usable for purposes of the codes as there might be linear size sets of outputs that are completely disconnected from the inputs – some bits of the codeword can be constantly set to zero. So we modify the random process to be asymmetric to obtain linear size set W^0 of outputs; U^0 will still be of polynomial albeit sublinear size.

Another difference, that also comes from the difference of Properties 1 and 2 discussed in the introduction, is in the use of the connectivity. In [28] a part of the argument uses only the property that any two k elements subsets are connected by at least one path. In the case of codes this might not be true for all k-element subsets so we have to argue differently using the fact that on the average we have a fraction of k paths between the sets.

3.2 Lower bounds for depth three and more

To prove the lower bound on the size of circuits of depth more than two computing good codes we will use known bounds on the number of wires in circuits satisfying connectivity Property 3. We use the following definitions of Pudlák.

Definition 13 (Pudlák [26]) *Let n be an integer and G be a directed acyclic graph with n inputs and n outputs. Let $0 < \epsilon, \delta$ and $0 \leq \mu \leq 1$. We say that G is ϵ, δ, μ-*densely regular *if for every k, where $\mu n \leq k \leq n$, there are probability distributions $\mathcal{X}$ and $\mathcal{Y}$ on k-element subsets of inputs and outputs, resp., such that for every $i, j \in \{1, \ldots, n\}$,*

$$\Pr_{X \in \mathcal{X}}[i \in X] \leq k/\delta n \qquad \Pr_{Y \in \mathcal{Y}}[j \in Y] \leq k/\delta n$$

and the expected number of vertex disjoint paths from X to Y is at least ϵk for randomly chosen $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. We denote by $D(n, d, \epsilon, \delta, \mu)$ the minimum number of wires in any depth-d ϵ, δ, μ-densely regular graph with n inputs and n outputs.

Extending lower bounds on superconcentrator size of Dolev et al. [11], Pudlák [26] proves the following:

Theorem 14 (Pudlák [26]) *Let $\epsilon, \delta > 0$ be constants. Then for every n, $1/n \leq \mu \leq 1$ and $d \geq 2$ the following holds:*

$$\begin{aligned} D(n, 3, \epsilon, \delta, \mu) &\in \Omega(n \cdot \lg\lg 1/\mu), \\ D(n, 2d, \epsilon, \delta, \mu) &\in \Omega(n \cdot \lambda_d(1/\mu)), \\ D(n, 2d+1, \epsilon, \delta, \mu) &\in \Omega(n \cdot \lambda_d(1/\mu)). \end{aligned}$$

Since Lemma 3 holds for any fixed k-element set X of inputs, it also holds in the case when the k-element set X of inputs is chosen at random. One gets the following corollary to Lemma 3.

Corollary 15 *Let $0 < \rho, \delta < 0$ be constants and C be a circuit computing a (ρ, δ)-good code. If we extend the circuit by $(1-\rho)n$ dummy inputs then its underlying graph is $\rho\delta, \rho, \frac{1}{n}$-densely regular.*

We add the dummy inputs only to have n inputs and n outputs. The parameters degrade because we apply the previous Lemma 3 which works only for k up-to ρn, over the ρn real inputs. So for $k \leq \rho n$ we pick a random k-element subset X of the ρn real inputs, and for $k > \rho n$ we pick a random k-element subset X of inputs by taking all the real inputs and adding a random $k - \rho n$-element subset of the $(1-\rho)n$ dummy inputs.

Theorem 16 *$w_{\rho,\delta,3}(n) \in \Omega(n \lg\lg n)$, and for any integer $d \geq 2$, $w_{\rho,\delta,2d}(n), w_{\rho,\delta,2d+1}(n) \in \Omega(n\lambda_d(n))$.*

Proof: The proof directly follows from Corollary 15 and Theorem 14. □

4 Construction of good codes with an efficient encoding

In this section we construct bounded-depth circuits computing good codes. All our circuits will consist of only parity gates of arbitrary fan-in; hence, the codes computed by our circuits will be linear. Our constructions are probabilistic.

Recall that for $0 < \alpha < 1$ and integers $0 \leq \ell \leq k \leq m$ and $0 < n$, a $(m, n, \ell, k, \alpha n)$-*range detector* is an XOR circuit with m inputs and n outputs such that on inputs of hamming weight at least ℓ and at most k, the circuit outputs at least αn ones.

The following proposition immediately follows from the existence of (ρ_0, δ_0)-good codes.

Proposition 17 *For all $1 \leq m$ there exists an $(m, m/\rho_0, 1, m, \delta_0 m)$-range detector of depth 1 and size m^2.*

We will use expander graphs to construct good range detectors. A bipartite graph $G = (V_1, V_2, E)$ is a (k, c) (vertex) expander if for all $S \subseteq V_1$ with $|S| \leq k$ it holds that $|\Gamma(S)| \geq c|S|$. We are interested in expander graphs where $c = (1 - \epsilon)d$, where d is the degree of the left side vertices V_1, for small $\epsilon > 0$. In this case G is known as a *lossless* expander. The reason we are interested in these is that when $\epsilon < 1/2$ at least $(1 - 2\epsilon)d|S|$ of the vertices in $\Gamma(S)$ must have a unique neighbor in S. In particular each of these vertices have an *odd* number of neighbors in S. For consistency with our application we will denote $m = |V_1|$ and $n = |V_2|$.

A (k, c) expander $G = (V_1, V_2, E)$ in particular is a $(k, 1 - ck/n)$ so-called disperser graph, and a lower bound for such graphs due to Radhakrishnan and Ta-Shma [28] states that $d = \Omega\left(\frac{\lg(m/k)}{\lg(n/(ck))}\right)$, when $k < m$, $2d \leq n - ck$ and $ck < n/2$.

The common setting of parameters for lossless expanders have $n \ll m$ and $n = O(ck)$, and as a consequence of the lower bound above we get $d = \Omega(\lg m)$. We are interested in having $d = O(\lg m / \operatorname{lglg} m)$, and as a consequence we at least require $n = \Omega(ck \lg^{\gamma} m)$ for some $\gamma > 0$. We show below that this can indeed be achieved. Also, for our setting of parameters the usual probabilistic existence proof does not work – instead we proceed similarly to a proof due to Buhrman et al. [7, Lemma 3.10].

Lemma 18 *Let $0 < \epsilon < 1$ and $\gamma > 0$ and let $k \leq m$ be an integer. Then there exists a $(k, (1 - \epsilon)d)$ expander graph $G = (V_1 = [m], V_2 = [n], E)$ with left side degree d, where $d = (2/\epsilon\gamma) \lg m / \operatorname{lglg} m$ and $n = (e/\epsilon)kd \lg^{\gamma} m = (2e/\epsilon^2\gamma)k \lg^{1+\gamma} m / \operatorname{lglg} m$.*

Proof: We choose G by choosing (with replacement) d neighbors in V_2 for each left side vertex in V_1. We will show that G is a $(k, (1 - \epsilon)d)$ expander with positive probability.

