On the Size of Depth-Three Boolean Circuits for Computing Multilinear Functions

Oded Goldreich* Avi Widgerson†

February 25, 2014

Abstract

We propose that multi-linear functions of relatively *low degree* over GF(2) may be good candidates for obtaining exponential[1] lower bounds on the size of constant-depth Boolean circuits (computing explicit functions). Specifically, we propose to move gradually from linear functions to multilinear ones, and conjecture that, for any $t \geq 2$, some explicit t-linear functions $F : (\{0,1\}^n)^t \to \{0,1\}$ require depth-three circuits of size $\exp(\Omega(tn^{t/(t+1)}))$.

Towards studying this conjecture, we suggest to study two frameworks for the design of *depth-three* Boolean circuits computing multilinear functions, yielding restricted models for which lower bounds may be easier to prove. Both correspond to constructing a circuit by expressing the target polynomial as a composition of simpler polynomials. The first framework corresponds to a direct composition, whereas the second (and stronger) framework corresponds to nested composition and yields depth-three Boolean circuits via a "guess-and-verify" paradigm. The corresponding restricted models of circuits are called D-canonical and ND-canonical, respectively.

Our main results are (1) a generic upper bound on the size of depth-three D-canonical circuits for computing any t-linear function, and (2) a lower bound on the size of any depth-three ND-canonical circuits for computing some (in fact, almost all) t-linear functions. These bounds match the foregoing conjecture (i.e., they have the form of $\exp(tn^{t/(t+1)})$). Another important result is a separation of the two models: We prove that ND-canonical circuits can be super-polynomially smaller than their D-canonical counterparts. We also reduce proving lower bounds for the ND-model to Valiant's matrix rigidity problem (for parameters that were not the focus of previous works).

The study of the foregoing (Boolean) models calls for an understanding of new types of arithmetic circuits, which we define in this paper and may be of independent interest. These circuits compute multilinear polynomials by using *arbitrary* multilinear gates of some limited arity. It turns out that a GF(2)-polynomial is computable by such circuits with at most s gates of arity at most s if and only if it can be computed by ND-canonical circuits of size $\exp(s)$. A similar characterization holds for D-canonical circuits if we further restrict the arithmetic circuits to have depth two. We note that the new arithmetic model makes sense over any field, and indeed all our results carry through to all fields. Moreover, it raises natural arithmetic complexity problems which are independent of our original motivation.

Keywords: Constant-depth Boolean circuits, depth-three Boolean circuits, arithmetic circuits, circuit lower bounds, multilinear functions, multilinear circuits, high-order tensors, matrix rigidity.

*Research performed when visiting the IAS. Partially supported by the Israel Science Foundation (grant No. 1041/08) and by the Minerva Foundation (with funds from the Federal German Ministry for Education and Research). Address: Department of Computer Science, Weizmann Institute of Science, Rehovot, Israel. oded.goldreich@weizmann.ac.il

†School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540, USA. avi@ias.edu

[1]Throughout this paper, when we say that a function f is exponential, we mean that $f(n) = \exp(\Theta(n))$.

An alternative summary

This paper introduces and initiates a study of a new model of arithmetic circuits coupled with new complexity measures. The new model consists of multilinear circuits *with arbitrary multilinear gates*, rather than the standard multilinear circuits that use only addition and multiplication gates. In light of this generalization, the *arity of gates* becomes of crucial importance and is indeed one of our complexity measures. Our second complexity measure is the *number of gates* in the circuit, which (in our context) is significantly different from the number of wires in the circuit (which is typically used as a measure of size). Our main *complexity measure*, denoted $\mathtt{C}(\cdot)$, is the maximum of these two measures (i.e., the maximum between the arity of the gates and the number of gates in the circuit). We also consider the depth of such circuits, focusing on depth-two and unbounded depth.

Our initial motivation for the study of this arithmetic model is the fact that the two main variants (i.e., depth-two and unbounded depth) yield natural classes of depth-three Boolean circuits for computing multi-linear functions. The resulting circuits have size that is exponential in the new complexity measure. Hence, lower bounds on the new complexity measure yield lower bounds on a restricted class of depth-three Boolean circuits (for computing multi-linear functions). Such lower bounds are a sanity check for our conjecture that multi-linear functions of relatively *low degree* over GF(2) are good candidates for obtaining exponential lower bounds on the size of constant-depth Boolean circuits (computing explicit functions). Specifically, we propose to move gradually from linear functions to multilinear ones, and conjecture that, for any $t \geq 2$, some explicit t-linear functions $F : (\{0,1\}^n)^t \to \{0,1\}$ require depth-three circuits of size $\exp(\Omega(tn^{t/(t+1)}))$.

Letting $\mathtt{C}_2(\cdot)$ denote the complexity measure $\mathtt{C}(\cdot)$, when minimized over all depth-two circuits of the above type, our main results are as follows.

- For every t-linear function F, it holds that $\mathtt{C}(F) \leq \mathtt{C}_2(F) = O((tn)^{t/(t+1)})$.
- For almost all t-linear function F, it holds that $\mathtt{C}_2(F) \geq \mathtt{C}(F) = \Omega((tn)^{t/(t+1)})$.
- There exists a bilinear function F such that $\mathtt{C}(F) = O(\sqrt{n})$ but $\mathtt{C}_2(F) = \Omega(n^{2/3})$.

The main open problem posed in this paper is proving a result analogous to (2) for an explicit function F. For starters, we seek lower bound of $\Omega((tn)^{0.51})$ for an explicit t-linear function F, preferably for constant t. We outline an approach that reduces this challenge (for $t = 3$) to a question regarding matrix rigidity.

Organization. The introduction contains an extensive motivation for the model of arithmetic circuits that is studied in the paper. Readers who are only interested in the model, may skip the introduction with little harm, except for the definition of three specific functions that appear (in displayed equations) towards the end of Section 1.1. See Section 1.7 for notes regarding material that was added after the first posting.

Contents

1 Introduction

Strong lower bounds on the size of constant-depth Boolean circuits computing parity and other explicit functions (cf., e.g., [30, 8] and [22, 25]) are among the most celebrated results of complexity theory. These quite tight bounds are all of the form $\exp(n^{1/(d-1)})$, where n denote the input length and d the circuit depth. But we do not know of any exponential lower bounds (i.e., of the form $\exp(\Omega(n))$) on the size of constant-depth circuits computing any explicit function (i.e., a Boolean function in $\mathcal{E} = \cup_{c\in\mathbb{N}}\mathrm{Dtime}(f_c)$, where $f_c(n) = 2^{cn}$).

Providing exponential lower bounds on the size of constant-depth Boolean circuits computing explicit functions is a central problem of circuit complexity, even when restricting attention to depth-three circuits (cf., e.g., [12, Chap. 11]). It seems that such lower bounds cannot be obtained by the standard interpretation of either the random restriction method [6, 8, 30] or the approximation by polynomials method [22, 25]. Many experts have tried other approaches (cf., e.g., [10, 13][2]), and some obtained encouraging indications (i.e., results that refer to restricted models, cf., e.g., [19]); but the problem remains wide open.

There are many motivations for seeking exponential lower-bounds for constant-depth circuits. Two notable examples are separating $\mathcal{NL}$ from $\mathcal{P}$ (see Appendix A) and presenting an explicit function that does not have linear-size circuits of logarithmic depth (see Valiant [28]). Another motivation is the derandomization of various computations that are related to $\mathcal{AC}_0$ circuits (e.g., approximating the number of satisfying assignments to such circuits). Such derandomizations can be obtained via "canonical derandomizers" (cf. [7, Sec. 8.3]), which in turn can be constructed based on strong average-case versions of circuit lower bounds; cf. [17, 18].

It seems that the first step should be beating the $\exp(\sqrt{n})$ size lower bound for depth-three Boolean circuits computing explicit functions (on n bits). A next step may be to obtain a truly exponential lower bound for depth-three Boolean circuits, and yet another one may be to move to any constant depth.

This paper focuses on the first two steps; that is, it focuses on depth-three circuits. Furthermore, within that confined context, we focus on a restricted type of circuits, which emerges rather naturally from the class of functions that we propose to study.

1.1 The candidate functions

We suggest to study specific *multilinear functions of relatively low degree* over the binary field, GF(2), and in the sequel all arithmetic operations are over this field. For $t, n \in \mathbb{N}$, we consider *t-linear* functions of the form $F : (\{0,1\}^n)^t \to \{0,1\}$, where F is linear in each of the t blocks of variables (which contain n variables each). Such a function F is associated with a t-dimensional array, called a tensor, $T \subseteq [n]^t$ such that

$$F(x^{(1)}, x^{(2)}, ..., x^{(t)}) = \sum_{(i_1,i_2,...,i_t)\in T} x^{(1)}_{i_1} x^{(2)}_{i_2} \cdots x^{(t)}_{i_t} \tag{1}$$

where here and throughout this paper $x^{(j)} = (x^{(j)}_1, ..., x^{(j)}_n) \in \{0,1\}^n$ for every $j \in [t]$. Indeed, we refer to a fixed partition of the Boolean variables to t blocks, each containing n variables, and to functions that are linear in the variables of each block. Such functions were called set-multilinear in [19]. Note that the input length for these functions is $t \cdot n$; hence, *exponential lower bounds mean bounds of the form* $\exp(\Omega(tn))$.

We will start with a focus on constant t, and at times we will also consider t to be a function of n, but n will always remain the main length parameter. Actually, it turns out that $t = t(n) = \Omega(\log n)$ is essential for obtaining exponential lower bounds (i.e., size lower bounds of the form $\exp(\Omega(tn))$ for depth-d circuits, when $d > 2$).

A good question to ask is whether there exists any multilinear function that requires constant-depth Boolean circuit of exponential size (i.e., size $\exp(\Omega(tn))$). We conjecture that the answer is positive.

Conjecture 1.1 (a sanity check for the entire approach): *For every $d > 2$, there exist t-linear functions $F : (\{0,1\}^n)^t \to \{0,1\}$ that cannot be computed by Boolean circuits of depth d and size $\exp(o(tn))$, where $t = t(n) \leq \mathrm{poly}(n)$.*

[2] The relevance of the Karchmer and Wigderson approach [13] to constant-depth circuits is stated explicitly in [14, Sec. 10.5].

We believe that the conjecture holds even for $t = t(n) = O(\log n)$, and note that, for any fixed t, there exist explicit t-linear functions that cannot be computed by depth-two circuits of size $2^{tn/4}$ (see Appendix C.3).

Merely proving Conjecture 1.1 may not necessarily yield a major breakthrough in the state-of-art regarding lower bounds, although it *seems* that a proof will need to do something more interesting than mere counting. However, disproving Conjecture 1.1 will cast a shadow on our suggestions, which may nevertheless maintain their potential for surpassing the $\exp((tn)^{1/(d-1)})$ barrier.[3]

Assuming that Conjecture 1.1 holds, one should ask which explicit functions may "enjoy" such lower bounds. Two obviously bad choices are (1) $F^{t,n}_{\mathtt{all}}(x^{(1)},...,x^{(t)}) = \sum_{i_1,...,i_t\in[n]} x^{(1)}_{i_1}\cdots x^{(t)}_{i_t}$ and (2) $F^{t,n}_{\mathtt{diag}}(x^{(1)},...,x^{(t)}) = \sum_{i\in[n]} x^{(1)}_{i}\cdots x^{(t)}_{i}$, since each is easily reducible to an n-way parity (the lower bounds for which we wish to surpass).[4] The same holds for any function that corresponds either to a rectangular tensor (i.e., $I_1\times\cdots\times I_t$, where $I_1,..,I_t\subseteq[n]$) or to a sparse tensor (e.g., $T\subseteq[n]^t$ such that $|T| = O(n)$). Ditto w.r.t the sum of few such tensors. Indeed, one should seek tensors $T\subseteq[n]^t$ that are far from the sum of few rectangular tensors (i.e., far from any tensor of low rank [26]). On the other hand, it seems good to stick to as "simple" tensors as possible so as to facilitate their analysis (let alone have the corresponding multilinear function be computable in exponential-time (i.e., in $\mathcal{E}$)).[5]

A less obvious bad choice. Consider the function $F^{t,n}_{\mathtt{leq}}:(\{0,1\}^n)^t\to\{0,1\}$ such that

$$F^{t,n}_{\mathtt{leq}}(x^{(1)},x^{(2)},...,x^{(t)}) = \sum_{1\le i_1\le i_2\le\cdots\le i_t\le n} x^{(1)}_{i_1}x^{(2)}_{i_2}\cdots x^{(t)}_{i_t} \tag{2}$$

(having the corresponding tensor $T^{t,n}_{\mathtt{leq}} = \{(i_1,...,i_t)\in[n]^t : i_1\le i_2\le\cdots\le i_t\}$). Note that this function is polynomial-time computable (e.g., via dynamic programming),[6] and that $t=1$ corresponds to `Parity`. Unfortunately, for every constant $t\ge 2$, the function $F^{t,n}_{\mathtt{leq}}$ is not harder than parity: It has depth-three circuits of size $\exp(O(\sqrt{n}))$; see Proposition 3.4. Thus, we move the slightly less simple candidates presented next.

Specific candidates. We suggest to consider the following t-linear functions, $F^{t,n}_{\mathtt{tet}}$ and $F^{t,n}_{\mathrm{mod}\,p}$ (especially for $p\approx 2^t\approx n$), which are presented next in terms of their corresponding tensors (i.e., $T^{t,n}_{\mathtt{tet}}$ and $T^{t,n}_{\mathrm{mod}\,p}$, resp).

$$T^{t,n}_{\mathtt{tet}} = \left\{(i_1,...,i_t)\in[n]^t : \sum_{j\in[n]}|i_j-(n/2)|\le n/2\right\} \tag{3}$$

$$T^{t,n}_{\mathrm{mod}\,p} = \left\{(i_1,...,i_t)\in[n]^t : \sum_{j\in[t]} i_j\equiv 0 \pmod p\right\} \tag{4}$$

Note that these functions are also computable in polynomial-time.[7] For $p<n$, it holds that $F^{t,n}_{\mathrm{mod}\,p}(x^{(1)},...,x^{(t)})$ equals $F^{t,p}_{\mathrm{mod}\,p}(y^{(1)},...,y^{(t)})$, where $y^{(j)}_r = \sum_{i\in[n]:i\equiv r \pmod p} x^{(j)}_i$ for every $j\in[t]$ and $r\in[p]$. This reduction

[3] Showing an upper bound of the form $\exp((tn)^{1/(d-1)})$ on the size circuits of depth d that compute any t-linear function seems unlikely (cf. [19], which proves an exponential in t lower bound on the size of depth-three arithmetic circuits).

[4] Note that $F^{t,n}_{\mathtt{all}}(x^{(1)},...,x^{(t)}) = \prod_{j\in[t]}\sum_{i_j\in[n]} x^{(j)}_{i_j}$, which means that it can be computed by a t-way conjunction of n-way parity circuits, whereas $F^{t,n}_{\mathtt{diag}}$ is obviously an n-way parity of t-way conjunctions of variables.

[5] Thus, these tensors should be constructible within $\exp(tn)$-time. Note that we can move from the tensor to the multilinear function (and vice versa) in $n^t << \exp(tn)$ oracle calls.