Let $S \subseteq V_1$ with $|S| = \ell \leq k$. We consider the process of choosing the neighbors of S, by choosing a neighbor one at a time. Let for $i = 1, \ldots, \ell d$ the random variable X_i indicate if the ith choice of neighbor was also chosen earlier, and let $X = \sum X_i$. We are interested in bounding the probability $\Pr[X > \epsilon \ell d]$. To this end, define for $i = 1, \ldots, \ell d$ independent Bernoulli variables Y_i, by $\Pr[Y_i = 1] = \ell d/n$, and let $Y = \sum Y_i$. Conditioned on any values of

$X_1, \ldots, X_{i-1}$ we have $\Pr[X_i = 1] \leq \ell d/n$ and thus we can bound $\Pr[X > \epsilon \ell d] \leq \Pr[Y > \epsilon \ell d]$. We have $E[Y] = (\ell d)^2/n$ and by the Chernoff bound for the upper tail, letting $\alpha = \epsilon n/\ell d$ we have

$$\Pr[Y \geq \epsilon \ell d] = \Pr[Y \geq \alpha E[Y]]$$
$$\leq \left(\frac{e^{\alpha-1}}{\alpha^\alpha}\right)^{E[Y]} \leq \left(\frac{e}{\alpha}\right)^{\alpha E[Y]} = \left(\frac{e\ell d}{\epsilon n}\right)^{\epsilon \ell d} .$$

Using $n \geq (e/\epsilon)\ell d \lg^\gamma m$ we have

$$\Pr[X \geq \epsilon \ell d] \leq (\lg^{-\gamma} m)^{\epsilon \ell d} = 2^{-\epsilon\gamma \ell d \lglg m} = 2^{-2\ell \lg m} = m^{-2\ell}.$$

Now, taking first a union bound over all $\binom{m}{\ell} \leq m^\ell$ sets $S \subseteq V_1$ of size $|S| = \ell$ and then another union bound over all $\ell \leq k$ we obtain the stated result. □

Setting $\epsilon = 1/4$, by the discussion and the lemma above we obtain the following.

Corollary 19 *Let $\gamma > 0$ be arbitrary. Then for any integers m and $\ell \leq k \leq m$ there exist a $(m, n, \ell, k, \alpha n, \beta n)$-range detector with m inputs, $n = (32ek \lg^{1+\gamma} m)/(\gamma \lglg m)$ outputs, $\beta = 1/(4e \lg^\gamma m)$, $\alpha = (\ell/2k)\beta$ that consists of at most $(8m \lg m)/(\gamma \lglg m)$ wires.*

We are now in position to present the construction of depth-2 circuits computing error correcting codes.

Theorem 20 *Let $0 < \kappa < 1$ be arbitrary and let m be an integer. There exists a depth-2 XOR circuit with m inputs and $32m$ outputs, using $O(m\,(\lg m/\lg\lg m)^2)$ wires computing an error correcting code of minimum distance $1/8$. Furthermore, the fan-in of each output gate is $O(\lg^{1+\kappa} m)$.*

Proof: Let $m \geq 1$ be an integer and let $\gamma, \lambda > 0$, be reals such that $\gamma + \lambda < \kappa$. Let k be an integer such that $k + 1 = \lceil \frac{\lg m}{\lambda \lglg m} \rceil$. The middle layer of the circuit will consist of $k + 1$ range detectors $W_0, \ldots, W_k$ where each W_i is a detector for the range of input weights $\lg^{\lambda i} m$ to $\lg^{\lambda(i+1)} m$. By Corollary 19 we can obtain this using a range detector of size $(8m \lg m)/(\gamma \lglg m)$ with parameters

$$\left(m, n, \lg^{\lambda i} m, \lg^{\lambda(i+1)} m, \frac{n}{8e \lg^{\gamma+\lambda} m}, \frac{n}{4e \lg^\gamma m}\right) ,$$

where

$$n = \frac{32e \lg^{1+\gamma+\lambda(i+1)} m}{\gamma \lglg m}.$$

Note that $k + 1 = O(\lg m/\lg\lg m)$, and hence we have used $O(m\,(\lg m/\lg\lg m)^2)$ wires in total so far.

Define $\alpha = 1/(8e \lg^{\gamma+\lambda} m)$ and $\ell = 8e \lg^{\gamma+\lambda} m$. For the last layer, each node is an XOR of $(k + 1)\ell$ nodes, where we chose ℓ nodes at random from each W_i. We will have a total of $32m$ such XOR's.

Consider now a fixed input x of weight $w > 0$, and let i be such that $\lg^{\lambda i} m \leq w \leq \lg^{\lambda(i+1)} m$. Our goal is to show that with high probability over the random choice of neighbors of the top XOR gates at least $4m$ of the top XOR gates will evaluate to 1 on x. On input x we have ensured that in W_i a fraction α of the XOR's of W_i are odd. Conditioned on the values of all the XOR's of the second layer outside W_i, by Proposition 37 the probability that a given XOR of the last layer is odd is at least $\alpha\ell/4 = 1/4$. Letting X be the random variable indicating the number of XOR's in the last layer that are odd, we then have $E[X] \geq 8m$, and using the Chernoff bound for the lower tail we have

$$\begin{aligned}\Pr[X < 4m] &\leq \Pr[X < E[X]/2] < \exp(-E[X]/8) \\ &\leq \exp(-m) < 2^{-m}\end{aligned}$$

Thus taking a union bound over all $2^m - 1$ inputs we have that the circuit can be constructed to compute an error correcting code of minimum distance $1/8$.

Between the second layer and the last layer we have used a total number of

$$32m(k+1)8e \lg^{\gamma+\lambda} m = O(m \lg^{1+\gamma+\lambda} m / \operatorname{lglg} m)$$

wires, and the fan-in of each gate of the last layer is $(k+1)8e \lg^{\gamma+\lambda} m = O(\lg^{1+\gamma+\lambda} m / \operatorname{lglg} m)$. Thus the total size of the circuit is $O(m (\lg m / \lg\lg m)^2)$ and the top fan-in is $O(\lg^{1+\kappa} m)$. □

4.1 Comparison with depth-2 superconcentrators

It is instructive to compare our construction to depth-2 superconcentrators. Radhakrishnan and Ta-Shma [28] show that depth-2 superconcentrators must be of size at least $\Omega(m \lg^2 m / \operatorname{lglg} m)$, while we show that error correcting codes can be computed by size $O(m (\lg m / \lg\lg m)^2)$ circuits. Furthermore we can have the fan-in of the output be $O(\lg^{1+\kappa} m)$ for arbitrary $\kappa > 0$, while maintaining the bound on the total size. Dutta and Radhakrishnan [12], show that if G is a depth-2 superconcentrator with m inputs and outputs, where the outputs have average fan-in a and inputs have average fanout b then $a \lg(2b/a) \lg b = \Omega(\lg^2 m)$ (by symmetry the reverse relation also holds). In particular if $a = \lg^{1+\kappa} m$ for $0 < \kappa < 1$ we have $b = 2^{\Omega(\lg^{(1-\kappa)/2} M)}$, hence the size is much bigger than in the case of circuits computing good codes.

4.2 Depth three and more

In this section we construct circuits of depth higher than two computing good codes. We will need several kinds of circuits—range detectors of depth 1 described in the following lemmas.

Lemma 21 *There exists $c_0 > 1$ such that for all $1 \leq r \leq m/k^{1.5}$ and $c_0 \leq k \leq m$ there exists an $(m, m/k, r, m/k^{1.5}, r)$-range detector of depth 1 and size $3m$.*

Proof: Let $n = m/k$. The edges of the range detector circuit form a bipartite graph where we denote by V_1 the input gates and by V_2 the output XOR gates. The bipartite graph could be a random graph where all vertices $x \in V_1$ have degree 3. However, to simplify the computation we will pick randomly 3 elements from V_2 for every $x \in V_1$ *allowing repetitions.* So some vertices in V_1 may have smaller degree.

Let $r \leq \ell \leq m/k^{1.5}$ and let $S \subseteq V_1$, $|S| = \ell$ be fixed. Let $y_1, \ldots, y_{3\ell}$ be the chosen neighbors, allowing repetitions. We want to show that S has at least r unique neighbors with high probability. That will suffice to prove the lower bound on the number of ones in the output.

For $i = 1, \ldots, 3\ell - 1$, let X_i be the random variable that is the indicator of the event $y_{i+1} \in \{y_1, \ldots, y_i\}$. Note that the number of unique neighbors is at least $3\ell - 2\sum X_i$. If $b_1, \ldots, b_j$ are fixed $\Pr[X_{j+1} = 1 | X_1 = b_1 \wedge \cdots \wedge X_j = b_j] \leq \frac{3\ell}{n}$. Hence we can apply the Chernoff bound, as in Lemma 18, to bound $\Pr[\sum_{i=1}^{3\ell-1} X_i \geq \ell]$. We have

$$E[\sum_{i=1}^{3\ell-1} X_i] \leq \frac{3\ell(3\ell-1)}{n} \leq \frac{9\ell^2 k}{m}.$$

Let $\alpha = \frac{m}{9\ell k}$. We have $\alpha > 1$ for $k > 81$ using $\ell \leq m/k^{1.5}$. By Lemma 35 we get

$$\begin{aligned}\Pr[\sum_{i=1}^{3\ell-1} X_i \geq \ell] &\leq \left(\frac{e^{\alpha-1}}{\alpha^\alpha}\right)^{\frac{9\ell^2 k}{m}} \leq \left(\frac{e}{\alpha}\right)^{\alpha\frac{9\ell^2 k}{m}} \\ &\leq \left(\frac{9e\ell k}{m}\right)^{\ell} \leq \left(\frac{9e}{\sqrt{k}}\right)^{\ell}.\end{aligned}$$

If k is sufficiently large, this is less than $2^{-\ell}$. Thus we have nonzero probability that for all ℓ, $r \leq \ell \leq m^{1.5}$, we have r unique neighbors. □

Corollary 22 *There exists $c_0 > 1$ such that for all $c_0^2 \leq m$ there is an $(m, \sqrt{m}/\rho_0, 1, m^{1/4}, \delta_0 m)$-range detector of depth 2 and size $4m$.*

Proof: Lemma 21 gives a depth-1 $(m, \sqrt{m}, 1, m^{1/4}, 1)$-range detector with $3m$ wires. The result follows by applying Proposition 17 on the output of that range detector. □

We can amplify the quality of range detectors using the following lemma.

Lemma 23 *For any $0 < \alpha$ and $1 \leq k \leq m \leq n \leq w$, if there is an $(m, n, m/k, m, \alpha n)$-range detector of depth d with w wires then there is an $(m, 32m, m/k, m, 4m)$-range detector of depth d with $65w/\alpha$ wires. Consequently, there is an $(m, 32m, m/k, m, 4m)$-range detector of depth 1 with $65mk$ wires.*

Proof:

First we construct a circuit of depth $d + 1$. Each of the $32m$ output vertices will be an XOR of at most $\lceil 2/\alpha \rceil$ outputs of the depth-d range detector chosen by the following random

process. In each of $\lceil 2/\alpha \rceil$ steps we with probability $1/2$ do nothing and with probability $1/2$ we pick a random gate of the range detector. We claim that on a fixed input of hamming weight at least m/k the XOR evaluates to 1 with probability at least $1/4$. This follows immediately from Proposition 37. Hence, if we take $32m$ of such XOR gates independently at random, the probability that less than $4m$ of them are 1 is by the Chernoff bound at most e^{-m}.

Since we want to maintain the depth of the circuits we cannot add these XOR gates as another layer but instead we collapse them with the existing output gates and we remove the original output gates. This possibly leads to an increase in the number of wires. A single new XOR gate contributes by at most $w/\alpha n$ wires in expectation so the new XOR gates will contribute by at most $32wm/\alpha n \leq 32w/\alpha$ wires in expectation. As all the random choices are independent by the Chernoff bound, the probability that the new XOR gates contribute by more than $64w/\alpha$ wires is at most $(e/4)^{32w/\alpha} < (e/4)^{32m} < 2^{-2m}$. So there is a particular choice of the $32m$ output gates that will contribute by at most $64w/\alpha$ wires and on any input of hamming weight at least m/k at least $4m$ of the output gates will evaluate to 1. This proves the first part of the lemma.

The existence of the depth-1 range detector follows by noting that a circuit computing the identity function is an $(m, m, m/k, m, m/k)$-range detector. □

Combining the two lemmas we get:

Corollary 24 *There exists $c_0 > 1$ such that for all $c_0 \leq k \leq m$, there exists an $(m, 32m/k, m/k^2, m/k^{1.5}, 4m/k)$-range detector of depth 2 and size $68m$.*

We are ready to construct circuits of even depth bigger than two using the following lemma. The case of depth-three circuits is left to the end of this section.

Lemma 25 *Let $d \geq 2$ and c_d be constants and $f(m) \leq \lg m$ be a non-decreasing unbounded function. There exists a constant c_{d+2} such that if for each $1 \leq r \leq m$, there is a depth-d $(m, 32m, m/r, m, 4m)$-range detector with at most $c_d m \cdot f^2(r)$ wires then for each $1 \leq r \leq m$ there is a depth-$d+2$ $(m, 32m, m/r, m, 4m)$-range detector with at most $c_{d+2} m \cdot f^*(r)$ wires.*

Proof: Let $k_1 = \min\{r, m^{3/4}\}$, let $k_{i+1} = f(k_i)^3$ and let t be the least integer such that $k_t \leq c_0^{1.5}$. The first layer of the circuit contains an $(m, m/f(k_i)^2, m/k_i, m/f(k_i)^3, m/k_i)$-range detector provided by Lemma 21, for each $i = 1, \ldots, t-1$. We assume that the detectors are disjoint except for the same input vertices. The output of the i-th depth-1 range detector feeds into a depth-d $(m/f(k_i)^2, 32m/f(k_i)^2, m/k_i, m/f(k_i)^2, 4m/f(k_i)^2)$-range detector, called W_i, given by the assumptions of the lemma.

Furthermore, we add the depth-1 $(m, 32m, m/k_{t-1}, m, 4m)$-range detector, called W_t, given in Lemma 23. (Observe, that $k_{t-1} < \min\{a, f(a) > c_0^{1.5}\}$ so the size of this detector can be upper-bounded in terms of properties of f.) If $r > k_1$ then to capture the strings of weights $< m/k_1$ we add the depth-2 $(m, \sqrt{m}/\rho_0, 1, m^{1/4}, \delta_0 m)$-range detector, called W_0 from Corollary 22.

For the last layer, we take Cm XOR gates for a sufficiently large constant C and each of the gates will be fed by at most one randomly picked output gate of each $W_0, \ldots, W_t$ ($W_1, \ldots, W_t$ if $r \leq k_1$.) For each output XOR gate and each W_i we do not connect them with probability $1/2$ and we connect them with probability $1/2$ (via the random output gate of W_i.) Thus on any input of weight at least m/r, each output XOR gate is one with probability at least $1/16$. An application of the Chernoff bound, as in the proof of Theorem 20 and amplification by Lemma 23 gives the required range detection property of the circuit.

To count the number of wires, notice that $f^3(f^3(a)) \leq \lceil 3 \lg f(a) \rceil^3 < f(a)$ for a sufficiently large, hence there is some constant c depending only on f such that $t \leq 2f^*(r) + c$. Furthermore, each W_i detector consists of at most $c_d \cdot \frac{m}{f(k_i)^2} \cdot f(k_i/f(k_i)^2)^2 \leq cm$ wires. Since all the other range detectors built into the construction also use only $O(m)$ wires, the total size of the resulting range detector is $O(tm) \in O(mf^*(r))$. □

We will use the above lemma recursively. As the base construction we will need also the following type of range detectors.

Lemma 26 *There exists a constant c_2 such that for all $1 \leq r \leq m$, there is a depth-2 $(m, 32m, m/r, m, 4m)$-range detector with at most $c_2 m \cdot \lg^2(r)$ wires.*

Proof: This construction is similar to our other constructions so we provide only a brief sketch. Let $k_1 = r$, let $k_{i+1} = k_i/2$ and let t be the least integer such that $k_t \leq 1$. Define $n_i = 2\lceil \lg \binom{m}{m/k_i} \rceil$. For $i = 1, \ldots, t-1$, one can construct $(m, 32n_i, m/k_i, m/k_{i+1}, 4n_i)$-range detectors of depth 1 and size $O(k_i n_i) = O(m \lg k_i)$ by a probabilistic argument. Hence, take these range detectors as the first layer of the circuit. The output layer is formed by taking random XOR's of outputs from these detectors similarly as in the proof of Lemma 25. Clearly, the size of the circuit is bounded by $O(tm \lg r) = O(m \lg^2 r)$. □

Recall that $(m, 32m, 1, m, 4m)$-range detector is a circuit computing a $(1/32, 1/8)$-good code. Thus, the previous two lemmas give the following corollary.