[6] Note that $F^{t,n}_{\mathtt{leq}}(x^{(1)},...,x^{(t)})$ equals $\sum_{i\in[n]} F^{t-1,i}_{\mathtt{leq}}(x^{(1)}_{[1,i]},...,x^{(t-1)}_{[1,i]})\cdot x^{(t)}_i$, where $x^{(j)}_{[1,i]} = (x^{(j)}_1,...,x^{(j)}_i)$. So, for every $t'\in[t-1]$, the dynamic program uses the n values $(F^{t',i}_{\mathtt{leq}}(x^{(1)}_{[1,i]},...,x^{(t')}_{[1,i]}))_{i\in[n]}$ in order to compute the n values $(F^{t'+1,i}_{\mathtt{leq}}(x^{(1)}_{[1,i]},...,x^{(t'+1)}_{[1,i]}))_{i\in[n]}$.

[7] Again, we use dynamic programming, but here we apply it to generalizations of these functions. Specifically, let $T^{t,n,d}_{\mathtt{tet}} = \{(i_1,...,i_t)\in[n]^t : \sum_{j\in[n]}|i_j-(n/2)|\le d\}$ and note that the associated function

may have a forbidding "size cost" in the context of circuits of a specific depth (especially if $p \ll n$), but its cost is insignificant if we are willing to double the depth of the circuit (and aim at lower bounds that are larger than those that hold for parity). Thus, in the latter cases, we may assume that $p = \Omega(n)$, but of course $p < tn$ must always hold.

We note that none of the bilinear versions of the foregoing functions can serve for beating the $\exp(\sqrt{n})$ lower bound. Specifically, the failure of $F^{2,n}_{\mathrm{mod}\,p}$ is related to the aforementioned reduction, whereas the failure of $F^{2,n}_{\mathtt{tet}}$ is related to the fact that boundary of its tensor has linear size (just as in the case of $F^{2,n}_{\mathtt{leq}}$). But these weaknesses do not seem to propagate to the trilinear versions. (In contrast, the function $F^{t,n}_{\mathtt{leq}}$ fails also for higher values of t, since the boundary of $T^{t,n}_{\mathtt{leq}}$ can be "decomposed" into a constant number of lower-dimensional tensors. But this does not seem to be the case for $F^{t,n}_{\mathtt{tet}}$.)

What's next? In an attempt to study the viability of our suggestions and conjectures, we defined two restricted classes of depth-three circuits and tried to prove lower bounds on the sizes of circuits from these classes that compute the foregoing functions. Our success in proving lower bounds was very partial, and will be discussed next – as part of the discussion of these two classes (in Sections 1.2 and 1.3).

1.2 Design from direct composition: the D-canonical model

What is a natural way of designing depth-three Boolean circuits that compute multilinear functions?

Let us take our cue from the linear case (i.e., $t = 1$). The standard way of obtaining a depth-three circuit of size $\exp(\sqrt{n})$ for n-way parity is to express this linear function as the $\sqrt{n}$-way sum of $\sqrt{n}$-ary functions that are linear in disjoint sets of variables. The final (depth-three) circuit is obtained by combing the depth-two circuit for the outer sum with the depth-two circuits computing the $\sqrt{n}$ internal sums.

Hence, a natural design strategy is to express the target multilinear function (F) as a polynomial (H) in some auxiliary multilinear functions (F_i's), and combine depth-two circuits that compute the auxiliary multilinear functions with a depth-two circuit that computes the main polynomial (i.e., H). That is, we "decompose" the multilinear function on the algebraic level, expressing it as a polynomial in auxiliary multilinear functions (i.e., $F = H(F_1, ..., F_s)$), and implement this decomposition on the Boolean level (i.e., each polynomial is implemented by a depth-two Boolean circuit). Specifically, to design a depth-three circuit of size $\exp(O(s))$ for computing a multilinear function F the following steps are taken:

1. Select s arbitrary multilinear functions, $F_1, ..., F_s$, each depending on s input bits;
2. Express F as a polynomial H in the F_i's;
3. Obtain a depth-three circuit by combining depth-two circuits for computing H and the F_i's.

Furthermore, we mandate that $H(F_1, ..., F_s)$ is a formal multilinear function; that is, the monomials of H do not multiply two F_i's that depend on the same block of variables. The size of the resulting circuit is taken to be $\exp(\Theta(s))$: The upper bound is justified by the construction, and the lower bound by the assumption that (low degree) polynomials that depend on s variables require depth-two circuits of $\exp(s)$ size. (The latter assumption is further discussed in Section 2.2.)[8]

Circuits that are obtained by following this framework are called D-canonical, where "D" stands for *direct* (or *deterministic*, for reasons that will become apparent in Section 1.3). D-canonical circuits seem natural in the context of computing multi-linear functions by depth-three Boolean circuits.

For example, the standard design, reviewed above, of depth-three circuits (of size $\exp(\sqrt{n})$) for (n-way) parity yields D-canonical circuits. In general, D-canonical circuits for a target multilinear function

satisfies $F^{t,n,d}_{\mathtt{tet}}(x^{(1)}, ..., x^{(t)}) = \sum_{i\in[n]} F^{t-1,n,d-i}_{\mathtt{tet}}(x^{(1)}, ..., x^{(t-1)}) \cdot x^{(t)}_i$. Likewise, consider the tensor $T^{t,n,r}_{\mathrm{mod}\,p} = \left\{(i_1, ..., i_t) \in [n]^t : \sum_{j\in[t]} i_j \equiv r \pmod p\right\}$ and note that the associated function satisfies $F^{t,n,r}_{\mathrm{mod}\,p}(x^{(1)}, ..., x^{(t)}) = \sum_{i\in[n]} F^{t-1,n,r-i}_{\mathrm{mod}\,p}(x^{(1)}, ..., x^{(t-1)}) \cdot x^{(t)}_i$.

[8] In brief, when computing t-linear polynomials, a lower bound of $\exp(\Omega(s/2^t))$ on the size of depth-two circuits can be justified (see Appendix C). Furthermore, for $2^t \ll s$, a lower bound of $\exp(\Omega(s))$ can be justified if the CNFs (or DNFs) used are "canonical" (i.e., use only s-way gates at the second level).

are obtained by combining depth-two circuits that compute auxiliary multilinear functions with a depth-two circuit that computes the function that expresses the target in terms of the auxiliary functions. The freedom of the framework (or the circuit designer) is reflected in the choice of auxiliary functions, whereas the restriction is in insisting that the target multilinear functions be computed by composition of a polynomial and multilinear functions (and that this composition corresponds to a formal multilinear function).

Our main results regarding D-canonical circuits are a generic upper bound on the size of D-canonical circuits computing any t-linear function and a matching lower bound that refers to almost all t-linear functions. That is:

Theorem 3.1: *For every $t \geq 2$, every t-linear function $F : (\{0,1\}^n)^t \to \{0,1\}$ can be computed by D-canonical circuits of size* $\exp((tn)^{t/(t+1)})$.

(Corollary to) Theorem 4.1: *For every $t \geq 2$, almost all t-linear functions $F : (\{0,1\}^n)^t \to \{0,1\}$ require D-canonical circuits of size at least* $\exp(\Omega(tn)^{t/(t+1)})$.

Needless to say, the begging question is what happens with explicit multilinear functions.

Problem 1.2 (main problem regarding D-canonical circuits): *For every $t \geq 2$, prove a* $\exp(\Omega(tn)^{t/(t+1)})$ *lower bound on the size of D-canonical circuits computing some explicit function. Ditto when t may vary with n, but $t \leq \mathrm{poly}(n)$.*

Of course, at this time, it would be interesting to obtain any lower bound that goes beyond the $\exp(\sqrt{tn})$ barrier. As mentioned in Section 1.1, for every $t \geq 2$, the function $F^{t,n}_{\mathtt{leq}}$ cannot be used towards that goal: By Proposition 3.4, $F^{t,n}_{\mathtt{leq}}$ *has D-canonical circuits of size* $\exp(O(\sqrt{n}))$. In contrast, $F^{t,n}_{\mathtt{tet}}$ seems quite promising (see Section 4.2).

We comment that we obtained the $\exp(O(\sqrt{n}))$-sized D-canonical circuits for $F^{t,n}_{\mathtt{leq}}$ by realizing that $F^{t,n}_{\mathtt{leq}}$ has linear-size circuits of logarithmic depth (i.e., it is simple in the sense of Valiant [28]), and thus it must have subexponential size depth-three circuits (cf. [28]). Reverse-engineering Valiant's argument, as applied to $F^{t,n}_{\mathtt{leq}}$, and optimizing the design, we arrived at the current proofs, which are presented (in Section 3.2) in a self-contained manner (without mentioning Valiant's method).

1.3 Design from nested composition: the ND-canonical model

As appealing as D-canonical circuits may appear, it turns out that one can build significantly smaller circuits by employing the "guess and verify" technique. This allows to express the target function in terms of auxiliary functions, which themselves are expressed in terms of other auxiliary functions, and so on. That is, the "expression depth" is no longer 1, it is even not a priori bounded, and yet the resulting circuit has depth-three.

The basic idea is to use s non-deterministic guesses for the values of s auxiliary functions, and to verify each of these guesses based on (some of) the other guesses and at most s bits of the original input. Thus, the verification amounts to the conjunction of s conditions, where each condition depends on at most $2s$ bits (and can thus be verified by a CNF of size $\exp(2s)$). The final depth-three circuit is obtained by replacing the s non-deterministic guesses by a 2^s-way disjunction.

This way of designing depth-three circuits leads to a corresponding framework, and the circuits obtained by it are called ND-canonical, where "ND" stands for *non-determinism*. In this framework depth-three circuits of size $\exp(O(s))$ for computing a multilinear function F are designed by the following three-step process:

1. Select s auxiliary multi-linear functions, $F_1, ..., F_s$;
2. Express F as well as each of the other F_i as a polynomial in the subsequent F_i's and in at most s input bits;
3. Obtain a depth-three circuit by combining depth-two circuits for computing these polynomials, where the combination implements s non-deterministic choices as outlined above.

As in the D-canonical framework, the polynomials used in Step (2) should be such that replacing the functions F_i's in them yields multilinear functions (i.e., this is a syntactic condition). Again, the size of the resulting circuit is taken to be $\exp(\Theta(s))$.

Note that, here (in the case of ND-canonical circuits), the combination performed in Step (3) is not a functional composition (as in the case of the D-canonical circuits). It is rather a verification of the claim that there exists $s+1$ values that fit all $s+1$ expressions (i.e., of F and the F_i's). The implementation of Step (3) calls for taking the conjunction of these $s+1$ depth-two computations as well as taking a 2^{s+1}-way disjunction over all possible values that these computations may yield.

The framework of ND-canonical circuits allows to express F in terms of F_i's that are themselves expressed in terms of F_j's, and so on. In contrast, in the D-canonical framework, the F_i's were each expressed in terms of s input bits. A natural question is whether this generalization actually helps. We show that the answer is positive.

Theorem 2.3: There exists bilinear functions $F : (\{0,1\}^n)^2 \to \{0,1\}$ that have ND-circuits of size $\exp(O(\sqrt{n}))$ but no D-circuits of size $\exp(o(n^{2/3}))$.

Turning to our results regarding ND-circuits, the upper bound on D-canonical circuits clearly holds for ND-circuits, whereas our lower bound is actually established for ND-canonical circuits (and the result for D-canonical circuits is a corollary). Thus, we have

(Corollary to) Theorem 3.1: *For every $t \geq 2$, every t-linear function $F : (\{0,1\}^n)^t \to \{0,1\}$ can be computed by ND-canonical circuits of size* $\exp((tn)^{t/(t+1)})$.

Theorem 4.1: *For every $t \geq 2$, almost all t-linear functions $F : (\{0,1\}^n)^t \to \{0,1\}$ require ND-canonical circuits of size at least* $\exp(\Omega(tn)^{t/(t+1)})$.

Again, the real challenge is to obtain such a lower bound for explicit multilinear functions.

Problem 1.3 (main problem regarding ND-canonical circuits): *For every $t \geq 2$, prove a* $\exp(\Omega(tn)^{t/(t+1)})$ *lower bound on the size of ND-canonical circuits computing some explicit function. Ditto when t may vary with n, but $t \leq \mathrm{poly}(n)$.*

For starters, *prove a* $\exp(\Omega(tn)^{0.51})$ *lower bound on the size of ND-canonical circuits computing some explicit t-linear function.*

As a possible step towards this goal we reduce the task of proving such a lower bound for $F^{3,n}_{\mathtt{tet}}$ to proving a lower bound on the rigidity of matrices with parameters that were not considered before. In particular, an $\exp(\omega(\sqrt{n}))$ lower bound on the size of ND-canonical circuits computing $F^{3,n}_{\mathtt{tet}}$ will follow from the existence of an n-by-n Toeplitz matrix that has rigidity $\omega(n^{3/2})$ with respect to rank $\omega(n^{1/2})$. For more details, see Section 4.2 (as well as Section 4.3).

1.4 An arithmetic circuit complexity perspective

The two models of canonical (depth-three) Boolean circuits are rooted in and correspond to two models of arithmetic circuits (for computing multilinear functions). In both arithmetic models, a (multilinear) function F is computed by composing auxiliary (multilinear) functions and variables of F. The D-canonical circuits are obtained by a straightforward implementation of some direct composition (i.e., $F = H(F_1, ..., F_s)$, where each F_i depends on at most s variables of F). The ND-canonical circuits are obtained by a Valiant-like (i.e., akin [28]) implementation of some general nested composition of auxiliary functions and variables; that is, guessing and verifying the values of all auxiliary functions, where each auxiliary function is expressed in terms of F's variables and subsequent auxiliary functions. In either case, the parameter that determines the size of the resulting Boolean circuit is the maximum between the number of auxiliary functions and the number of variables that appear explicitly in each auxiliary function. This parameter restricts the power of the underlying arithmetic circuits or rather serves as their complexity measure. Let us spell out these two models of arithmetic circuit complexity.

The arithmetic circuits we refer to arise when viewing the foregoing auxiliary functions as gates that compute arbitrary multilinear functions of their arguments, which correspond to other auxiliary functions

and/or input variables (such that arguments that depend on variables in the same block are not multiplied by such gates). The aforementioned parameter corresponds to the maximum between the number of gates and the arity of these gates.[9] Direct composition corresponds to depth-two arithmetic circuits (with such general gates), where the target function corresponds to the top gate and the auxiliary functions correspond to the gates that feed into the top gate. Nested composition corresponds to arithmetic circuits (with such general gates) of arbitrary depth, where gates may feed into gates that are not necessarily the top gate. More specifically:

- Following [19], we say that an arithmetic circuit is multilinear if its input variables are partitioned into blocks and the gates of the circuit compute multilinear functions such that if two gates have directed paths from the same block of variables then the results of these two gates are not multiplied together.
- We say that the direct-composition complexity of F, denoted $\mathsf{C}_2(F)$, is at most s if F *can be computed by a depth-two multi-linear circuit with at most s gates that are each of arity at most s.*
- We say that the nested-composition complexity of F, denoted $\mathsf{C}(F)$, is at most s if F *can be computed by a multi-linear circuit with at most s gates that are each of arity at most s.*

We stress that the multilinear circuits in the foregoing definition employ arbitrary multilinear gates, whereas in the standard arithmetic model the gates correspond to either (unbounded) addition or multiplication. Our complexity measure is related to but different from circuit size: On the one hand, we only count the number of gates (and discard the number of leaves, which in our setting may be larger). On the other hand, our complexity measure also bounds the arity of the gates.