Corollary 27 *For any $d \geq 2$ there exists a constant c_d and a family of depth-$2d$ circuits with $c_d m \lambda_d(m)$ wires that computes $(1/32, 1/8)$-good codes.*

It remains to construct a circuit of depth 3 that has size $O(m \lg \lg m)$ and computes a good code.

Theorem 28 *There is a family of depth-3 circuits with $O(m \lg \lg m)$ wires that computes $(1/32, 1/8)$-good codes.*

Proof: The proof is similar to the proof of Lemma 25. However, we let parameters $k_1 = \sqrt{m}$, $k_{i+1} = k_i^{3/4}$ and let t be the least integer such that $k_t \leq c_0$ (i.e., $t < \lg_{4/3} \lg_{k_1} \sqrt{m}$). The first two layers of the circuit consist of the depth-2 $(m, 32m/k_i, m/k_i^2, m/k_i^{1.5}, 4m/k_i)$-range detectors, for $i = 2, \ldots, t-1$, given in Corollary 24, together with the $(m, 32m, m/k_t, m, 4m)$-range detector of depth 1 from Lemma 23 to capture the input strings of weight $> m/k_t$,

and the depth-2 $(m, \sqrt{m}/\rho_0, 1, m^{1/4}, \delta_0 m)$-range detector from Corollary 22 to capture the strings of weights $\leq m/k_1$. In total we use $O(m \lg \lg m)$ wires for the first two layers.

The output layer is formed by taking random XOR's of outputs from the above detectors as in previous proofs. □

4.3 Unbounded-depth circuits

In this section we prove Theorem 2: we obtain linear-size circuits computing good codes with slowly-growing depth. To obtain such circuits, consider the constant-depth constructions in Theorem 1. Then apply the following corollary to trade depth for size.

Corollary 29 *Let $d \geq 2$, c_d be a constant and $f(m) \leq \lg m$ be a non-decreasing unbounded function. There exists a constant c such that if for each $1 \leq m$, there is a depth-$2d$ $(m, 32m, 1, m, 4m)$-range detector with at most $c_d m \cdot f(m)$ wires then for any $m \geq 1$ there is depth-$2d + 2\lg_{c_0} f(m)$ $(m, 32m, 1, m, 4m)$-range detector with at most cm wires where c_0 is the constant from Lemma 21.*

Hence, for any fixed $d > 0$, there are linear size circuits of depth $O(\lg(\lambda_d(m)))$ computing good codes.

The proof of the above corollary relies on a lemma which we state and prove next.

Lemma 30 *For any $m \geq c_0$, any d and w if there is a depth-d $(m/c_0, 32m/c_0, 1, m/c_0, 4m/c_0)$-range detector of size w then there is also a depth-$d+2$ $(m, 32m, 1, m, 4m)$-range detector of size $w + (9 + (2c_0^{1.5} + 16)32)m$.*

Proof: Take the $(m, m/c_0, 1, m/c_0^{1.5}, 1)$-range detector of depth 1 and size $9m$ provided by Lemma 21 and apply on its output the depth-d $(m/c_0, 32m/c_0, 1, m/c_0, 4m/c_0)$-range detector provided by the lemma. This gives a circuit with the property that on each non-zero input either the input has at least $1/c_0^{1.5}$ fraction of ones or the output of the depth-d detector has at least $1/8$ fraction of ones. Consider an XOR gate that takes at most $2c_0^{1.5}$ random bits of the input and at most 16 random outputs of the depth-d range detector and is obtained as follows: $2c_0^{1.5}$ times repeat: with probability $1/2$ take a random input bit and with probability $1/2$ take nothing; then repeat 16 times: with probability $1/2$ take a random output bit of the detector and with probability $1/2$ take nothing. Such an XOR gate will evaluate to one with probability at least $1/4$ by Proposition 37. By the Chernoff bound, on any fixed input out of $32m$ such independently chosen XOR gates at least $4m$ will evaluate to one with probability at least $1 - e^{-m}$. Hence, there is a particular choice of the $32m$ XOR gates so that on any non-zero input at least $4m$ of them will evaluate to one. That forms our depth-$d+2$ $(m, 32m, 1, m, 4m)$-range detector of size at most $w + 9m + (2c_0^{1.5} + 16)32m$. □

Proof:[of Corollary 29] Let $t = \lceil \lg_{c_0} f(m) \rceil$ and for $i = 0, \ldots, t$, define $m_i = m/c_0^{t-i}$. The depth-$2d$ range detector guaranteed by the assumption of the lemma on inputs of size m_0

uses $c_d m_0 f(m_0) \leq c_d m$ wires. For $i = 1, \ldots, t$, iteratively apply to this range detector the previous lemma to construct $(m_i, 32m_i, 1, m_i, 4m_i)$-range detector. The depth of the resulting range detector is $2d+2t$ and its size is bounded by $c_d m + \sum_{i=1}^{t}(9+(2c_0^{1.5}+16)32)m_i \in O(m)$. □

5 Hash functions

In this section we first show how our constructions of efficient encoding circuits imply similarly efficient circuits for pairwise independent hash functions. At the end we also deduce lower bounds for various types of multiplication.

For the first result, we follow closely the results by Ishai, Kushilevitz, Ostrovsky, and Sahai [17]. Our main contribution is observing that their construction can be applied in our constant-depth setting. In fact, in our setting the proof is somewhat simpler (the presentation in [17] relies on several previous results in coding theory; it also uses an argument about obtaining a circuit for the transpose of the function computed by another circuit which is immediate in our setting).

A *pairwise independent hash function* is a map $h : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^m$ such that for fixed m-bit strings $x \neq y$, the values $h(x, R)$ and $h(y, R)$ are uniformly and independently distributed for uniform $R \in \{0,1\}^r$.

Theorem 5 (Implicit in [17]) *Suppose there are constants ρ, δ, d, and an increasing function $w(n) \geq n$ such that*

1. *for any n there is a linear function $C : \{0,1\}^{\rho n} \to \{0,1\}^n$ that is a (ρ, δ)-good code, and that can be computed by a depth-d XOR circuit with $w(n)$ wires.*

Then there is a constant c such that

2. *for every m we can compute a pairwise independent hash function $h : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^m$ with $r \leq cm$, by a depth-$2d$ circuit with $\leq cw(cm)$ wires (using arbitrary gates).*

Moreover, if the codes are explicit then the hash functions are too.

We now turn to the proof of the reduction in Theorem 5. The explicitness of the reduction is immediate from the proof and we will not address it explicitly. The proof needs several ingredients. First, we need codes over large alphabet with relative distance close to 1. To achieve this with a linear number of wires, [17] cite [15] (presumably Theorem 11) which – inspired by [2] – uses expander graphs on top of the code in [29]. We observe that a similar approach works in our setting with no blow-up in the depth.

We say that a function $C' : \{0,1\}^m \to (\{0,1\}^t)^n$ is a t-bit-alphabet code with distance δ if $\forall x \neq y$, there are $\geq \delta n$ indices $i \in [n]$ such that the t-bit symbols $C(x)_i$ and $C(y)_i$ are different.

Lemma 31 *Under the assumption of Theorem 5, for every $\epsilon > 0$ there is an integer t such that for every n there is a t-bit-alphabet code $C' : \{0,1\}^{\rho n} \to (\{0,1\}^t)^{n4^t}$ with distance $(1-\epsilon)$ that can be computed by a depth-d circuit with $\leq t4^t w(n)$ wires.*

Proof: For a message $x \in \{0,1\}^{\rho n}$, we first encode x into $C(x) \in \{0,1\}^n$ with the code $C : \{0,1\}^{\rho n} \to \{0,1\}^n$ in Theorem 5.

Now consider an expander graph on n nodes with degree, say, 4 and normalized second largest eigenvalue $\leq 1-\Omega(1)$. Let $t = t(\epsilon)$ be a constant to be determined later. The indexes of t-bit codeword symbols of the new code C' are identified with walks of length t on the expander. Since a walk can be written with $\lg n + 2t$ bits, a codeword in C' has $n4^t$ symbols of t bits. The ith symbol of C' is set to the concatenation of the t bits in $C(x)$ specified by the walk. Since to any $x \neq y$ there correspond in C codewords at constant relative hamming distance, the encodings of x and y in the new code agree in at most a $2^{-\Omega(t)}$ fraction of t-bit symbols. Here we use standard hitting properties of random walks on expanders [1, 19]. For large enough t, this gives the desired relative distance.

Regarding the complexity, note that the construction just amounts to repeating bits of the code according to the expander. This clearly does not affect depth. As for the number of wires, for example we can note that for any index bit $i \in [n]$ of C, a random walk hits i with probability $\leq t/n$, by the union bound. Since the number of walks is $n4^t$, bit i participates in at most $t4^t$ walks. Therefore, the number of wires increases by a factor $\leq t4^t$. □

Another component of the construction is a *resilient function* [10]. A function $f : \{0,1\}^n \to \{0,1\}^m$ is *resilient to fixing t bits* if for every random variable $X \in \{0,1\}^n$ where at least $n-t$ bits are uniform and i.i.d. and the others are constants, we have that $f(X)$ is uniform over $\{0,1\}^m$. We denote by M^T the transpose of the matrix M.

Fact 32 ([10]) *Let G be the $n \times m$ generator matrix of a (ρ, δ)-good code. Then $f : \{0,1\}^n \to \{0,1\}^m$ defined by $f(x) = G^T x$ is resilient to fixing $\delta n - 1$ bits.*

Proof: Let $X \in \{0,1\}^n$ be a r.v. where at least $n - \delta n + 1$ bits are uniform and i.i.d. and the others are constants. By the XOR lemma in [10], it is enough to prove that for any vector $a \in \{0,1\}^m$ the value $aG^T X$ is a uniform bit. Note $aG^T X = (Ga)X$. Since G generates a linear code with distance δn, Ga is a vector of hamming weight $\geq \delta n$. Hence Ga has a 1 in a position corresponding to a uniform bit of X, and so $(Ga)X$ is unbiased. □

Corollary 33 *Under the assumption of Theorem 5, for every n there is a function $f : \{0,1\}^n \to \{0,1\}^{\rho n}$ that is resilient to fixing $\delta n - 1$ bits and is computable by a depth-d XOR circuit with w wires.*

Proof: The generator matrix G of the code in Theorem 5 can be decomposed as $G = G_1 G_2 \cdots G_d$ where the sum of the number of 1's in the matrices is $\leq w$ (cf. "Decomposing the generator matrix of good codes into sparse matrices." in Section 1.1). Note $G^T = G_d^T G_{d-1}^T \cdots G_1^T$ and apply Fact 32. □

We can now prove Theorem 5.

Proof:[of Theorem 5] The construction in [17] consists of the following 3 steps. On input $x \in \{0,1\}^m$ and R:

(1) Let $m = \rho n$. Using Lemma 31 encode $x \in \{0,1\}^{\rho n}$ into $C'(x)$ with a t-bit-alphabet code $C' : \{0,1\}^{\rho n} \to (\{0,1\}^t)^{n4^t}$ with distance $(1-\epsilon)$, for a sufficiently small ϵ and a t depending on ϵ only. Each codeword takes $n' := nt4^t$ bits to specify.

(2) Use the randomness R to apply a pairwise-independent function to every t-bit symbol of $C'(x)$, using different seed for each symbol.

(3) Use Corollary 33 to apply a function with input length $= n'$ and range $\rho n'$ bits that is resilient to fixing $\delta n' - 1$ bits.

For a small enough constant ϵ, for any $x \neq y$ the corresponding images after (1) have $n4^t(1-\epsilon)$ different symbols. After (2), we are in the following situation. There is a set $S \subseteq [n']$ of size $\geq n'(1-\epsilon)$ such that the projections over S of the encodings of x and y are jointly distributed, over the choice of R, uniformly over $\left(\{0,1\}^{|S|}\right)^2$.

Setting ϵ sufficiently small, the complement of S has size $\leq \epsilon n' \leq \delta n' - 1$. (Recall δ is fixed.) Since the function in (3) is resilient to fixing $\delta n' - 1$, the images corresponding to x and y are uniformly and independently distributed over $\rho n' = \rho nt4^t = mt4^t \geq m$ bits, since $m = \rho n$.

Regarding the complexity, note that (2) can be collapsed with (1) with no increase in depth and only a constant-factor increase in the number of wires. Using Lemma 31 for (1) and Corollary 33 we get the desired complexity. □

We remark that in the previous proof if one were to settle for a hashing circuit of depth $2d+1$ then one could build it from XOR gates and binary AND gates.

Finally, we mention three other natural functions to which our lower bounds in Theorem 1 apply as stated. The first is multiplication by some fixed (non-explicitly given) element a of the finite field with 2^n elements: $f_1 : \{0,1\}^n \to \{0,1\}^n$ defined as $f_1(x) := ax$. The second is multiplication by some fixed (non-explicitly given) $n \times n$ Toeplitz matrix M: $f_2 : \{0,1\}^n \to \{0,1\}^n$ defined as $f_2(x) := Mx$. The third is multiplication by some fixed (non-explicitly given) n-bit integer a: $f_3 : \{0,1\}^n \to \{0,1\}^{2n}$ defined as $f_3(x) := ax$.

We now give the details in the case of f_1. The reasoning for f_2 is identical. Recall that $h(x,(a,b)) := ax + b$ is a pairwise independent hash function. By Fact 4, there exist fixed a and b such that, for appropriate $m = \Omega(n)$, the map $f_1' : \{0,1\}^m \to \{0,1\}^n$ defined as $f_1'(x) = ax + b$ is the encoding map of a good error-correcting code. Hence, the lower bounds in Theorem 1 apply as stated to f_1'. Since addition by b does not affect complexity in our model, the same lower bounds apply to f_1.

Here is an alternative argument for f_1. One can directly argue that there are field elements a such that $x \to (x, ax)$ defines a good code. This is because if $x \neq 0$, then x and ax determine a. Since the number of pairs $(x,y) \in \{0,1\}^{2n}$ of hamming weight $< H^{-1}(1/2)2n$ is less than 2^n, we cannot have for every a an x such that both x and ax have hamming weight $< H^{-1}(1/2)n$.

This alternative argument also applies to f_3, except the relative distance is smaller. For depth $d \geq 3$, these lower bounds for f_2 and f_3 are not new; they follow from [11, 3, 26].

6 Some remarks on explicit constructions

In this section we make some remarks on explicit constructions. First, for every $d > 0$ the encoding circuit with depth d and size $O(n^{1+1/d})$ mentioned in the introduction is explicit: use $\Theta(n^{(d-1)/d})$ explicit depth-one circuits of size $O(n^{2/d})$ for computing good codes on input size $\Theta(n^{1/d})$. In the rest of this section we make some steps towards making our depth-2 construction with $n \cdot \operatorname{poly} \lg n$ wires explicit. For simplicity, we refer to the depth-2 construction of size $O(n \lg^2 n)$ mentioned in the introduction. Recall in this construction the input length is m, the output length is $n = O(m)$, and the middle layer is made of the output gates of $\lg m$ range detectors. The i-th range detector has $\leq bm/c^i$ output gates for some constants $b > 1, c > 1$. Its property is that for every non-zero input, at least one range detector has a constant α fraction of its output gates that evaluate to 1. The output layer of the depth-2 construction then combines the outputs of the range detectors into the codeword. The next claim obtains this last layer explicitly. It bounds the output degree by $O(\lg m)$ which obviously implies $\leq O(m \lg m)$ wires.