Note that for any *linear* function F, it holds that $\mathsf{C}_2(F) = \Theta(\mathsf{C}(F))$, because all intermediate gates can feed directly to the top gate (since, in this case, all gates compute linear functions).[10] Also note that $\mathsf{C}_2(F)$ equals the square root of the number of variables on which the linear function F depends. In general, $\mathsf{C}(F) \geq \sqrt{tn}$ for any t-linear function F that depends on all its variables, and $\mathsf{C}(F) \leq \mathsf{C}_2(F) \leq tn$ for any t-linear function F. Thus, our complexity measures (for non-degenerate t-linear functions) range between $\sqrt{tn}$ and tn.

Clearly, F has a D-canonical (resp., ND-canonical) circuit of size $\exp(\Theta(s))$ if and only if $\mathsf{C}_2(F) = s$ (resp., $\mathsf{C}(F) = s$). Thus, all results and open problems presented above (i.e., in Sections 1.2 and 1.3) in terms of canonical (Boolean) circuits are actually results and open problems regarding the complexity of (direct and nested) composition (i.e., $\mathsf{C}_2(\cdot)$ and $\mathsf{C}(\cdot)$). Furthermore, the results are actually proved by analyzing these complexity measures. Specifically, we have:

Thm. 3.1: For every t-linear function F, it holds that $\mathsf{C}(F) \leq \mathsf{C}_2(F) = O((tn)^{t/(t+1)})$.

Thm. 4.1: For almost all t-linear function F, it holds that $\mathsf{C}_2(F) \geq \mathsf{C}(F) = \Omega((tn)^{t/(t+1)})$.

Thm. 2.3: There exists a bilinear function F such that $\mathsf{C}(F) = O(\sqrt{n})$ but $\mathsf{C}_2(F) = \Omega(n^{2/3})$.

We stress that the foregoing lower bounds are existential, whereas we seek $\omega(\sqrt{n})$ lower bounds for explicit multilinear functions.

Hence, this paper introduces and initiates a study of a new model of arithmetic circuits and accompanying new complexity measures. The new model consists of multilinear circuits *with arbitrary multilinear gates*, rather than the standard multilinear circuits that use only addition and multiplication gates. In light of this generalization, the *arity of gates* becomes of crucial importance and is indeed one of our complexity measures. Our second complexity measure is the *number of gates* in the circuit, which (in our context) is significantly different from the number of wires in the circuit (which is typically used as a measure of size).

[9] There is a small discrepancy between the parameter as defined in the prior paragraph and the way it is defined here: In the prior definition we only bounded the number of leaves (variables) that feed into each gate, while the number of non-leaves that feed a gate is bounded by the total number of gates. Thus, the arity of the gate (as defined here) is at most twice the value defined before. Also, our current gate count also counts the top gate, whereas it was not counted before. On the other hand, when defining direct composition complexity before, we did not allow the top gate to have leaves, but this can be fixed by adding dummy gates that take a single leaf each.

[10] A more general argument is presented in Remark 2.4, which asserts that if gate G computes a monomial that contains no leaves, then this monomial can be moved up to the parent of G.

Our main complexity measure is the maximum of these two measures (i.e., the maximum between the arity of the gates and the number of gates in the circuit). Our initial motivation for the study of this arithmetic model is its close relation to canonical Boolean circuits, and from this perspective depth-two arithmetic circuits have a special appeal.

A natural question is whether our complexity measure (i.e., $\mathtt{C}$) decreases if one waives the requirement that the arithmetic circuit be a multilinear one (i.e., the gates compute multilinear functions and they never multiply the outcomes of gates that depend on the same block of variables). The answer is that waiving this restriction in the computation of any t-linear function may decrease the complexity by at most a factor of 2^t (see Remark 2.5).

We note that the arithmetic models discuss above make sense with respect to any field. The reader may verify that all results stated for $\mathtt{C}_2(\cdot)$ and $\mathtt{C}(\cdot)$ hold for every field, rather than merely for the binary field. Ditto for the open problems.

1.5 Related work

Multilinear functions were studied in a variety of models, mostly in the context of algebraic and arithmetic complexity. In particular, Nisan and Wigderson [19] initiated a study of *multilinear circuits* as a natural model for the computation of multilinear functions. Furthermore, they obtained an exponential (in t) lower bound on the size of depth-three multilinear circuits that compute a natural t-linear function (i.e., iterated matrix multiplication for 2-by-2 matrices).[11]

The multilinear circuit model was studied in subsequent works (cf., e.g., [21]); but, to the best of our knowledge, the complexity measure introduced in Section 1.4 was not studied before. Nevertheless, it may be the case that techniques and ideas developed in the context of the multilinear circuit model will be useful for the study of this new complexity measure (and, equivalently, in the study of canonical circuits). For example, it seems that the latter study requires a good understanding of tensors, which were previously studied with focus at a different type of questions (cf., e.g., [20]).

In the following two paragraphs we contrast our model of multilinear circuits, which refers to arbitrary gates of arity that is reflected in our complexity measure, with the standard model of multilinear circuits [19], which uses only addition and multiplication gates (of unbounded arity). For the sake of clarity, we shall refer to canonical circuits rather than to our model of multilinear circuits, while reminding the reader that the two are closely related.

The difference between the standard model of constant-depth *multilinear circuit* and the model of constant-depth Boolean circuits is rooted in the fact that the (standard) *multilinear circuit* model contains unbounded fan-in addition gates as basic components, whereas unbounded fan-in addition is hard for constant-depth Boolean circuits. Furthermore, the very fact that n-way addition requires $\exp(n)$-size depth-two Boolean circuits is the basis of the approach that we are suggesting here. In contrast, hardness in the multilinear circuit model is related to the total degree of the function to be computed.[12]

The foregoing difference is reflected in the contrast between the following two facts: (1) multilinear functions of low degree have small depth-two *multilinear circuits* (i.e., each t-linear function $F : (\{0,1\}^n)^t \to \{0,1\}$ can be written as the sum of at most n^t products of variables), but (2) almost all such functions require depth-three Boolean circuits of subexponential size (because parity is reducible to them). Furthermore, (2') almost all t-linear functions require depth-three *canonical* circuits of size at least $\exp(\Omega(tn)^{t/(t+1)})$, see Theorem 4.1. Hence, in the context of low-degree multilinear functions, depth-three Boolean circuits (let alone canonical ones) are weaker than standard (constant-depth) multilinear circuits, and so proving lower bounds for the former may be easier.

[11] Thus, $n = 4$ and t is the number of matrices being multiplied.

[12] Concretely, the conjectured hardness of computing a multilinear function by constant-depth Boolean circuits may stem from the number (denoted n) of variables of the same type (i.e., the variables in $x^{(j)}$), even when the arity of multiplication (denoted t) is relatively small (e.g., we even consider bilinear functions), whereas in the multilinear circuits hardness seem to be related to t (cf., indeed, the aforementioned lower bound for iterated matrix multiplication).

Decoupling arity from the number of gates. In a work done independently (but subsequent to our initial posting[13]), Hrubes and Rao studied Boolean circuits with general gates [11]. They decoupled the two parameters (i.e., the number of gates and their arity), and studied the asymmetric case of large arity and a small number of gates. We refrained from decoupling these two parameters here, since for our application their maximum is the governing parameter.

1.6 Various conventions

As stated up-front, throughout this paper, when we say that a function $f : \mathbb{N} \to \mathbb{N}$ is exponential, we mean that $f(n) = \exp(\Theta(n))$. Actually, $\exp(n)$ often means $\exp(cn)$, for some unspecified constant $c > 0$. Throughout this paper, we restrict ourselves to the field GF(2), and all arithmetic operations are over this field.[14]

Tensors. Recall that any t-linear function $F : (\{0,1\}^n)^t \to \{0,1\}$ is associated with the tensor $T \subseteq [n]^t$ that describes its existing monomials (cf., Eq. (1)). This tensor is mostly viewed as a subset of $[n]^t$, but at times such a tensor is viewed in terms of its corresponding characteristic predicate or the predicate's truth-table; that is, $T \subseteq [n]^t$ is associated with the predicate $\chi_T : [n]^t \to \{0,1\}$ or with the t-dimensional array $(\chi_T(i_1, ..., i_t))_{i_1,...,i_t \in [n]}$ such that $\chi_T(i_1, ..., i_t) = 1$ iff $(i_1, ..., i_t) \in T$. The latter views are actually more popular in the literature, and they also justify our convention of writing $\sum_{k\in[m]} T_k$ instead of the symmetric difference of $T_1, ..., T_m \subseteq [n]^t$ (i.e., $(i_1, ..., i_t) \in \sum_{k\in[m]} T_k$ iff $|\{k \in [m] : (i_1, ..., i_t) \in T_k\}|$ is odd).

1.7 Organization

The rest of this paper focuses on the study of the direct and nested composition complexity of multilinear functions (and its relation to the two canonical circuit models). This study is conducted in terms of the arithmetic model outlined in Section 1.4; that is, of multilinear circuits with general multilinear gates and a complexity measure that accounts for both the number of these gates and their arity. The basic definitional issues are discussed in Section 2, upper bounds are presented in Section 3, and lower bounds in Section 4.

In Section 5.2 we study a restricted arithmetic model obtained by allowing only standard addition and multiplication gates (and considering the same complexity measure as above, except for not counting multiplication gates that are fed only by variables). While this model is quite natural, it is quite weak. Nevertheless, this model allows to separate $F^{t,n}_{\mathtt{all}}$ and $F^{t,n}_{\mathtt{diag}}$ from the "harder" $F^{2,n}_{\mathtt{leq}}$, which means that in this model we are able to prove a non-trivial lower bound on an explicit function.

In addition, mainly due to their role in the canonical framework, we also studied the size of depth-two circuits computing various multilinear functions (see Appendix C). Even in this case, we leave several open problems. One key notion in our study of depth-two circuits is that of the number of variables that influence the linear function that is obtained from the t-linear function by fixing random values to all other $t-1$ blocks of variables.

Two shorter appendices refer to (1) the effect of lower bounds (on the size of constant-size circuits) on the class $\mathcal{NL}$, and (2) worst-case vs average-case (size) complexity in the context of constant-depth circuits for multilinear functions. See Appendices A and B, respectively.

Added after first posting

Lower bounds on the size of ND-canonical circuits will follow also from a relaxed notion rigidity, which we call structured rigidity. This notion is introduced and studied in Section 4.3. In particular, we show that structured rigidity is strictly separated from the standard notion of rigidity.

In Section 5.1, we study another restricted model of arithmetic circuits – circuits that compute functions without relying on cancellations. We show that such circuits are weaker than the general arithmetic circuits considered in the bulk of the paper. Specifically, we prove a $\Omega(n^{2/3})$ lower bound on the complexity of circuits that compute an explicit function without cancellation.

[13] See ECCC TR13-043, March 2013.

[14] However, as stated in Section 1.4, all results extend to other fields.

2 Multilinear circuits with general gates

In this section we introduce a new model of arithmetic circuits, where gates may compute arbitrary multilinear functions (rather than either addition or multiplication, as in the standard model). Accompanying this new model is a new complexity measure, which takes into account both the number of gates and their arity. This model (and its restriction to depth-two circuits) is presented in Section 2.1 (where we also present a separation between the general model and its depth-two restriction). As is clear from the introduction, the model is motivated by its relation to canonical depth-three Boolean circuits. This relation is discussed in Section 2.2.

Recall that we consider t-linear functions of the form $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$, where the tn variables are partitioned into t blocks with n variables in each block and F is linear in the variables of each block. Specifically, for t and n, we consider the variable blocks $x^{(1)}, x^{(2)}, ..., x^{(t)}$, where $x^{(j)} = (x_1^{(j)}, ..., x_n^{(j)}) \in \mathrm{GF}(2)^n$.

2.1 The two complexity measures

We are interested in multilinear functions that are computed by composition of other multilinear functions, and define a conservative (or syntactic) notion of linearity that refers to the way these functions are composed. Basically, we require that this composition does not result in a polynomial that contains terms that are not multilinear, even if these terms cancel out. Let us first spell out what this means in terms of standard multilinear circuits that use (unbounded) addition and multiplication gates, as defined in [19]. This is done by saying that a function is J-linear whenever it is multilinear (but not necessarily homogeneous) in the variables that belongs to blocks in J.

- Each variable in $x^{(j)}$ is a $\{j\}$-linear function.
- If an addition gate computes the sum $\sum_{i\in[m]} F_i$, where F_i is a J_i-linear function computed by its i^{th} child, then this gate computes a $(\bigcup_{i\in[m]} J_i)$-linear function.
- If a multiplication gate computes the product $\prod_{i\in[m]} F_i$, where F_i is a J_i-linear function computed by its i^{th} child, and the J_i's are pairwise disjoint, then this gate computes a $(\bigcup_{i\in[m]} J_i)$-linear function.

We stress that if the J_i's mentioned in the last item are not pairwise disjoint, then their product cannot be taken by a gate in a multilinear circuit. We now extend this formalism to arithmetic circuits with arbitrary gates, which compute arbitrary polynomials of the values that feed into them. Basically, we require that when replacing each gate by the corresponding depth-two arithmetic circuit that computes this polynomial as a sum of products (a.k.a monomials), we obtain a standard multilinear circuit. In other words, we require the following.

Definition 2.1 (multilinear circuits with general gates): *An arithmetic circuit with arbitrary gates is called multilinear if each of its gates satisfies the following condition. Suppose that a gate computes $H(F_1, ..., F_m)$, where H is a polynomial and F_i is a J_i-linear function computed by the i^{th} child of this gate. Then, each monomial in H computes a function that is J-linear, where J is the disjoint union of the sets J_i that define the linearity of the functions multiplied in that monomial; that is, if for some set $I \subseteq [m]$ this monomial multiplies J_i-linear functions for $i \in I$, then these J_i's should be disjoint and their union should equal I. The function computed by the gate is J'-linear, where J' is the union of all the sets that define the linearity of the functions that correspond to the different monomials in H.*

Alternatively, we may require that if a gate multiplies two of its inputs (in one of the monomials computed by this gate), then the sub-circuits computing these two inputs do not depend on variables from the same block (i.e., the two sets of variables in the directed acyclic graphs rooted at these two vertices belong to two sets of blocks with empty intersection).

Definition 2.2 (the complexity of multilinear circuits with general gates): *The* arity *of a multilinear circuit is the maximum arity of its* (general) *gates, and in the* number of gates *we count only the general gates and*

not the leaves (variables). *The* complexity *of a multilinear circuit is the maximum between its arity and the number of its gates.*

- *The* nested complexity *of a multilinear function F, denoted $\mathtt{C}(F)$, is the minimum complexity of a multilinear circuit that computes F.*
- *The* direct complexity *of a multilinear function F, denoted $\mathtt{C}_2(F)$, is the minimum complexity of a depth-two multilinear circuit that computes F.*

More generally, for any $d \geq 3$, we may denote by $\mathtt{C}_d(F)$ the minimum complexity of a multilinear circuit that computes F.[15]

Clearly, $\mathtt{C}_2(F) \geq \mathtt{C}(F)$ for every multilinear function F. For linear functions F, it holds that $\mathtt{C}_2(F) \leq 2\mathtt{C}(F)$, because in this case all gates are addition gates and so, w.l.o.g., all intermediate gates can feed directly to the top gate. This is no longer the case for bilinear functions; that is, there exists bilinear functions F such that $\mathtt{C}_2(F) \gg \mathtt{C}(F)$.