Claim 34 *Fix any $b > 1, c > 1, \alpha > 0$. For all sufficiently large m there is a depth-1 circuit, consisting of parity gates, with $O(m)$ output gates, $O(m)$ input gates divided in $\lg m$ groups of bm/c^i for $i = 1, 2, \ldots, \lg m$, and output degree $O(\lg m)$ such that: on any input where at least some group of input bits has $\geq \alpha$ relative hamming weight, the output has $\Omega(1)$ relative hamming weight.*

Proof: We are going to construct a bipartite graph with the same input nodes as the input gates in the claim, $\ell := m/\lg m$ output nodes, and output degree $O(\lg m)$, with the property that on every input as in the statement of the claim a constant fraction of the output nodes are adjacent to at least one node set to 1. Then the claim follows by replacing each output node of the graph by a good error-correcting code on its $O(\lg m)$ neighbors. The code is implemented by a depth-1 circuit with $O(\lg m)$ output parity gates of output degree $O(\lg m)$. Thus the total number of output gates will be $O(\lg m)m/\lg m = O(m)$. The hamming weight guarantee follows from the fact that each node that is adjacent to at least one node set to 1 will give rise to $\Omega(\lg m)$ ones.

The graph is constructed by adding neighbors separately for each group.

If the group consists of $t \geq \ell$ nodes, we divide the t nodes in the group into ℓ blocks of size t/ℓ, and connect an output node to each node in the corresponding group. The output degree is $t/\ell \leq (bm/c^i)/(m/\lg m) = b \lg m/c^i$, if this is the i-th group. Since picking a random neighbor of a random output node yields a random node in the group, we see that if this group has a constant fraction of ones we also have a constant fraction of output nodes that are adjacent to some node set to one.

If the group consists of $t < \ell$ nodes, we divide the output nodes in t blocks of size ℓ/t; and we connect each node in the group to all the output nodes in the corresponding block.

Here the output degree is 1. It is easy to see that if this group has a constant fraction of ones we also have a constant fraction of output nodes that are adjacent to some node set to one.

The i-th group contributes $\leq \max\{1, b \lg m/c^i\} \leq 1 + b \lg m/c^i$ to the output degree. Summing over all groups gives output degree $O(\lg m)$. □

It is an open problem to construct suitable range detectors. Specifically, we know of constructions when the hamming weight to be detected is close to 0 or close to m, but we do not know how to handle, say, hamming weight m^ϵ.

Acknowledgements. We are grateful to Jaikumar Radhakrishnan for explaining us the proof of superconcentrator lower bound [28] during the workshop *Synergies in Lower Bounds* in Aarhus. We thank Mahdi Cheraghchi for pointing us to some references. Finally, we are grateful to the organizers of the Dagstuhl Seminar (11121) on "Computational Complexity of Discrete Problems" for inviting all of us to participate. The seminar provided a perfect environment for us to continue our research.

References

[1] M. Ajtai, J. Komlós, and E. Szemerédi. Deterministic simulation in LOGSPACE. In *19th Annual ACM Symposium on Theory of Computing (STOC'87)*, pages 132–140. ACM Press, 1987.

[2] N. Alon, J. Bruck, J. Naor, M. Naor, and R. M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509–516, 1992.

[3] N. Alon and P. Pudlák. Superconcentrators of depth 2 and 3; odd levels help (rarely). *Journal of Computer and System Sciences*, 48(1):194–202, 1994.

[4] L. Bazzi, M. Mahdian, and D. A. Spielman. The minimum distance of turbo-like codes. *IEEE Transactions on Information Theory*, 55(1):6–15, 2009.

[5] L. Bazzi and S. K. Mitter. Encoding complexity versus minimum distance. *IEEE Transactions on Information Theory*, 51(6):2103–2112, 2005.

[6] R. Boppana. The average sensitivity of bounded-depth circuits. *Information Processing Letters*, 63(5):257–261, 1997.

[7] H. Buhrman, P. B. Miltersen, J. Radhakrishnan, and S. Venkatesh. Are bitvectors optimal? *SIAM Journal on Computing*, 31(6):1723–1744, 2002.

[8] A. K. Chandra, S. Fortune, and R. J. Lipton. Lower bounds for constant depth circuits for prefix problems. In *10th Colloquium on Automata, Languages and Programming (ICALP'83)*, volume 154 of *LNCS*, pages 109–117. Springer, 1983.

[9] A. K. Chandra, S. Fortune, and R. J. Lipton. Unbounded fan-in circuits and associative functions. *Journal of Computer and System Sciences*, 30(2):222–234, 1985.

[10] B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions. In *26th Annual IEEE Symposium on Foundations of Computer Science (FOCS'85)*, pages 396–407. IEEE Computer Society Press, 1985.

[11] D. Dolev, C. Dwork, N. Pippenger, and A. Wigderson. Superconcentrators, generalizers and generalized connectors with limited depth. In *15th Annual ACM Symposium on Theory of Computing (STOC'83)*, pages 42–51. ACM Press, 1983.

[12] C. Dutta and J. Radhakrishnan. Tradeoffs in depth-two superconcentrators. In *23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS'06)*, volume 3884 of *LNCS*, pages 372–383. Springer, 2006.

[13] L. R. Ford and D. R. Fulkerson. *Flows in networks*. Princeton University Press, 1962.

[14] S. Gelfand, R. Dobrushin, and M. Pinsker. On the complexity of coding. In *2nd International Symposium on Information Theory*, pages 177–184. Akademiai Kiado, 1973.

[15] V. Guruswami and P. Indyk. Expander-based constructions of efficiently decodable codes. In *42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'01)*, pages 658–667. IEEE Computer Society Press, 2001.

[16] G. Hansel. Nombre minimal de contacts de fermature nécessaires pour réaliser une fonction booléenne symétrique de n variables. *C. R. Acad. Sci. Paris*, 258:6037–6040, 1964.

[17] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with constant computational overhead. In *14th Annual ACM Symposium on Theory of Computing (STOC'08)*, pages 433–442. ACM Press, 2008.

[18] S. Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012.

[19] N. Kahale. Eigenvalues and expansion of regular graphs. *Journal of the ACM*, 42(5):1091–1106, 1995.

[20] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[21] S. Lovett and E. Viola. Bounded-depth circuits cannot sample good codes. In *25th Annual IEEE Conference on Computational Complexity (CCC'10)*, pages 243–251. IEEE Computer Society Press, 2010.

[22] Y. Mansour, N. Nisan, and P. Tiwari. The computational complexity of universal hashing. *Theoretical Computer Science*, 107(1):121–133, 1993.

[23] P. B. Miltersen. Error correcting codes, perfect hashing circuits, and deterministic dynamic dictionaries. In *9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 556–563. ACM Press, 1998.

[24] J. G. Oxley. *Matroid theory*. Oxford University Press, 1992.

[25] H. Perfect. Applications of Menger's graph theorem. *Journal of Mathematical Analysis and Applications*, 22:96–111, 1968.

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

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

[28] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[29] D. Spielman. *Computationally Efficient Error-Correcting Codes and Holographic Proofs*. PhD thesis, Massachusetts Institute of Technology, 1995.

[30] D. A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996.

[31] L. Valiant. On non-linear lower bounds in computational complexity. In *7th Annual ACM Symposium on Theory of Computing (STOC'75)*, pages 45–53. ACM Press, 1975.

[32] E. Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2004.

[33] D. J. Welsh. *Matroid theory*. Academic Press, London, 1976.

7 Appendix

7.1 Some helpful facts

In this section we provide a few simple facts on probability.