Theorem 2.3 (separating $\mathtt{C}_2$ from $\mathtt{C}$): *There exist bilinear functions $F : (\mathrm{GF}(2)^n)^2 \to \mathrm{GF}(2)$ such that $\mathtt{C}(F) = O(\sqrt{n})$ but $\mathtt{C}_2(F) = \Omega(n^{2/3})$. Furthermore, the upper bound is established by a depth-three multilinear circuit.*

The furthermore clause is no coincidence: As outlined in Remark 2.4, for every t-linear function F, the value of $\mathtt{C}(F)$ is obtained by a multilinear circuit of depth at most $t+1$.

Proof: Consider a generic bilinear function $g : \mathrm{GF}(2)^{n+s} \to \mathrm{GF}(2)$, where g is linear in the first n bits and in the last $s = \sqrt{n}$ bits. Using the fact that g is linear in the first n variables, it will be useful to write $g(x,z)$ as $\sum_{i\in[s]} g_i((x_{(i-1)s+1},...,x_{is}),z)$, where each g_i is a bilinear function on $\mathrm{GF}(2)^s \times \mathrm{GF}(2)^s$. Define $f : \mathrm{GF}(2)^{2n} \to \mathrm{GF}(2)$ such that $f(x,y) = g(x, L_1(y), ..., L_s(y))$, where $L_i(y) = \sum_{k=(i-1)s+1}^{si} y_k$.

Clearly, $\mathtt{C}(f) \leq 2s+1$ by virtue of a depth-three multilinear circuit that first computes $v \leftarrow (L_1(y), ..., L_s(y))$ (using s gates each of arity s), then computes $w_i \leftarrow (g_i((x_{(i-1)s+1}, ..., x_{is}), v)$ for $i \in [s]$ (using s gates of arity $2s$), and finally compute the sum $\sum_{i\in[s]} w_i$ (in the top gate). The rest of the proof is devoted to proving that for a random g, with high probability, the corresponding f satisfies $\mathtt{C}_2(f) = \Omega(n^{2/3})$.

We start with an overview of the proof strategy. We consider all functions $f : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n \to \mathrm{GF}(2)$ that can be derived from a generic bilinear function $g : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^s \to \mathrm{GF}(2)$ (by letting $f(x,y) = g(x, L_1(y), ..., L_s(y))$). For each such function f, we consider a hypothetical depth-two multilinear circuit of complexity at most $m = 0.9n^{2/3}$ that computes f. Given such a circuit, we derive a circuit that computes the underlying function g, whereas the circuit that we derive belongs to a set of size smaller than $2^{0.9sn}$. But since the number of possible functions g is 2^{sn}, this means that most functions f derived as above from a generic g do not have depth-two multilinear circuit of complexity at most $m = 0.9n^{2/3}$; that is, for almost all such functions f, it holds that $\mathtt{C}_2(f) > 0.9n^{2/3}$. The actual argument follows.

Consider an arbitrary depth-two multilinear circuit of complexity m that computes a generic f (derived as above from a generic g). (We shall assume that the top gate of this circuit is not fed directly by any variable, which can be enforced by replacing such variable with singleton linear functions while possibly doubling m.) By the multilinear condition, the top gate of this circuit computes a function of the form

$$B(F_1(x), ..., F_{m'}(x), G_1(y), ..., G_{m''}(y)) + \sum_{i\in[m''']} B_i(x,y), \tag{5}$$

where B is a bilinear function (over $\mathrm{GF}(2)^{m'} \times \mathrm{GF}(2)^{m''}$), the F_i's and G_i's are linear functions, the B_i's are bilinear functions, each of these function depends on at most m variables, and $m' + m'' + m''' < m$.

We now consider a random restriction of y that selects at random $i_j \in \{(j-1)s+1, ..., js\}$ for each $j \in [s]$, and sets all other bit locations to zero. Thus, for a selection as above, we get y' such that $y'_i = y_i$ if $i \in \{i_1, ..., i_s\}$ and $y'_i = 0$ otherwise. In this case, $f(x,y')$ equals $g(x, y_{i_1}, ..., y_{i_s})$. We now look at the effect of this random restriction on the expression given in Eq. (5).

[15] This general definition is not used in the current paper.

The key observation is that the expected number of "live" y' variables (i.e., $y'_i = y_i$) in each B_i is at most m/s; that is, in expectation, $B_i(x, y')$ depends on m/s variables of the y-block. It follows that each $B_i(x, y')$ can be specified by $((m + m/s)\log_2 n) + m^2/s$ bits (in expectation), because $B_i(x, y')$ is a bilinear form in the surviving y-variables and in at most m variables of x, whereas such a function can be specified by identifying the variables and the bilinear form applied to them. Hence, in expectation, the residual $\sum_i B_i(x, y')$ is specified by less than $m^3/s + 2m^2 \log_2 n$ bits, and we may pick a setting (of $i_1, ..., i_s$) that yields such a description length. This means that, no matter from which function g (and f) we start, the number of possible (functionally different) circuits that we derive from Eq. (5) is at most

$$2^{m^2} \cdot \left(\sum_{k \in [m]} \binom{n}{k} \right)^m \cdot 2^{m^3/s + 2m^2 \log_2 n} \tag{6}$$

where the first factor reflects the number of possible bilinear functions B, the second factor reflects the possible choices of the linear functions $F_1, ..., F_{m'}, G_1, ..., G_{m''}$, and the third factor reflects the number of possible bilinear functions that can be computed by $\sum_i B_i(x, y')$. However, for $m = 0.9n^{2/3}$, the quantity in Eq. (6) is smaller than $2^{1.1m^3/s} < 2^{0.9sn}$, which is much smaller than the number of possible functions g (which is 2^{sn}). Hence, for $m = 0.9n^{2/3}$, not every function f can be computed as in Eq. (5), and the theorem follows. ■

Digest. The proof of the lower bound in Theorem 2.3 is quite unusual in its combination of the method of random restrictions with a counting argument.[16] This argument may be decoupled into two parts pivoted at an artificial complexity class, denoted G, that contains all functions g that have multilinear circuits of a small description. Using the random restriction, we show that if f has complexity smaller than $0.9n^{2/3}$, then the underlying g is in G. The counting argument then shows that most g's are not in G. Combining these two facts, we conclude that most functions f (constructed based on a function g as in the proof) have complexity at least $0.9n^{2/3}$. A more appealing abstraction is obtained by letting G contains all functions g that have multilinear circuits of complexity at most $0.9n^{2/3}$ such that each gate is fed by at most $n^{1/6}$ variables from the y-block.

Remark 2.4 (on the depth of multilinear circuits achieving $\mathtt{C}$): *In light of the above, it is natural to study the depth of general multilinear circuits* (as in Definition 2.1)*, and the trade-offs between depth and other parameters* (as in Definition 2.2)*. While this is not our primary focus here, we make just one observation: If $\mathtt{C}(F) = s$ for any t-linear function F, then there is a depth $t + 1$ circuit with arity and size $O(s)$ computing F as well.*[17] *This observation is proved in Proposition 4.5.*

Remark 2.5 (waiving the multilinear restriction): *We note that arbitrary arithmetic circuits that compute t-linear functions can be simulated by multilinear circuits of the same depth, while increasing their complexity measure by a factor of at most 2^t. This can be done by replacing any gate in the original circuit with $2^t - 1$ gates in the multilinear circuit such that the gate associated with $I \subseteq [t]$ computes the monomials that are I-linear (but not I'-linear, for any $I' \subset I$). The monomials that are not $[t]$-linear are not computed, and this is OK because their influence must cancel out at the top gate. This refers to the standard arithmetic model in which the computation of a polynomial must yield the same polynomial over the extension field.*[18]

2.2 Relation to canonical circuits

As outlined in Section 1.4, the direct and nested complexity of multilinear functions are closely related to the size of D-canonical and ND-canonical circuits computing the functions. Below, we spell out constructions of canonical circuits, which are depth-three Boolean functions, having size that is exponential in the relevant parameter (i.e., D-canonical circuits of size $\exp(\mathtt{C}_2)$ and ND-canonical circuits of size $\exp(\mathtt{C})$).

[16] Indeed, in some sense, this combination is also present in Andreev's super-quadratic proof for formula size [1].

[17] That is, for any t-linear F, it holds that $\mathtt{C}_{t+1}(F) = O(\mathtt{C}(F))$.

[18] We refer to the infinite extension field obtained by extending the base field with tn formal variables. In this extension field of GF(2), the polynomials x^2 and x are different.

Construction 2.6 (D-canonical circuits of size $\exp(\mathtt{C}_2)$): *Let $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$ be a t-linear function, and consider a depth-two multilinear circuit that computes F such that the top gate applies an m-ary polynomial H to the results of the m gates that compute $F_1, ..., F_m$, where each F_i is a multilinear function of at most m variables.* (Indeed, we assume, without loss of generality, that the top gate is fed by the second-level gates only, which in turn are fed by variables.)[19] *Consider the following depth-three Boolean circuit that computes F.*

1. *Let C_H be a CNF* (resp., DNF) *that computes H.*
2. *For each $i \in [m]$, let C_i be a DNF* (resp., CNF) *that computes F_i, and let C'_i be a DNF* (resp., CNF) *that computes $1 + F_i$.*
3. *Composing C_H with the various C_i's and C'_i's, while collapsing the two adjacent levels of* OR*-gates* (resp., AND-gates)*, we obtain a depth-three Boolean circuit C.*

The derived circuit C is said to be D-canonical*, and a circuit is said to be D-canonical only if it can be derived as above.*

Clearly, C computes F and has size exponential in $m \leq \mathtt{C}_2(F) - 1$. In particular, we have

Proposition 2.7 (depth-three Boolean circuits of size $\exp(\mathtt{C}_2)$): *Every multilinear function F has depth-three Boolean circuits of size $\exp(\mathtt{C}_2(F))$.*

It turns out that the upper bound provided in Proposition 2.7 is not tight: There exists multilinear functions that have depth-three Boolean circuits of size $\exp(\mathtt{C}_2(F)^{3/4})$. This follows by combining Theorem 2.3 and Proposition 2.9, where Theorem 2.3 shows that for some bilinear functions F it holds that $\mathtt{C}(F) = O(\sqrt{n}) = O(n^{2/3})^{3/4} = O(\mathtt{C}_2(F))^{3/4}$, and Proposition 2.9 shows that every multilinear function F has depth-three Boolean circuits of size $\exp(\mathtt{C}(F))$. This leads us to the construction of ND-canonical circuits.

Construction 2.8 (ND-canonical circuits of size $\exp(\mathtt{C})$): *Let $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$ be a t-linear function, and consider a multilinear circuit that computes F such that the each of the m gates applies an m-ary polynomial H_i to the results of prior gates and some variables, where H_1 corresponds to the polynomial applied by the top gate. Consider the following depth-three Boolean circuit that computes F.*

1. *For each $i \in [m]$ and $\sigma \in \mathrm{GF}(2)$, let C_i^σ be a CNF that computes $H_i + 1 + \sigma$. That is, C_i^σ evaluates to 1 iff H_i evaluate to σ.*
2. *For each $\overline{v} \stackrel{\text{def}}{=} (v_1, v_2, ..., v_m) \in \mathrm{GF}(2)^m$, let*

$$C_{\overline{v}}(x^{(1)}, ..., x^{(t)}) = \bigwedge_{i \in [m]} C_i^{v_i}(\Pi_{i,1}(x^{(1)}, ..., x^{(t)}, \overline{v}), ..., \Pi_{i,m}(x^{(1)}, ..., x^{(t)}, \overline{v})),$$

   where the $\Pi_{i,j}$'s are merely the projection functions that describe the routing in the multilinear circuit; that is, $\Pi_{i,j}(x^{(1)}, ..., x^{(t)}, \overline{v})) = v_k$ if the j^{th} input of gate i is fed by gate k and $\Pi_{i,j}(x^{(1)}, ..., x^{(t)}, \overline{v})) = x_k^{(\ell)}$ if the j^{th} input of gate i is fed by the k^{th} variable in the ℓ^{th} variable-block (i.e., the variable $x_k^{(\ell)}$).
3. *We obtain a depth-three Boolean circuit C by letting*

$$C(x^{(1)}, ..., x^{(t)}) = \bigvee_{(v_2, ..., v_m) \in \mathrm{GF}(2)^{m-1}} C_{(1, v_2, ..., v_m)}(x^{(1)}, ..., x^{(t)})$$

The derived circuit C is said to be ND-canonical*, and a circuit is said to be ND-canonical only if it can be derived as above.*

[19] Variables that feed directly into the top gate can be replaced by 1-ary identity gates.

Note that $C(x^{(1)},...,x^{(t)}) = 1$ if and only if there exists $\overline{v} = (v_1 1, v_2, ..., v_m) \in \mathrm{GF}(2)^m$ such that $v_1 = 1$ and for every $i \in [m]$ it holds that $H_i(\Pi_{i,1}(x^{(1)},...,x^{(t)},\overline{v}),...,\Pi_{i,m}(x^{(1)},...,x^{(t)},\overline{v})) = v_i$. For this choice of $\overline{v}$, the v_i's represent the values computed in the original arithmetic circuit (on an input that evaluates to 1), and it follows that C computes F. Clearly, C has size exponential in $m \leq \mathtt{C}(F)$. In particular, we have

Proposition 2.9 (depth-three Boolean circuits of size $\exp(\mathtt{C})$): *Every multilinear function F has depth-three Boolean circuits of size* $\exp(\mathtt{C}(F))$.

A key question is whether the upper bound provided in Proposition 2.9 is tight. The answer depends on two questions: The main question is whether smaller depth-three Boolean circuits can be designed by deviation from the construction paradigm presented in Construction 2.8. The second question is whether the upper bound of $\exp(m)$ on the size of the depth-two Boolean circuits used to compute m-ary polynomials (of degree at most t) is tight. In fact, it suffices to consider t-linear polynomials, since only such gates may be used in a multilinear circuit.

The latter question is addressed in Appendix C, where it is shown that any t-linear function that depends on m variables requires depth-two Boolean circuits of size at least $\exp(\Omega(\exp(-t)\cdot m))$. (Interestingly, this lower bound is tight; that is, there exist t-linear functions that depends on m variables and have depth-two Boolean circuits of size at most $\exp(O(\exp(-t)\cdot m))$.) Conjecturing that the main question has a negative answer, this leads to the following conjecture.

Conjecture 2.10 ($\mathtt{C}$ yields lower bounds on the size of general depth-three Boolean circuits): *No t-linear function $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$ can be computed by a depth-three Boolean circuit of size smaller than* $\exp(\Omega(\exp(-t)\cdot \mathtt{C}(F)))/\mathrm{poly}(n)$.

When combined with adequate lower bounds on $\mathtt{C}$ (e.g., Theorem 4.1), Conjecture 2.10 yields size lower bounds of the form $\exp(\Omega(\exp(-t)\cdot n^{t/(t+1)}))$, which yields $\exp(n^{1-o(1)})$ for $t = \sqrt{\log n}$. In the special cases that emerge from lower bounds on $\mathtt{C}$, a tighter relation may hold – as stated in the following Conjecture 2.11, which allows using larger values of t.

Conjecture 2.11 (Conjecture 2.10, stronger form for special cases): *None of the multilinear functions $F \in \{F^{t,n}_{\mathtt{tet}}, F^{t,n}_{\mathrm{mod}\,p} : p \geq 2\}$* (see Eq. (3) and Eq. (4), resp.) *can be computed by a depth-three Boolean circuit of size smaller than* $\exp(\Omega(\mathtt{C}(F)))/\mathrm{poly}(n)$. *The same holds for almost all t-linear functions.*

When combined with adequate lower bounds on $\mathtt{C}$ (e.g., Theorem 4.1), Conjecture 2.11 yields size lower bounds of the form $\exp(\Omega((tn)^{t/(t+1)}))$, which for $t = \log n$ yields $\exp(\Omega(tn))$.

The authors are in disagreement regarding the validity of Conjecture 2.10 (let alone Conjecture 2.11), but agree that also refutations will be of interest.

3 Upper Bounds

We first present a generic upper bound on the direct complexity (i.e., $\mathtt{C}_2$-value) of any t-linear function, and then present improved upper bounds that hold ("non-trivially") for some specific t-linear functions (e.g., $F^{t,n}_{\mathtt{leq}}$).

3.1 A generic upper bound

The following upper bound is derived by a (depth-two) multilinear circuit with a top gate that computes addition (i.e., a linear function).

Theorem 3.1 (an upper bound on $\mathtt{C}_2(\cdot)$ for any multilinear function): *Every t-linear function $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$ has D-canonical circuits of size* $\exp(O(tn)^{t/(t+1)})$*; that is,* $\mathtt{C}_2(F) = O((tn)^{t/(t+1)})$.

Here (and elsewhere), we use the fact that $t^{t/(t+1)} = \Theta(t)$.

Proof: We partition $[n]^t$ into m subcubes such that the side-length of each subcube (i.e., $\ell \stackrel{\text{def}}{=} n/m^{1/t}$) equals m/t. This balances the number of subcubes against the number of variables corresponding to each subcube (i.e., $t \cdot \ell$). We then write the tensor that corresponds to F as a sum of tensors that are each restricted to one of the aforementioned subcubes. Details follow.

We may assume that $t = O(\log n)$, since the claim holds trivially for $t = \Omega(\log n)$. Partition $[n]^t$ into m cubes, each having a side of length $\ell = (n^t/m)^{1/t} = n/m^{1/t}$; that is, for $k_1, ..., k_t \in [n/\ell]$, let $C_{k_1,...,k_t} = I_{k_1} \times \cdots \times I_{k_t}$, where $I_k = \{(k-1)\ell + j : j \in [\ell]\}$. Clearly, $[n]^t$ is covered by this collection of cubes, and the sum of the lengths of each cube is $t\ell$. Let T be the tensor corresponding to F. Then,

$$\begin{aligned} F(x^{(1)}, ..., x^{(t)}) &= \sum_{k_1,...,k_t \in [n/\ell]} F_{k_1,...,k_t}(x^{(1)}, ..., x^{(t)}) \\ &\text{where } F_{k_1,...,k_t}(x^{(1)}, ..., x^{(t)}) = \sum_{(i_1,...,i_t) \in T \cap C_{k_1,...,k_t}} x^{(1)}_{i_1} \cdots x^{(t)}_{i_t}. \end{aligned}$$

It follows that $\mathtt{C}_2(F) \leq \max(t\ell, m)$, which in turn is $O((tn)^{t/(t+1)})$ if we choose $m = t\ell$ (and use $\ell = n/m^{1/t}$). ■

3.2 Improved upper bounds for specific functions (e.g., $F^{t,n}_{\mathtt{leq}}$)

Clearly, the generic upper bound can be improved upon in many special cases. Such cases include various t-linear functions that are easily reducible to linear functions such as (1) $F^{t,n}_{\mathtt{all}}(x^{(1)}, ..., x^{(t)}) = \sum_{i_1,...,i_t \in [n]} x^{(1)}_{i_1} \cdots x^{(t)}_{i_t} = \prod_{j \in [t]} \sum_{i \in [n]} x^{(j)}_i$ and (2) $F^{t,n}_{\mathtt{diag}}(x^{(1)}, ..., x^{(t)}) = \sum_{i \in [n]} x^{(1)}_i \cdots x^{(t)}_i$. Specifically, we can easily get $\mathtt{C}_2(F^{t,n}_{\mathtt{all}}) \leq t\sqrt{n} + 1$ and $\mathtt{C}_2(F^{t,n}_{\mathtt{diag}}) \leq t\sqrt{n}$. In both cases, the key observation is that each n-way sum can be written as a sum of $\sqrt{n}$ functions such that each function depends on $\sqrt{n}$ of the original arguments. Furthermore, in both cases, we could derive (depth-three) multilinear formulae of complexity $t\sqrt{n}+1$ that use only ($\sqrt{n}$-way) addition and (t-way) multiplication gates. While such multilinear formulae do not exist for $F^{2,n}_{\mathtt{leq}}$ (see Section 5.2), the full power of (depth-two) multilinear circuits with general gates yields $\mathtt{C}_2(F^{2,n}_{\mathtt{leq}}) = O(\sqrt{n})$.

Proposition 3.2 (an upper bound on $\mathtt{C}_2(F^{2,n}_{\mathtt{leq}})$): *The bilinear function* $F^{2,n}_{\mathtt{leq}}$ (of Eq. (2)) *has D-canonical circuits of size* $\exp(O(\sqrt{n}))$*; that is,* $\mathtt{C}_2(F^{2,n}_{\mathtt{leq}}) = O(\sqrt{n})$.

Proof: Letting $s \stackrel{\text{def}}{=} \sqrt{n}$, we are going to express $F^{2,n}_{\mathtt{leq}}$ as a polynomial in $3s$ functions, where each of these functions depends on $O(s)$ variables. The basic idea is to partition $[n]^2$ into s^2 squares of the form $S_{i,j} = [(i-1)s+1, is] \times [(j-1)s+1, js]$, and note that $\cup_{i<j} S_{i,j} \subset T^{2,n}_{\mathtt{leq}} \subset \cup_{i \leq j} S_{i,j}$. Thus, $F^{2,n}_{\mathtt{leq}}$ can be computed by computing separately the contribution of the diagonal squares and the contribution of the squares that are above the diagonal. The contribution of the square $S_{i,i}$ can be computed as a function of the $2s$ variables that correspond to it, while the contribution of each off-diagonal square can be computed as the product of the corresponding sum of $x^{(1)}$-variables and the corresponding sum of $x^{(2)}$-variables. Details follow.

- For every $i \in [s]$, let $L_i(x^{(1)}) = \sum_{j \in [s]} x^{(1)}_{(i-1)s+j}$, which means that $L_i(x^{(1)})$ only depends on $x^{(1)}_{(i-1)s+1}, ..., x^{(1)}_{is}$.
- For every $i \in [s]$, let $L'_i(x^{(2)}) = \sum_{j \in [s]} x^{(2)}_{(i-1)s+j}$.
- For every $i \in [s]$, let $Q_i(x^{(1)}, x^{(2)}) = \sum_{(j_1,j_2) \in T^{2,s}_{\mathtt{leq}}} x^{(1)}_{(i-1)s+j_1} \cdot x^{(2)}_{(i-1)s+j_2}$, which means that $Q_i(x^{(1)}, x^{(2)})$ only depends on $x^{(1)}_{(i-1)s+1}, ..., x^{(1)}_{is}$ and $x^{(2)}_{(i-1)s+1}, ..., x^{(2)}_{is}$.

Noting that

$$F^{2,n}_{\mathtt{leq}}(x^{(1)}, x^{(2)}) = \sum_{i \in [s]} Q_i(x^{(1)}, x^{(2)}) + \sum_{1 \leq i < j \leq s} L_i(x^{(1)}) \cdot L'_j(x^{(2)}),$$

the claim follows. ■

We turn to another bilinear function, the function $F^{2,n}_{\mathrm{mod}\,p}$, where $F^{t,n}_{\mathrm{mod}\,p}$ is defined in Eq. (4).

Proposition 3.3 (an upper bound on $\mathtt{C}_2(F^{2,n}_{\mathrm{mod}\,p})$): *The bilinear function $F^{2,n}_{\mathrm{mod}\,p}$ has D-canonical circuits of size $\exp(O(\sqrt{n}))$; that is, $\mathtt{C}_2(F^{2,n}_{\mathrm{mod}\,p}) = O(\sqrt{n})$.*

Proof: Let $s = \sqrt{n}$, and let's consider first the case $p \leq s$. For every $r \in \mathbb{Z}_p$, consider the functions $L_r(x^{(1)}) = \sum_{i \equiv r \pmod p} x^{(1)}_i$ and $L'_r(x^{(2)}) = \sum_{i \equiv r \pmod p} x^{(2)}_i$. Then, $F^{2,n}_{\mathrm{mod}\,p}(x^{(1)}, x^{(2)}) = \sum_{r \in \mathbb{Z}_p} L_r(x^{(1)}) \cdot L'_{p-r}(x^{(2)})$. Each of the foregoing $p \leq s$ linear functions depend on n/p variables, which is fine if $p = \Omega(s)$. Otherwise, we replace each linear function by $\lceil n/ps \rceil$ auxiliary functions (in order to perform each n/p-way summation), which means that in total we have $2p \cdot \lceil n/ps \rceil = O(s)$ functions (each depending on $\frac{n/p}{\lceil n/ps \rceil} \leq s$ variables).

In the case of $p > s$, we face the opposite problem; that is, we have too many linear functions, but each depends on $n/p < s$ variables. So we just group these functions together; that is, for a partition of $\mathbb{Z}_p$ to s equal parts, denoted $P_1, ..., P_s$, we introduce s functions of the form

$$Q_i(x^{(1)}, x^{(2)}) = \sum_{r \in P_i} \left(\sum_{j \equiv r \pmod p} x^{(1)}_j \right) \cdot \left(\sum_{j \equiv p-r \pmod p} x^{(2)}_j \right)$$

where $i \in [s]$ (and so avoid using the linear functions). Clearly, $F^{2,n}_{\mathrm{mod}\,p}(x^{(1)}, x^{(2)}) = \sum_{i \in [s]} Q_i(x^{(1)}, x^{(2)})$, and each Q_i depends on $2 \cdot \lceil p/s \rceil \cdot \lceil n/p \rceil = O(s)$ variables. ■

Finally, we turn to t-linear functions with $t > 2$. Specifically, we consider the t-linear function $F^{t,n}_{\mathtt{leq}}$ (of Eq. (2)), focusing on $t \geq 3$.

Proposition 3.4 (an upper bound on $\mathtt{C}_2(F^{t,n}_{\mathtt{leq}})$): *For every t, it holds that $\mathtt{C}_2(F^{t,n}_{\mathtt{leq}}) = O(\exp(t) \cdot \sqrt{n})$.*

Proof: The proof generalizes the proof of Proposition 3.2, and proceeds by induction on t. We (again) let $s \stackrel{\text{def}}{=} \sqrt{n}$ and partition $[n]^t$ into s^t cubes of the form $C_{k_1,...,k_t} = I_{k_1} \times \cdots \times I_{k_t}$, where $I_k = \{(k-1)s + j : j \in [s]\}$. Actually, we prove the following inductive claim that refers to the simultaneously expressibility of the functions $F^{t,[(k-1)s+1,n]}_{\mathtt{leq}}$ for all $k \in [s]$, where

$$F^{t,[i,n]}_{\mathtt{leq}}(x^{(1)}, ..., x^{(t)}) = \sum_{(i_1,...,i_t) \in T^{t,n}_{\mathtt{leq}} : i_1 \geq i} x^{(1)}_{i_1} \cdots x^{(t)}_{i_t}. \tag{7}$$

Indeed, $F^{t,n}_{\mathtt{leq}} = F^{t,[1,n]}_{\mathtt{leq}}$. We prove, by induction on t, that the functions $F^{t,[(k-1)s+1,n]}_{\mathtt{leq}}$, for all $k \in [s]$, can be expressed as polynomials in $t2^t \cdot s$ multilinear functions such that each of these functions depends on $t \cdot s$ variables. The base case (of $t = 1$) follows easily by using the s functions $L_i(x^{(1)}) = \sum_{j \in [s]} x^{(1)}_{(i-1)s+j}$.

In the induction step, for every $j \in [t]$, define $T_j \stackrel{\text{def}}{=} \{(k_1, ..., k_t) \in T^{t,s}_{\mathtt{leq}} : k_1 = k_j < k_{j+1}\}$, where $k_{t+1} \stackrel{\text{def}}{=} s + 1$. Note that, for every $k \in [s]$, the elements of $T^{t,[(k-1)s+1,n]}_{\mathtt{leq}}$ are partitioned according to these T_j's; that is, each $(i_1, ..., i_t) \in T^{t,[(k-1)s+1,n]}_{\mathtt{leq}}$ corresponds to some $j \in [t]$ and $k_1 \geq k$ such that $(i_1, ..., i_j) \in I_{k_1} \times \cdots \times I_{k_1}$ and $(i_{j+1}, ..., i_t) \in T^{t-j,[k_1 s+1,n]}_{\mathtt{leq}}$. Thus, for every $k \in [s]$, it holds that

$$\begin{aligned} F^{t,[(k-1)s+1,n]}_{\mathtt{leq}}(x^{(1)}, ..., x^{(t)}) &= \sum_{j \in [t]} \sum_{k_1 \geq k} P^{(j)}_{k_1}(x^{(1)}, ..., x^{(j)}) \cdot F^{t-j,[k_1 s+1,n]}_{\mathtt{leq}}(x^{(j+1)}, ..., x^{(t)}) \\ &\text{where } P^{(j)}_{k_1}(x^{(1)}, ..., x^{(j)}) \stackrel{\text{def}}{=} \sum_{(i_1,...,i_j) \in (T^{j,n}_{\mathtt{leq}} \cap (I_{k_1})^j)} x^{(1)}_{i_1} \cdots x^{(j)}_{i_j}. \end{aligned}$$

It follows that all $F_{\mathtt{leq}}^{t,[(k-1)s+1,n]}$ are expressed in terms of $t \cdot s$ new functions (each depending on at most $t \cdot s$ inputs) and the functions provided by the induction hypothesis (but with different variable names).[20] So, in total we used $ts + \sum_{j \in [t-1]}(t-j)2^{t-j} \cdot s$ functions, each depending on at most ts variables. Noting that $ts + \sum_{j \in [t-1]}(t-j)2^{t-j} \cdot s$ is upper bounded by $t2^t s$, and it follows that $\mathtt{C}(F_{\mathtt{leq}}^{t,n}) =\leq t2^t \cdot \sqrt{n}$.

In order to prove $\mathtt{C}_2(F_{\mathtt{leq}}^{t,n}) \leq t2^t \cdot \sqrt{n}$, we take a closer look at the foregoing expressions. Specifically, note that all $F_{\mathtt{leq}}^{t,[(k-1)s+1,n]}$ are expressed in terms of $t2^t s$ such that each function is either expressed in terms of other functions or expressed in terms of variables. In terms of multilinear circuits this means that each gate is fed either only by other gates or only by variables. It follows that the top gate is a function of all gates that are fed directly by variables, and so we can obtain a depth-two multilinear circuit with the same (or even slightly smaller) number of gates and the same gate arity. ■

4 Lower Bounds

We believe that the generic upper bound established by Theorem 3.1 (i.e., every t-linear function F satisfies $\mathtt{C}(F) \leq \mathtt{C}_2(F) = O((tn)^{t/(t+1)})$ is tight for many explicit functions. However, we were only able to show that almost all multilinear functions have a lower bound that meets this upper bound. This result is presented in Section 4.1, whereas in Section 4.2 we present an approach towards proving such lower bounds for explicit functions.

Before proceeding to these sections, we comment that it is easy to see that the n-way Parity function P_n has complexity at least $\sqrt{n}$. Of course, $\mathtt{C}(P_n) = \Omega(\sqrt{n})$ follows by combining Proposition 2.9 with either [8] or [10], but the foregoing proof is much simpler *(to say the least)* and yields a better constant in the Ω-notation.