Lemma 35 (Multiplicative Chernoff bound)
Let $X_1, \ldots, X_n$ be independent Bernoulli random variables, and $X = \sum X_i$. Then for any $\epsilon > 0$ we have the upper tail *bound*

$$\Pr[X \geq (1+\epsilon)E[X]] \leq \left(\frac{e^{\epsilon}}{(1+\epsilon)^{(1+\epsilon)}}\right)^{E[X]} \leq \left(\frac{e}{1+\epsilon}\right)^{(1+\epsilon)E[X]}.$$

and for any $0 < \epsilon < 1$ *the* lower tail *bound*

$$\Pr[X \leq (1-\epsilon)E[X]] \leq \left(\frac{e^{-\epsilon}}{(1-\epsilon)^{(1-\epsilon)}}\right)^{E[X]} \leq \exp\left(-\frac{\epsilon^2}{2}E[X]\right)$$

The following formula is well known.

Lemma 36 *Let* $X_1, \ldots, X_\ell$ *be independent Bernoulli random variables, with* $\Pr[X_i = 1] = \alpha$. *Let* $X = X_1 + \cdots + X_\ell$. *Then* $\Pr[X \equiv 1 \pmod 2] = \frac{1-(1-2\alpha)^\ell}{2}$.

Proof:

$$\begin{aligned}
\Pr[X \equiv 1 \pmod 2] &= \mathrm{E}\left[(1-(-1)^X)/2)\right] \\
&= \tfrac{1}{2} - \tfrac{1}{2}\,\mathrm{E}\left[\textstyle\prod_{i=1}^{\ell}(-1)^{X_i}\right] \\
&= \tfrac{1}{2} - \tfrac{1}{2}\textstyle\prod_{i=1}^{\ell}\mathrm{E}\left[(-1)^{X_i}\right] \\
&= \tfrac{1}{2} - \tfrac{1}{2}\textstyle\prod_{i=1}^{\ell}(1-2\alpha) \\
&= \tfrac{1-(1-2\alpha)^\ell}{2}
\end{aligned}$$

□

Proposition 37 *Let* $n \geq 1$ *be an integer and* S *be an arbitrary subset of* $\{1, \ldots, n\}$ *of size at most* $n/2$. *Let further* $\ell \geq 1$ *be an integer. Denote* $\alpha = |S|/n$. *Let* $X \in \{1, \ldots, n\}^\ell$ *be chosen uniformly at random.*

$$\frac{\min(1, \alpha\ell)}{4} < \Pr_X[|\{i;\ X_i \in S\}| \text{ is odd}] \leq \frac{1}{2}\ .$$

Proof: By the lemma we have

$$\Pr_X[|\{i;\ X_i \in S\}| \text{ is odd}] = \frac{1-(1-2\alpha)^\ell}{2}$$

We can bound

$$0 \leq (1-2\alpha)^\ell < e^{-2\alpha\ell}\ ,$$

where we use the estimate $1 - x < e^{-x}$ valid for all $x \neq 0$. If $\alpha\ell \leq 1$ then

$$(1-2\alpha)^\ell < e^{-2\alpha\ell} < e^{-\alpha\ell} < 1 - \alpha\ell/2\ .$$

using the estimate $e^{-x} < 1 - x/2$ valid for $0 < x \leq 1.59$. If $\alpha\ell > 1$ then

$$(1-2\alpha)^\ell < e^{-2\alpha\ell} < 1/4.$$

□

Lemma 9 *Let $X_1, \ldots, X_k$ be 0-1 random variables and $C, \alpha > 0$ be reals. Suppose that for every $i \in \{1, \ldots, k\}$, there are at most C indices $j \in \{1, \ldots, k\}$ such that X_i and X_j are not independent. Let $\mu = \mathrm{E}\left(\sum_{i=1}^{k} X_i\right)$. Then*

$$\Pr\left[\left|\sum_{i=1}^{k} X_i - \mu\right| \geq \alpha\mu\right] \leq \frac{C}{\alpha^2\mu}.$$

Proof: We need to estimate the variance.

$$\begin{aligned}\mathrm{Var}\left[\sum_i X_i\right] &= \mathrm{E}\left[\left(\sum_i X_i\right)^2\right] - \mu^2 \\ &= \sum_{i,j} \mathrm{E}\left[X_i X_j\right] - \mu^2\end{aligned}$$

Since any X_i and X_j are 0-1 random variables, $\mathrm{E}\left[X_i X_j\right] \leq \mathrm{E}\left[X_i\right]$. Furthermore, if X_i and X_j are independent then $\mathrm{E}\left[X_i X_j\right] = \mathrm{E}\left[X_i\right] \cdot \mathrm{E}\left[X_j\right]$. Since X_j's are non-negative, we get

$$\begin{aligned}\sum_{i,j} \mathrm{E}\left[X_i X_j\right] &\leq \sum_i \mathrm{E}\left[X_i\right] \left(C + \sum_{j,\, j \text{ indep. of } i} \mathrm{E}\left[X_j\right]\right) \\ &\leq \mu(C + \mu)\end{aligned}$$

Thus, $\mathrm{Var}\left[\sum_i X_i\right] \leq C\mu$. The lemma now follows from Chebyshev's Inequality.

□

7.2 Proof of Lemma 6

We will use induction on the number of vertices in V, where $V \subseteq W \setminus Y$ is the set of bad vertices for Y.. The statement is trivial if $|V| = 1$. First we prove the statement for $|V| = 2$. We will see later, that this is sufficient to prove the induction step.

Let ℓ be the largest number of vertex disjoint paths from X to Y. Let v_1 and v_2 be bad vertices for Y, that is the largest number of vertex disjoint paths from X to $Y \cup \{v_1\}$ and from X to $Y \cup \{v_2\}$ is also ℓ.

Suppose for a contradiction, that there are $\ell + 1$ vertex disjoint paths from X to $Y \cup \{v_1, v_2\}$. Fix a collection of $\ell + 1$ vertex disjoint paths from X to $Y \cup \{v_1, v_2\}$. Note that both v_1 and v_2 must participate in this collection. Let us denote the paths in the collection by $Q_1, \ldots, Q_{\ell-1}, R_1, R_2$, where v_1 is the endpoint of R_1 and v_2 is the endpoint of R_2. Fix also a collection of ℓ vertex disjoint paths from X to Y, call them $P_1, \ldots, P_\ell$.

We refer to the paths P_i as "solid" paths, and the paths Q_i as "dotted" paths. The paths R_1 and R_2 are "curly". Note that while the solid paths form a vertex disjoint collection, they can intersect the dotted and curly paths in various ways. Some solid paths may even

be identical to some dotted paths. It is also possible that the dotted collection of paths is a subcollection of the solid collection, but this does not have to be the case.

Next we note that since the largest number of vertex disjoint paths from X to $Y \cup \{v_1\}$ is ℓ, by Menger's theorem (Theorem 7), there exists a cut of size ℓ. That is, there are ℓ vertices, such that every path from X to $Y \cup \{v_1\}$ contains one of them. We will refer to this cut as the *blue cut*, and call the vertices in it *blue vertices*. Similarly, there are ℓ vertices, such that every path from X to $Y \cup \{v_2\}$ contains one of them. We will refer to this cut as the *red cut*, and call the vertices in it *red vertices*. Note that the blue cut and the red cut may intersect. We call the vertices that participate in both cuts *purple vertices*.

Next note that the solid paths $P_1, \ldots, P_\ell$ must be blocked by both cuts. Since the ℓ solid paths are vertex disjoint, this means that each solid path either contains one blue and one red vertex, or it contains one purple vertex. The collection $Q_1, \ldots, Q_{\ell-1}, R_1$ is blocked by the blue cut, thus each dotted path and the path R_1 contains exactly one vertex of the blue cut. Similarly, the collection $Q_1, \ldots, Q_{\ell-1}, R_2$ is blocked by the red cut, thus each dotted path and the path R_2 contains exactly one vertex of the red cut. Since $Q_1, \ldots, Q_{\ell-1}, R_1, R_2$ are vertex disjoint, the vertex of the blue cut on R_1 cannot participate in the red cut, so it cannot be purple. We call the vertex of the blue cut on R_1 the *blue star*. Similarly, the vertex of the red cut on R_2 cannot participate in the blue cut, so it cannot be purple. We call the vertex of the red cut on R_2 the *red star*. We also get that just like the solid paths, each dotted path either contains one blue and one red vertex, or it contains one purple vertex.