4.1 On the complexity of almost all multilinear functions

Theorem 4.1 (a lower bound on $\mathtt{C}(\cdot)$ for almost all t-linear functions): *For all $t = t(n)$, almost all t-linear functions $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$ satisfy $\mathtt{C}(F) = \Omega(tn^{t/(t+1)})$. Furthermore, such a t-linear function can be found in $\exp(n^t)$ time.*

Recall that $t = \Theta(t^{t/(t+1)})$. Combined with Theorem 3.1, it follows that almost all t-linear functions satisfy $\mathtt{C}(F) = \Theta(tn^{t/(t+1)})$.

Proof: For $m > t\sqrt{n}$ to be determined at the end of this proof, we upper bound the fraction of t-linear functions F that satisfy $\mathtt{C}(F) \leq m$. Each such function F is computed by a multilinear circuit with at most m gates, each of arity at most m. Let us denote by H_i the function computed by the i^{th} gate.

Recall that each of these polynomials (i.e., H_i's) is supposed to compute a $[t]$-linear function. We shall only use the fact that each H_i is t-linear in the original variables and in the other gates of the circuit; that is, we can label each gate with an integer $i \in [t]$ (e.g., i may be an block of variables on which this gate depends) and require that functions having the same label may not be multiplied nor can they be multiplied by variables of the corresponding block.

Thus, each gate specifies (1) a choice of at most m original variables, (2) a t-partition of the m auxiliary functions, and (3) a t-linear function of the m variables and the m auxiliary function. (Indeed, choice (2) is common to all gates.) Thus, the number of such choices is upper bounded by

$$\binom{tn}{m} \cdot t^m \cdot 2^{((2m/t)+1)^t} \tag{8}$$

where $((2m/t)+1)^t$ is an upper bound on the number of monomials that may appear in a t-linear function of $2m$ variables, which are partitioned into t blocks. (Denoting by m_j the number of variables and/or

[20] By the induction hypothesis, for every $t' \in [t-1]$, we can express the functions $F_{\mathtt{leq}}^{t-t',[(k-1)s+1,n]}(x^{(1)},...,x^{(t-t')})$ for all $k \in [s]$, but here we need the functions $F_{\mathtt{leq}}^{t-t',[(k-1)s+1,n]}(x^{(t'+1)},...,x^{(t)})$. Still, these are the same functions, we just need to change the variable names in the expressions.

gates that belong to the j^{th} block, the number of possible monomials is $\prod_{j\in[t]}(m_j+1)$, where in our case $\sum_{j\in[t]} m_j \leq 2m$.) Note that Eq. (8) is upper bounded by $\exp((m/t)^t + m\log tn) = \exp((m/t)^t)$, where the equality is due to $m > t\sqrt{n} > t\log n$ and $t \geq 2$ (as we consider here).

It follows that the number of functions that can be expressed in this way is $\exp((m/t)^t)^m$, which equals $\exp(m^{t+1}/t^t)$. This is a negligible fraction of the number (i.e., 2^{n^t}) of t-linear functions over $(\mathrm{GF}(2)^n)^t$, provided that $m^{t+1}/t^t \ll n^t$, which does hold for $m = O(tn^{t/(t+1)})$. The main claim follows.

The furthermore claim follows by noting that, as is typically the case in counting arguments, both the class of admissible functions and the class of computable functions (or computing devices) are enumerable in time that is polynomial in the size of the class. Moreover, the counting argument asserts that the class of t-linear functions is the larger one (and it is also larger than 2^{tn}, which represents the number of possible inputs to each such function). ■

Open problems. The obvious problem that arises is proving a similar lower bound for some explicit multilinear function. A modest start is the following:

Problem 4.2 (the first goal regarding lower bounds regarding $\mathtt{C}$): *Prove that* $\mathtt{C}(F) = \Omega((tn)^c)$ *for some* $c > 1/2$ *and some explicit multilinear function* $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$.

Actually, an even more modest start is to prove that $\mathtt{C}_2(F) = \Omega((tn)^c)$ for some $c > 1/2$ and some explicit multilinear function $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$; that is, to consider only *depth-two* multilinear circuits.

Problem 4.3 (the ultimate goal regarding lower bounds regarding $\mathtt{C}$): *For every* $t \geq 2$, *prove that* $\mathtt{C}(F) = \Omega((tn)^{t/(t+1)})$ *for some explicit* t*-linear function* $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$. *Ditto when* t *may vary with* n, *but* $t \leq \mathrm{poly}(n)$.

Actually, a lower bound of the form $\mathtt{C}(F) = \Omega((tn)^{\epsilon t/(\epsilon t+1)})$, for some fixed constant $\epsilon > 0$, will also allow to derive exponential lower bounds when setting $t = O(\log n)$. A concrete suggestion regarding Problem 4.2 is presented in the next subsection.

4.2 The complexity of bilinear functions and matrix rigidity

In this section we show that lower bounds on the rigidity (i.e., Valiant's matrix rigidity) of matrices yield lower bounds on the $\mathcal{C}$-value of bilinear functions associated with these matrices. We then show that even lower bounds for non-explicit matrices (e.g., generic Toeplitz (or circulant) matrices) would yield lower bounds for explicit trilinear functions, specifically, for our candidate function $F^{3,n}_{\mathtt{tet}}$ (of Eq. (3)).

Let us first recall the definition of matrix rigidity (as defined by Valiant [27] and surveyed in [15]). We say that a matrix A has rigidity d for target rank r *if every matrix of rank at most* r *disagrees with* A *on more than* d *entries*. Although matrix rigidity problems are notoriously hard, it seems that they were not extensively studied in the range of parameters that we need (i.e., rigidity $\Omega(n^{3/2})$ for rank $\Omega(n^{1/2})$). Here is its basic connection to our model.

Theorem 4.4 (reducing $\mathtt{C}$ lower bounds to matrix rigidity): *If* T *is an* n*-by-*n *matrix that has rigidity* m^3 *for rank* m, *then the corresponding bilinear function* F *satisfies* $\mathtt{C}(F) > m$.

In particular, *if there exists an* n*-by-*n *Toeplitz matrix that has rigidity* m^3 *for rank* m, *then the corresponding bilinear function* F *satisfies* $\mathtt{C}(F) > m$.

Proof: As a warm-up, we first prove that $\mathtt{C}_2(F) > m$; that is, we prove a lower bound referring to depth-two multilinear circuits rather than to general multilinear circuits. Suppose towards the contradiction that $\mathtt{C}_2(F) \leq m$, and consider the multilinear circuit that guarantees this bound. Without loss of generality,[21] it

[21] As in Construction 2.6, we may replace variables that feed directly into the top gate by 1-ary identity gates. That is, if $F(x^{(1)}, x^{(2)}) = H(F_1(x^{(1)}, x^{(2)}), ..., F_{m'}(x^{(1)}, x^{(2)}), z_{m'+1}..., z_{m-1})$, where each z_i belongs either to $x^{(1)}$ or to $x^{(2)}$, then we let $F(x^{(1)}, x^{(2)}) = H(F_1(x^{(1)}, x^{(2)}), ..., F_{m-1}(x^{(1)}, x^{(2)})$, where $F_i(x^{(1)}, x^{(2)}) = z_i$ for every $i \in [m'+1, m-1]$.

holds that $F(x^{(1)},x^{(2)}) = H(F_1(x^{(1)},x^{(2)}),...,F_{m-1}(x^{(1)},x^{(2)}))$, where H is computed by the top gate and F_i is computed by its i^{th} child. W.l.o.g, the first m' functions (F_i's) are quadratic functions whereas the others are linear functions (in either $x^{(1)}$ or $x^{(2)}$). Furthermore, each F_i depends on at most m variables. Since $H(F_1(x^{(1)},x^{(2)}),...,F_{m-1}(x^{(1)},x^{(2)}))$ is a formal bilinear polynomial (in $x^{(1)}$ and $x^{(2)}$), it follows that it has the form

$$\sum_{i\in[m']} Q_i(x^{(1)},x^{(2)}) + \sum_{(j_1,j_2)\in P} L_{j_1}(x^{(1)})L_{j_2}(x^{(2)}), \tag{9}$$

where $P \subset [m'+1,m-1]\times[m'+1,m-1]$ and each Q_i and L_j depends on at most m variables. Furthermore, each of the L_j's is one of the auxiliary functions F_i's, which means that the second sum (in Eq. (9)) depends on at most $m-1$ different (linear) functions. Note that the matrix that corresponds to the first sum in Eq. (9) has less than m^3 one-entries (since the sum of the Q_i's depends on at most $m'\cdot m^2 < m^3$ variables), whereas the matrix that corresponds to the second sum in Eq. (9) has rank at most $m-1$ (since the sum of the $L_{j_1}L_{j_2}$'s depends on at most $m-1$ linear functions). But this contradicts the hypothesis that T has rigidity m^3 for rank m.

Turning to the actual proof (of $\mathtt{C}(F) > m$), which refers to multilinear circuits of arbitrary depth, we note that in the bilinear case the benefit of depth is very limited. This is so because nested composition is beneficial only when it involves free occurrence of the original variables (since terms that are product of auxiliary functions only can be moved from the expression for F_i to the expressions that use F_i). In particular, without loss of generality, linear F_i's may be expressed in terms of the original variables only, whereas quadratic F_i's are expressed in terms of the original variables and possibly linear F_i's. Thus, the expression for $F(x^{(1)},x^{(2)})$ is as in Eq. (9), except that here for every $(j_1,j_2)\in P$ either L_{j_1} or L_{j_2} is one of the auxiliary functions F_i's (whereas the other linear function may be arbitrary).[22] This suffices for completing the argument. Details follow.

Suppose towards the contradiction that $\mathtt{C}(F) \leq m$, and consider a multilinear circuit that supports this bound. It holds that $F(x^{(1)},x^{(2)}) = H(F_1(x^{(1)},x^{(2)}),...,F_{m-1}(x^{(1)},x^{(2)}),x^{(1)}_{I_1},x^{(2)}_{I_2})$, where H is the bilinear function computed by the top gate, $|I_1|+|I_2| \leq m$ and the F_i's are auxiliary functions that are computed by other gates of the circuit, where each such gate has arity at most m. Each gate computes a bilinear (or linear) function of its arguments, which we express as a sum of monomials of the following three types.

1. Monomials that contain only auxiliary functions F_j's: Such a monomial may be either a single multilinear function or a product of two linear functions.

   Without loss of generality, such monomials exist only in the computation of the top gate (and not in the computation for any other gate, because the computation of such monomials can be moved from the current gate to all gates to which it feeds without increasing the number of variables that feed directly to these gates). For example, we replace $F = H(F_1,F_2,x^{(1)}_3)$ and $F_1 = F_5F_6 + x^{(1)}_1x^{(2)}_1$ by $F'_1 = x^{(1)}_1x^{(2)}_1$ (i.e., omitting F_5F_6 from F_1) and $F = H'(F'_1,F_5,F_6,F_2,x^{(1)}_3)$, where $H'(X_1,X_2,X_3,X_4,X_5) = X_2X_3 + H(X_1,X_4,X_5)$.

2. Monomials that contain only original variables. Each quadratic (resp., linear) function computed by any gate has at most m^2 (resp., m) such monomials.

3. Mixed monomials that consist of the product of a linear function and an original variable. Such monomials cannot exist in the computation of linear functions.

Summing together all mixed monomials (*regardless of the gate to which they belong*), we obtain at most $m-1$ quadratic forms, since each quadratic form is the product of one of the auxiliary (linear) functions F_i and a linear combination (of an arbitrary number) of the original variables. Adding to this sum (denoted S_1) the sum (denoted S_2) of all monomials (computed by the top gate) that are a product of two linear F_i's, we still have at most $m-1$ quadratic forms that are each a product of one of the auxiliary (linear) functions F_i and a linear combination of the original variables.[23] Let us denote the resulting function (i.e., S_1+S_2) by F', and the corresponding matrix by T'. Note that T' has rank at most $m-1$ (since it is the sum of at most

[22] Actually, we can combine all products that involve F_i, see below.

[23] This relies on the fact that $F_i\cdot F_j$ may be viewed as a product of F_i and the linear combination of the original variables given by the expression for F_j.

$m-1$ rank-1 matrices, which correspond to the different linear F_i's). Lastly, note that $F+F'$ contains only quadratic monomials that are each either a product of two variables or an auxiliary function, which in turn consists of at most m^2 monomials that are each a product of two variables.[24] Thus, $F+F'$ consists of at most $m \cdot m^2$ such products, which implies that T' differs from T on less than m^3 entries. This implies that T does not have rigidity m^3 for rank m, and the claim follows. ■

Before proceeding, let us generalize one of the observations used in the proof of Theorem 4.4 in order to prove the following

Proposition 4.5 (on the depth of multilinear circuits achieving $\mathtt{C}$): *If $\mathtt{C}(F) = s$ for any t-linear function F, then there is a depth $t+1$ circuit with arity and size $O(s)$ that computes F.*

Proof: Generalizing an observation made in the proof of Theorem 4.4, note that monomials in the expression for F_i that contain *only* auxiliary functions can be moved to the expressions of all functions that depend on F_i (while increasing the arity of gates by at most s). Thus, without loss of generality, each auxiliary function F_i (computed by a internal gate) can be expressed in terms of input variables and auxiliary functions that are of smaller degree (than the degree of F_i). It follows that the depth of multilinear circuits computing a t-linear function needs not exceed $t+1$. ■

Implications on $F^{3,n}_{\mathtt{tet}}$. We now suggest to try to obtain an improved lower bound on $\mathtt{C}(\cdot)$ for the trilinear function $F^{3,n}_{\mathtt{tet}}$ (see Eq. (3)), which is an *explicit* multilinear function (with $t=3$), via a reduction to proving a rigidity lower bound for a *random* (or actually *any*) Toeplitz matrix (corresponding to t=2). Recall that a Toeplitz matrix is a matrix $(t_{i,j})_{i,j\in[n]}$ such that $t_{i+1,j+1} = t_{i,j}$. The reduction, which is presented next, actually reduces proving lower bounds on $\mathtt{C}(F^{3,n}_{\mathtt{tet}})$ to proving lower bounds on the $\mathtt{C}$-value of any bilinear function that corresponds to a Toeplitz matrix.

Proposition 4.6 (from $F^{3,n}_{\mathtt{tet}}$ to Toeplitz matrices): *If there exists an n-by-n Toeplitz matrix such that the corresponding bilinear function F satisfies $\mathtt{C}(F) \geq m$, then $\mathtt{C}(F^{3,n}_{\mathtt{tet}}) = \Omega(m)$.*

Proof: For simplicity, assume that $n = 2n'+1$ is odd, and consider the trilinear function $F_3 : (\mathrm{GF}(2)^{n'+1})^3 \to \mathrm{GF}(2)$ associated with the tensor $T_3 = \{(i_1,i_2,i_3) \in [[n']]^3 : \sum_j i_j \leq n/2\}$, where $[[n']] \stackrel{\mathrm{def}}{=} \{0,1,...,n'\}$. Note that multilinear circuits for $F^{3,n}_{\mathtt{tet}}$ yield circuits of similar complexity for F_3: For $y^{(j)}_{[[n']]} = (y^{(j)}_0, y^{(j)}_1, ..., y^{(j)}_{n'})$, the value of $F_3(y^{(1)}_{[[n']]}, y^{(2)}_{[[n']]}, y^{(3)}_{[[n']]})$ equals $F^{3,n}_{\mathtt{tet}}(0^{n'}y^{(1)}_{[[n']]}, 0^{n'}y^{(2)}_{[[n']]}, 0^{n'}y^{(3)}_{[[n']]})$. This means that we may modify each of the expressions used for $F^{3,n}_{\mathtt{tet}}$ by replacing the first n' variables in each variable-block with the value 0 (i.e., omit the corresponding monomials).[25]

Next, note that if $F_3(x,y,z) = \sum_{(i,j,k)\in T_3} x_i y_j z_k$ satisfies $\mathtt{C}(F_3) \leq m$, then the same upper bound holds for any bilinear function that is associated with an $(n'+1)$-by-$(n'+1)$ triangular Toeplitz matrix (i.e., $t_{j+1,k+1} = t_{j,k}$ and $t_{j,k} = 0$ if $j < k$). This holds because any linear combination of the 1-slices of T_3 (i.e., the two-dimensional tensors $T'_i = \{(j,k) : (i,j,k) \in T\}$ for every $i \in [[n']]$) yields a transpose of a triangular Toeplitz matrix, and all such matrices can be obtained by such a combination; that is, for every $I \subseteq [[n']]$,

[24] In other words, assuming that the first $m' < m$ auxiliary functions (i.e., F_i's) are bilinear functions, we observe that

$$F = F_0 = \sum_{i=0}^{m'} Q_i + \sum_{i=m'+1}^{m-1} L_i F_i \,,$$

where Q_i is the sum of the products of pairs of variables that appear in F_i and the L_i's are arbitrary linear functions (which may depend on an arbitrary number of variables in either $x^{(1)}$ or $x^{(2)}$). Thus, $F' = \sum_{i=m'+1}^{m-1} L_i F_i$ corresponds to a tensor of rank at most $m-1$, whereas $F - F' = \sum_{i=1}^{m'} Q_i$ is the sum of at most $(m'+1)\cdot m^2$ products of pairs of variables.

[25] The opposite direction is equally simple: Just note that $F^{3,n}_{\mathtt{tet}}$ can be expressed as a sum of the values in the eight directions corresponding to $\{\pm 1\}^3$.

it holds that the matrix $(t_{j,k})_{j,k\in[[n']]}$ such that $t_{j,k} = |\{i \in I : (i,j,k) \in T\}| \bmod 2$ satisfies $t_{j,k+1} = t_{j+1,k}$ and $t_{j,k} = 0$ if $j + k > n'$, and each such matrix can be obtained by a choice of such an I. (We can and will ignore the transpose operation in the sequel.)

Finally, note that multilinear circuits for any bilinear function that is associated with a triangular Toeplitz matrix yields circuits of similar complexity for general Toeplitz matrix. This holds because each Toeplitz matrix can be written as the sum of two triangular Toeplitz matrices (i.e., an upper-triangular one and a lower-triangular one). ■

Hence, establishing an $\Omega(n^c)$ lower bound on $\mathtt{C}(F^{3,n}_{\mathtt{tet}})$ reduces to establishing this bound for some Toeplitz matrix. This gives rise to the following

Problem 4.7 (on the complexity of Toeplitz matrices): *Prove that there exists an n-by-n Toeplitz matrix such that the corresponding bilinear function F satisfies $\mathtt{C}(F) \geq n^c$, for some $c > 1/2$.*

As we saw, Problem 4.7 would be resolved by

Problem 4.8 (on the rigidity of Toeplitz matrices): *For some $c > 1/2$, prove that there exists an n-by-n Toeplitz matrix T that has rigidity n^{3c} for rank n^c.*

4.3 On structured rigidity

The proof of Theorem 4.4 shows that if a bilinear function F has complexity at most m, then the corresponding matrix T can be written as a sum of a rank $m-1$ matrix T' and a matrix that has at most m^3 one-entries. However, even a superficial glance at the proof reveals that the matrix $T - T'$ is structured: It consists of the sum of m matrices such that the one-entries of each matrix are confined to some m-by-m rectangle. This leads us to the following definition.

Definition 4.9 (structured rigidity): *We say that a matrix T has* structured rigidity (m_1, m_2, m_3) for rank *r if for every matrix R of rank at most r and for every $I_1, ..., I_{m_1}, J_1, ..., J_{m_1} \subseteq [n]$ such that $|I_1| = \cdots = |I_{m_1}| = m_2$ and $|J_1| = \cdots = |J_{m_1}| = m_3$ it holds that $T - R \not\subseteq \bigcup_{k=1}^{m_1}(I_k \times J_k)$, where $M \subseteq S$ means that all non-zero entries of the matrix M reside in the set $S \subseteq [n] \times [n]$. We say that a matrix T has* structured rigidity m^3 for rank *r if T has structured rigidity (m,m,m) for rank r.*

Clearly, rigidity is a lower bound on structured rigidity (i.e., if T has rigidity m^3 for rank r, then T has structured rigidity m^3 for rank r), but (as shown below) this lower bound is not tight. Before proving the latter claim, we apply the notion of structured rigidity to our study.[26]

Theorem 4.10 (reducing C lower bounds to structured rigidity): *If T is an n-by-n matrix that has structured rigidity m^3 for rank m, then the corresponding bilinear function F satisfies $\mathtt{C}(F) \geq m$.*

In particular, *if there exists an n-by-n Toeplitz matrix that has structured rigidity m^3 for rank m, then the corresponding bilinear function F satisfies $\mathtt{C}(F) \geq m$.* Hence, Problem 4.7 would be resolved by

Problem 4.11 (on the structured rigidity of Toeplitz matrices): *For some $c > 1/2$, prove that there exists an n-by-n Toeplitz matrix T that has structured rigidity n^{3c} for rank n^c.*

In light of the following separation result, Problem 4.11 may be easier than Problem 4.8.

Theorem 4.12 (rigidity versus structured rigidity): *For any $m \in [n^{0.501}, n^{0.666}]$, consider a uniformly selected n-by-n Boolean matrix M with exactly $3mn$ ones. Then, with very high probability, M has structured rigidity m^3 for rank m.*

[26] As stated above, Theorem 4.10 follows from the proof of Theorem 4.4.

Note that M does not have rigidity $t = 3nm \ll m^3$ for rank zero, let alone for rank m. Hence, the gap between structured rigidity and standard rigidity (for rank m) is a factor of at least $m^3/t = \Omega(m^2/n)$.

Proof: For each sequence $M_0, R_1, ..., R_m$ such that M_0 has rank m and each $R_i \subseteq [n] \times [n]$ is an m-by-m rectangle, we shall show that

$$\Pr_M \left[M - M_0 \subseteq \bigcup_{i \in [m]} R_i \right] < 2^{-3nm}, \tag{10}$$

where M is a uniformly selected n-by-n matrix with exactly $t = 3mn$ ones (and $M - M_0 \subseteq S$ means that all non-zero entries of the matrix $M - M_0$ reside in the set $S \subseteq [n] \times [n]$). The theorem follows since the number of such sequences (i.e., a rank m matrix M_0 and sparse rectangles $R_1, ..., R_m$) is at most $2^{2mn} \cdot \binom{n}{m}^{2m} \ll 2^{2.5nm}$, using $m^2 \log n < nm/4$ (equiv., $m = o(n/\log n)$). We shall also use $m \leq n^{2/3}/2$, which implies $m^3 \leq n^2/8$ and $t = o(n^2)$. We consider two cases

Case 1: M_0 ***has at least*** $n^2/3$ ***one-entries.*** Since $t = o(n^2)$, it follows that $M - M_0$ has at least $n^2/4$ non-zero entries, but these cannot be covered by the $\bigcup_i R_i$, since the latter has at most $m^3 \leq n^2/8$ elements. Hence, $M - M_0 \subseteq \bigcup_{i \in [m]} R_i$ never holds in this case.

Case 2: M_0 ***has at most*** $n^2/3$ ***one-entries.*** In this case the union of the one-entries of M_0 and $\bigcup_i R_i$, denoted U, covers at most half of a generic n-by-n matrix. Now, selecting t random entries in the matrix, the probability that all entries reside in U at most $(1/2)^t$. But if some one-entry of M does not reside in U, then this entry is non-zero in $M - M_0$ but does not reside in $\bigcup_i R_i$. In this case, $M - M_0 \not\subseteq \bigcup_{i \in [m]} R_i$ holds. Hence, the expression at the l.h.s. of Eq. (10) is upper bounded by $2^{-t} = 2^{-3nm}$.

We conclude that with probability at least $1 - 2^{-mm/2}$, the matrix M has (m, m, m) rigidity for rank m. ■

Perspective. Recall that T has rigidity s for rank r if for every rank r matrix R and every matrix S of at most s one-entries it holds that $T = R + S$. The definition of structure rigidity further restricts the structure of S. Although we proved that this restriction may significantly increase the measure of density of the potential matrices S, we were not able to capitalize on it in order to prove rigidity bounds that improve over the n^2/r barrier for explicit matrices T. We note that an alternative restriction that allows for improving over this barrier was introduced by Dvir *et al.* [4], where it was called *monotone rigidity*. Specifically, T has monotone rigidity s for rank r if for every rank r matrix R and every matrix S of at most s one-entries it holds that $T = R \vee S$; that is, the effect of S is restricted to turning zero-entries of R into one-entries of T (equiv., turning one-entries of T into zero-entries of R). They presented an explicit matrix T such that for any matrix R of *real*[27] rank $n/100$, the matrix S must have at least $n^{1.1}$ ones.

5 On two restricted models

Focusing on our arithmetic circuit model, we consider two restricted versions of it: The first restricted model is of computation without cancellation, and the second is of computation that use only addition and multiplication gates while parametrizing their arity.

5.1 On computing without cancellation

A natural model in the context of arithmetic computation is that of computing without cancellations. We note that all our upper bounds (of Section 3) were obtained by computations that use no cancellations. Nevertheless, as one may expect, computations that use cancellation may be more efficient than computations

[27] Indeed, in contrast to the rest of our exposition, which refers to the arithmetics of GF(2) (and, in particular, to rank over GF(2), the result of [4] refers to the rank of the matrix over the real.

that do not use it. Furthermore, obtaining such a separation result is quite easy. A striking example is provided by the bilinear function $F^{2,n}_{\mathtt{had}}$ that corresponds to the Hadamard matrix $T^{2,n}_{\mathtt{had}}$ (i.e., $T^{2,n}_{\mathtt{had}} = \{(i,j) \in [n]^2 : \mathtt{ip}_2(i,j)\}$, where $n = 2^\ell$ and $\mathtt{ip}_2(i,j)$ is the inner product (mod 2) of the ℓ-bit binary expansions of $i-1$ and $j-1$).

Proposition 5.1 (computing $F^{2,n}_{\mathtt{had}}$ without cancellation): *Computing $F^{2,n}_{\mathtt{had}}$ without cancellations requires a circuit of complexity $\Omega(n^{2/3})$, where complexity is as in Definition 2.2. In contrast, $F^{2,n}_{\mathtt{had}}$ can be computed by a circuit of complexity $\widetilde{O}(\sqrt{n})$ with cancellation; actually, $\mathtt{C}_2(F^{2,n}_{\mathtt{had}}) = O(\sqrt{n \log n})$.*

Proof: Suppose that $F^{2,n}_{\mathtt{had}}$ can be computed by a circuit of complexity m that uses no cancellation. Following the argument in the proof of Theorem 4.4 and assuming that the first $m' < m$ auxiliary functions (i.e., F_i's) are bilinear functions, we observe that

$$F^{2,n}_{\mathtt{had}} = F_0 = \sum_{i=0}^{m'} Q_i + \sum_{i=m'+1}^{m-1} L_i F_i \,, \tag{11}$$

where Q_i is a sum of the products of pairs of variables that appear in F_i and the L_i's are arbitrary linear functions (which may depend on an arbitrary number of variables in either $x^{(1)}$ or $x^{(2)}$).[28] Hence, each Q_i corresponds to a tensor (or matrix) with at most m^2 one-entries, whereas each $L_i F_i$ corresponds to a rectangular tensor. By the non-cancellation hypothesis, these rectangles must be pairwise disjoint and their one-entries must be contained in $T^{2,n}_{\mathtt{had}}$ (since they cannot be cancelled). But by Lindsey's Lemma (cf., e.g., [5, p. 88]) rectangles of area greater than n must contain zero-entries of $T^{2,n}_{\mathtt{had}}$, which implies that each rectangle may have area at most n. It follows that the total area covered by all m tensors is at most $(m'+1) \cdot m^2 + (m-m') \cdot n$, whereas $T^{2,n}_{\mathtt{had}}$ has $n^2/2$ one-entries. The main claim follows.

The secondary claim follows by the fact that $T^{2,n}_{\mathtt{had}}$ has rank $\ell = \log_2 n$. Specifically, any binary function F that corresponds to a rank r matrix can be computed as the sum of r functions that correspond to rectangular tensors, where each of these r functions is computed as the product of two linear functions, and each linear function is computed as the sum of $\sqrt{n/r}$ functions that compute the sum of at most $\sqrt{rn}$ variables. This yields a depth-two circuit of complexity $\sqrt{rn}$, where the top gate is a quadratic expression in $\sqrt{rn}$ linear functions. ■

Computing $F^{3,n}_{\mathtt{tet}}$ without cancellation. While we were unable to prove that $\mathtt{C}(F^{3,n}_{\mathtt{tet}}) = \omega(\sqrt{n})$, it is quite easy to prove such a lower bound for circuits that compute $F^{3,n}_{\mathtt{tet}}$ without cancellation.

Proposition 5.2 (computing $F^{3,n}_{\mathtt{tet}}$ without cancellation): *Computing $F^{3,n}_{\mathtt{tet}}$ without cancellations requires a circuit of complexity $\Omega(n^{2/3})$, where complexity is as in Definition 2.2.*

Proof: Proceeding as in the proof of Proposition 5.1, we consider the top gate of a circuit (with m gates) that computes $F^{3,n}_{\mathtt{tet}}$ without cancellations. Here, we can write $F^{3,n}_{\mathtt{tet}}$ as

$$F_0 = \sum_{i=0}^{m'} C_i + \sum_{i=m'+1}^{m'+m''} L_i F_i + \sum_{i=m'+m''+1}^{m'+m''+m'''} Q_i F_i \,, \tag{12}$$

where $m' + m'' + m''' \leq m-1$, the cubic function C_i is a sum of the products of triples of variables that appear in the cubic function F_i (for $i \in [0, m']$), the L_i's (resp., Q_i's) are arbitrary linear (resp., quadratic)

[28] Recall that, w.l.o.g., gates that compute quadratic F_i's (for $i \in [m']$) may only feed into the top gate. Ditto for gates computing products of two linear F_i's (for $i \in [m'+1, m-1]$). Thus, $F_0 = Q_0 + \sum_{i \in [m']} F_i + \sum_{i=m'+1}^{m-1} L_{0,i} F_i$, where Q_0 is a sum of the products of pairs of variables that appear in F_0, the $L_{0,i}$'s are arbitrary linear functions, and for $i > m'$ the linear function F_i is computed by an internal gate. Furthermore, for every $i \in [m']$, it holds that $F_i = Q_i + \sum_{j=m'+1}^{m-1} L_{i,j} F_j$, where Q_i is a sum of the products of pairs of variables that appear in F_i, the $L_{i,j}$'s are arbitrary linear functions, and for $j > m'$ the linear function F_j is computed by an internal gate. Letting $L_j = \sum_{i=0}^{m'} L_{i,j}$, we get Eq. (11).

functions (which may depend on an arbitrary number of variables in either $x^{(1)}, x^{(2)}, x^{(3)}$), and the other F_i's are either quadratic (for $i \in [m'+1, m'+m'']$) or linear (for $i \in [m'+m''+1, m'+m''+m''']$).[29] Combining the two last summations in Eq. (12), we obtain

$$F_0 = \sum_{i=0}^{m'} C_i + \sum_{i=m'+1}^{m-1} L_i Q_i \tag{13}$$

where C_i is a sum of the products of triples of variables that appear in a cubic F_i, and the L_i's (resp., Q_i's) are arbitrary linear (resp., quadratic) functions (which may depend on an arbitrary number of variables in either $x^{(1)}, x^{(2)}, x^{(3)}$). Note that C_i corresponds to a tensor with one-entries that are confined to a m-by-m-by-m box, and each L_iQ_i corresponds to a tensor that is the outer product of a subset of $[n]$ and a subset of $[n]^2$. By the non-cancellation condition, *all these tensors are disjoint, and none may contain a zero-entry of* $T^{3,n}_{\mathtt{tet}}$.