Since both the blue cut and the red cut consists of ℓ vertices, the above implies that the blue star must be at the intersection of R_1 with some solid path, and the red star must be at the intersection of R_2 with some solid path. All other vertices in the cuts (blue, red and purple) must be at the intersection of a solid path with a dotted path.

Recall that each solid path contains either a blue and a red vertex, or one purple vertex. Since the blue star cannot be purple, the solid path containing the blue star must also contain a red vertex. Similarly, the solid path containing the red star must also contain a blue vertex.

One of the consequences of the following two claims is that the blue star and the red star cannot be on the same solid path.

Claim 38 *Let P' from $x' \in X$ to $y' \in Y$ be the solid path that contains the blue star. Then the red vertex on P' must appear later in the path (that is closer to the output) than the blue star.*

Proof: Suppose that the red vertex on P' appears before the blue star. Suppose that the path R_1 starts at the vertex x_1, that is R_1 is a path from x_1 to v_1. Consider the path P^* that consists of the first part of R_1 from x_1 to the blue star, and the second part of P' from the blue star to y_1. Then, we can show that the path P^* does not contain any vertex of the red cut. This is however a contradiction, since P^* is a path from X to Y and the red cut blocks every path from X to $Y \cup \{v_2\}$. Recall that the red star is on the path R_2, and all other vertices of the red cut are at the intersection of some solid path with a dotted path. However, R_1 is vertex disjoint from R_2 as well as from all dotted paths. Thus, the part of P^* from R_1 cannot contain any vertex of the red cut. On the other hand, the second part

of P^* is part of the solid path P'. But P' contains only one vertex of the red cut, which appears before the blue star on the path P'. Thus, the common part of P' and P^* cannot contain any vertex from the red cut. Hence, P^* is not blocked by the red cut, which is a contradiction. □

We get the following analogous claim for the red star, by the same argument.

Claim 39 *Let P'' from $x'' \in X$ to $y'' \in Y$ be the solid path that contains the red star. Then the blue vertex on P'' must appear later in the path (that is closer to the output) than the red star.*

To continue the proof of the lemma, consider the following walk starting from the blue star. Recall that since the blue star cannot be purple, the solid path P' containing the blue star must also contain a red vertex, say r_1. We start the walk by following the segment of P' from the blue star to the red vertex r_1. By Claim 38, the red vertex r_1 appears after the blue star on the path P'. Thus, by Claim 39, the red vertex on P' cannot be the red star. Thus, r_1 is at the intersection of P' with some dotted path, say Q_1.

Note that while some solid paths may be identical to a dotted path, this is never the case for the solid paths containing the blue star and the red star, since R_1 and R_2 are vertex disjoint from all dotted paths. Thus, Q_1 is not identical to P'. Moreover, since Q_1 intersects P', and the solid paths are vertex disjoint, Q_1 cannot be identical to any solid path. By similar reasoning, the walk below never reaches a solid path that is identical to a dotted path.

Since r_1 is red (not purple), Q_1 must also contain a blue vertex, say b_1. We continue the walk by following the segment of Q_1 from r_1 to b_1. Note that b_1 may appear either before or after r_1 on the path Q_1, so we may be walking in the opposite direction (e.g. towards the inputs), in case b_1 appears before r_1 on the path Q_1. Note that since each dotted path is vertex disjoint from R_1, b_1 cannot be the blue star. On the other hand, b_1 is also at the intersection of Q_1 with some solid path P_1, that must be a different path than P' (since b_1 is not the blue star). P_1 must also contain a red vertex, say r_2 and we continue the walk by following the segment of P_1 from b_1 to r_2. Note that $r_2 \neq r_1$ since P' and P_1 are vertex disjoint. Unless we reached the red star, the red vertex r_2 is at the intersection of P_1 with a dotted path Q_2. Since $r_2 \neq r_1$, and each dotted path contains only one vertex from the red cut, we get that Q_2 is a different path than Q_1. Note that we can never run into a purple vertex, since we arrive to each new path at either a blue or a red vertex, and the paths that contain a blue or a red vertex never contain a purple vertex. Note also that along the walk, we always get to the next red vertex following a solid path. This guarantees that we do not revisit a previously visited red vertex, since the collection of solid paths is vertex disjoint. Similarly, we always get to the next blue vertex by following a dotted path. This guarantees that we do not revisit a previously visited blue vertex, since the collection of dotted paths is vertex disjoint. We never get back to the blue star, since the collection of dotted paths is vertex disjoint from R_1. Thus, we get a walk starting from the blue star, and passing through the vertices $r_1, b_1, r_2, b_2, \ldots$ until we reach the red star.

Claim 40 *In the walk constructed this way, we traverse each solid segment in the direction towards the outputs, and each dotted segment in the direction towards the inputs.*

Proof: Recall that by Claim 38, the red vertex r_1 appears after the blue star on the path P'. Thus, we start the walk by following the segment of P' from the blue star to r_1 in the direction towards the outputs.

Suppose that the claim does not hold. This means that there must be two consecutive segments during the walk, where we do not switch direction: if we switched direction between each segment of the walk, we would have to traverse each solid segment of the walk in the direction towards the outputs, and each dotted segment in the direction towards the inputs.

We will show that having two consecutive segments of the walk where we do not switch direction gives a contradiction. Suppose without loss of generality, that the segments from b_{i-1} to r_i and from r_i to b_i are both going towards the outputs. That is on the solid path P_{i-1}, r_i appears after b_{i-1}, and on the dotted path Q_i, b_i appears after r_i. (Note that possibly P_{i-1} could be P', in which case b_{i-1} would be the blue star.)

Consider the path P^* that consists of the first part of Q_i from some input vertex to r_i, and the second part of P_{i-1} from r_i to some output vertex. Then, we can show that the path P^* does not contain any vertex of the blue cut. This is however a contradiction, since P^* is a path from X to Y and the blue cut blocks every path from X to $Y \cup \{v_1\}$. Recall that each dotted path contains exactly one vertex of the blue cut. The blue vertex b_i appears on the dotted path Q_i after the vertex r_i, thus the first part of P^* that consists of the first part of Q_i from some input to r_i does not contain any vertex of the blue cut. Similarly, each solid path contains exactly one vertex of the blue cut. The blue vertex b_{i-1} appears on the solid path P_{i-1} before the vertex r_i, thus the second part of P^* that consists of the second part of P_{i-1} from r_i to some output does not contain any vertex of the blue cut. Hence, P^* is not blocked by the blue cut, which is a contradiction. □

Let b_k be the last blue vertex visited during the walk, before we reach the red star. Recall that by Claim 39 the blue vertex b_k must appear after the red star on the path P''. Thus, we finish the walk by following the segment of P'' between the red star and the vertex b_k in the reverse direction, from b_k to the red star, that is in the direction towards the inputs. This however contradicts Claim 40. This concludes the proof of the statement of the Lemma when $|V| = 2$.

Next we show that this also implies the statement for any number of bad vertices. We use induction on the size of V. We have already seen that the statement holds when $|V| = 1$ and when $|V| = 2$. Suppose the statement of the lemma holds for $|V| = j$, we show that this implies the statement for $|V| = j + 1$.

Let $Y \subseteq W$ be a set of outputs. Let $V = \{v_1, \ldots, v_j, v_{j+1}\}$ be a set of bad vertices for Y. Let ℓ be the largest number of vertex disjoint paths from X to Y. Let $Y' = Y \cup \{v_1, \ldots, v_{j-1}\}$. Since the statement of the lemma holds for $|V| \leq j$, we have that the largest number of vertex disjoint paths from X to Y' is ℓ, and that v_j and v_{j+1} must be bad for Y'. Applying the statement with two bad vertices for Y' implies that the largest number of vertex disjoint paths from X to $Y' \cup \{v_j, v_{j+1}\} = Y \cup V$ is also ℓ. This concludes the proof of the lemma.