We consider the boundary of the tensor $T^{3,n}_{\mathtt{tet}}$ (i.e., the set of one-entries that neighbor zero-entries), and consider the contributions of the aforementioned tensors to covering this boundary (without covering zero-entries of $F^{3,n}_{\mathtt{tet}}$). We will upper bound this contribution by $m^3 + mn$, and the claim will follow since the size of the boundary is $\Omega(n^2)$.

Actually, we shall consider covering the upper-boundary of $T^{3,n}_{\mathtt{tet}}$, defined as the part of the boundary that resides in $[n/2, n]^3$. In other words, the upper-boundary consists of all points $(i_1, i_2, i_3) \in [n/2, n]$ such that $i_1 + i_2 + i_3 = 2n$, and it has size $\Omega(n^2)$.

We first observe that the tensor corresponding to each C_j can cover at most m^2 points of the upper-boundary, because this tensor is confined to an m-by-m-by-m box $I'_j \times I''_j \times I'''_j$ and for each $(i_1, i_2) \in I'_j \times I''_j$ there exists at most one i_3 such that (i_1, i_2, i_3) resides in the upper-boundary. Hence, the contribution of $\sum_{j=0}^{m'} C_j$ to the cover is at most m^3.

Turning to the tensors that correspond to the L_jQ_j's, we note that (w.l.o.g.) each such tensor has the form $I'_j \times I''_j$, where $I'_j \subseteq [n]$ and $I''_j \subseteq [n]^2$. We first observe that only the largest $i_1 \in I'_j$ can participate in (a point that resides in) the upper-boundary, because if $(i_1, i_2, i_3) \in I'_j \times I''_j$ participates in the upper-boundary and $i'_1 > i_1$, then (i'_1, i_2, i_3) must be a zero-entry of $T^{3,n}_{\mathtt{tet}}$ (and contradiction is reached in case $i'_i \in I'_j$, since then $(i'_1, i_2, i_3) \in I'_j \times I''_j$). Next, fixing the largest $i_1 \in I'_j$, we observe that the upper-boundary contains at most n points of the form $(i_1, \cdot, \cdot)$. Hence, the contribution of $\sum_{j=m'+1}^{m-1} L_jQ_j$ to the cover is at most mn.

Having shown that the union of the aforementioned tensors can cover at most $m^3 + mn$ points in the upper-boundary, the claim follows since the size of the upper-boundary is $\Omega(n^2)$. ∎

5.2 Addition and multiplication gates of parameterized arity

In continuation to Definition 2.2, we consider a restricted complexity measure that refers only to multilinear circuits that use standard addition and multiplication gates. Needless to say, the multiplication gates in a multilinear circuit computing a t-linear function have arity at most t, whereas the arity of the addition gates is accounted for in our complexity measure. Furthermore, in our complexity measure we do *not* count multiplication gates that are fed by variables only. For sake of clarify, we spell out the straightforward adaptation of Definition 2.2:

Definition 5.3 (the complexity of multilinear circuits with standard gates): *A* standard multilinear circuit *is a multilinear circuit* (as in Definition 2.2) *having only addition and multiplication gates, and its complexity*

[29] Recall that, w.l.o.g., gates that compute cubic F_i's (for $i \in [m']$) may only feed into the top gate. Ditto for gates computing products of linear F_i's and quadratic F_i's (for $i \in [m'+1, m-1]$). Thus, $F_0 = C_0 + \sum_{i\in[m']} F_i + \sum_{i=m'+1}^{m'+m''} L_{0,i}F_i + \sum_{i=m'+m''+1}^{m'+m''+m'''} Q_{0,i}F_i$, where C_0 is a sum of the products of triples of variables that appear in F_0, the $L_{0,i}$'s (resp., $Q_{0,i}$'s) are arbitrary linear (resp., quadratic) functions, and for $i > m'$ the quadratic (resp., linear) function F_i is computed by an internal gate. Furthermore, for every $i \in [m']$, it holds that $F_i = C_i + \sum_{j=m'+1}^{m'+m''} L_{i,j}F_j + \sum_{j=m'+m''+1}^{m'+m''+m'''} Q_{i,j}F_j$, where C_i is a sum of the products of triples of variables that appear in F_i, the $L_{i,j}$'s (resp., $Q_{i,j}$'s) are arbitrary linear (resp., quadratic) functions, and for $j > m'$ the quadratic (resp., linear) function F_j is computed by an internal gate. Letting $L_j = \sum_{i=0}^{m'} L_{i,j}$ and $Q_j = \sum_{i=0}^{m'} Q_{i,j}$, we get Eq. (12).

is the maximum between the arity of its gates and the number of its non-trivial gates, where the trivial gates *are multiplication gates that are fed by variables only. The* restricted complexity *of a multilinear function F, denoted $\mathtt{RC}(F)$, is the minimum complexity of a standard multilinear circuit that computes F.*

Indeed, we avoided introducing a depth-two version of Definition 5.3. Note that for every t-linear function F, it holds that $\mathtt{C}(F) \leq t \cdot \mathtt{RC}(F)$, since trivial multiplication gates can be eliminated by increasing the arity of the circuit (in the general model) by a factor of at most t.[30]

5.2.1 The restricted model separates $F^{t,n}_{\mathtt{all}}$ and $F^{t,n}_{\mathtt{diag}}$ from $F^{2,n}_{\mathtt{leq}}$

As stated (implicitly) in Section 3.2, it holds that $\mathtt{RC}(F^{t,n}_{\mathtt{all}}) \leq t\sqrt{n}+1$ and $\mathtt{RC}(F^{t,n}_{\mathtt{diag}}) \leq t\sqrt{n}$. We show that this upper bound does not hold for $F^{2,n}_{\mathtt{leq}}$. We start with a general result.

Theorem 5.4 (lower bound on the restricted complexity of bilinear functions): *Let $F : (\mathrm{GF}(2)^n)^2 \to \mathrm{GF}(2)$ be a bilinear function with a corresponding tensor $T \subseteq [n]^2$. If T has rigidity ϵn^2 with respect to rank $r > 1$, then $\mathtt{RC}(F) \geq \min(r, \sqrt{\epsilon} \cdot n)$.*

Using $r = \Omega(1/\epsilon)$, we obtain $\mathtt{RC}(F) = \Omega(\min(1/\epsilon, \sqrt{\epsilon} \cdot n))$, which is optimized at $\epsilon = n^{-2/3}$ yielding $\mathtt{RC}(F) = \Omega(n^{2/3})$. Such a rigidity bound can be established for $T^{2,n}_{\mathtt{leq}}$ (cf. Proposition 5.5). For a random matrix T, we can obtain rigidity $\Omega(n^2)$ with respect to rank $\Omega(n)$, which implies that for almost all bilinear functions F it holds that $\mathtt{RC}(F) = \Omega(n)$. The latter lower bound is tight, since (for any $t \geq 1$) any t-linear function F satisfies $\mathtt{RC}(F) \leq n^{t/2}$ (via a multilinear formula with addition gates that sum-up all the relevant monomials).

Proof: We assume that $m \stackrel{\mathrm{def}}{=} \mathtt{RC}(F) < \sqrt{\epsilon} \cdot n$, and show that $m \geq r$. Consider a standard multilinear circuit that computes F with m' addition gates of arity at most m and m'' non-trivial multiplication gates, where $m' + m'' \leq m$. Note that the top gate cannot be a multiplication gate, because such a multilinear circuit can only compute bilinear functions that correspond to rank-1 matrices. Thus, the circuit, which is a directed acyclic graph (DAG) rooted at the top gate, can be decomposed into a top layer that consists of a DAG of addition gates, an intermediate layer of multiplication gates, and a bottom layer that consists of a DAG of addition gates and variables (which feeds linear functions to the multiplication gates). We note that the number of trivial multiplication gates that feed the top DAG is at most m^2, because this DAG has $m' \leq m$ addition gates each of in-degree at most m.

We truncate the foregoing circuit at the trivial multiplication gates (which compute products of variables), obtaining a new circuit that computes a bilinear function F' with a tensor T' such that $|T + T'| \leq m^2$ (since $T + T'$ corresponds to the function computed by the sum of the trivial multiplication gates). This new circuit has no trivial gates and it has m'' non-trivial multiplication gates (each computing a bilinear function that corresponds to a rank-1 matrix). Hence T' has rank at most m'' (since it is the sum of m'' rank-1 matrices). We consider two cases:

1. If $m'' \leq r$, then T' has rank at most r, and we derive a contradiction to the hypothesis that T has rigidity ϵn^2 with respect to rank r, since $|T + T'| \leq m^2 < \epsilon n^2$ (recalling our assumption that $m < \sqrt{\epsilon} \cdot n$).
2. Otherwise, $m'' \geq r$, and it follows that $m \geq r$.

The claim follows. ■

Proposition 5.5 (a bound on the rigidity of $T^{2,n}_{\mathtt{leq}}$): *For every $r < n/O(1)$, the tensor $T^{2,n}_{\mathtt{leq}}$* (of Eq. (2)) *has rigidity at least $\Omega(n^2/r)$ with respect to rank r.*

[30] In a gate that is fed by a trivial multiplication-gate, the argument representing the trivial gate's output is replaced by the (up to) t input variables feeding this trivial gate.

The rigidity lower bound is quite tight, since $T^{2,n}_{\texttt{leq}}$ is $O(1/r)$-close to $\sum_{k\in[r]}(I_k \times J_k)$, where for every $k \in [r]$ it holds that $I_k = \{(k-1)n/r+1, ..., kn/r\}$ and $J_k = \{(k-0.5)n/r+1, ..., n\}$.

Proof: For a constant $c > 1$ to be determined later, we consider any $r < n/c$. We shall prove that any matrix $T' = (T'_{i,j})_{i,j\in[n]}$ of rank r is $\Omega(1/r)$-far from $T \stackrel{\text{def}}{=} T^{2,n}_{\texttt{leq}}$; that is, $|T' + T| = \Omega(n^2/r)$.

Let T' be an arbitrary matrix of rank at most r. We say that $i \in [n]$ is good if $|\{j \in [n] : T'_{i,j} \neq T_{i,j}\}| < n/cr$. The claim of the proposition reduces to proving that at least half of $i \in [n]$ are not good, since in this case T' disagrees with T on at least $\frac{n}{2} \cdot \frac{n}{cr} = \frac{n^2}{2cr}$ entries. It is thus left to prove the latter claim.

Let G denote the set of good $i \in [n]$, and supposed towards the contradiction that $|G| > n/2$. For $c' \in [1, c/2]$ to be (implicitly) determined later, select $c'r$ indices $i_1, ..., i_{c'r} \in G$ such that for every $k \in [c'r-1]$ it holds that $i_{k+1} > i_k + (n/2c'r)$. Let us denote the i_k^{th} row of T by v_k, and the i_k^{th} row of T' by v'_k. Then, for a random non-empty set $K \subseteq [c'r]$, it holds that

1. with probability greater than $1 - 2^{-r}$, the vector $\sum_{k\in K} v_k$ has weight greater than $n/6$; and
2. with probability at least 2^{-r}, the vector $\sum_{k\in K} v'_k$ has weight 0.

(The first claim follows from the structure of T and the distance between the i_k's, whereas the second claim follows from the rank of T'.)[31] Combining (1) and (2), it follows that there exists non-empty set $K \subseteq [c'r]$ such that the vector $\sum_{k\in K} v_k$ has weight greater than $n/6$ but the vector $\sum_{k\in K} v'_k$ has weight 0. But this is impossible because the distance between these two vectors is at most $|K| \cdot n/(cr) \leq c'n/c < n/6$, where the last inequality require selecting $c > 6c'$. The claim (that $|G| \leq n/2$) follows. ■

Corollary 5.6 (lower bound on the restricted complexity of $F^{2,n}_{\texttt{leq}}$): $\texttt{RC}(F^{2,n}_{\texttt{leq}}) = \Omega(n^{2/3})$.

Indeed, Corollary 5.6 follows by combining Theorem 5.4 and Proposition 5.5, while using $r = n^{2/3}$ and $\epsilon = 1/r$. The resulting lower bound is tight:

Proposition 5.7 (upper bound on the restricted complexity of $F^{2,n}_{\texttt{leq}}$): $\texttt{RC}(F^{2,n}_{\texttt{leq}}) = O(n^{2/3})$.

Proof: Consider a partition of $[n]^2$ into $n^{4/3}$ squares, each with side $\ell = n^{1/3}$: For $i, j \in [n/\ell]$, let $S_{i,j} = [(i-1)n/\ell + 1, in/\ell] \times [(j-1)n/\ell + 1, jn/\ell]$, and note that $\cup_{i<j} S_{i,j} \subset T^{2,n}_{\texttt{leq}} \subset \cup_{i\leq j} S_{i,j}$. Thus, $F^{2,n}_{\texttt{leq}}$ can be computed by computing separately the contribution of the $n/\ell = n^{2/3}$ diagonal squares and the contribution of the squares that are above the diagonal. The contribution of the square $S_{i,i}$ can be computed as the sum of its relevant $\ell^2 = n^{2/3}$ entries, which means that the sum of the contribution of all diagonal squares consists of less than $n^{4/3}$ monomials. This sum can be computed by $n^{2/3} + 1$ addition gates, each of arity $n^{2/3}$.

The contribution of the above-diagonal squares can be computed by writing $\cup_{i<j} S_{i,j}$ as $\sum_{i\in[n/\ell]} R_i$, where $R_i = [(i-1)n/\ell + 1, in/\ell] \times [(i-1)n/\ell]$. The contribution of each of the $n/\ell = n^{2/3}$ rectangles (i.e., R_i's) can be computed by multiplying two linear expressions. The point is that there are $n^{2/3}$ linear expressions each involving $\ell = n^{1/3}$ variables of the first block, and $n^{2/3}$ linear expressions each involving a prefix of the sequence of variables of the second block. The former $n^{2/3}$ linear expressions can be computed by $n^{2/3}$ addition gates, each of arity $n^{1/3}$, whereas the latter can be computed by $n^{2/3}$ addition gates, each of arity $n^{1/3} + 1$ by using $[(i-1)n/\ell] = [(i-2)n/\ell] \cup [(i-2)n/\ell + 1, (i-1)n/\ell]$ (i.e., the i^{th} addition gate sums the result of the $i-1^{\text{st}}$ addition gate and ℓ new variables). The claim follows. ■

[31] Specifically, for a random K, the weight of the vector $\sum_{k\in K} v_k$ is distributed as $\sum_{j\in[c'r-1]}(i_{j+1} - i_j) \cdot X_j$, where $X_j = \sum_{k\in K} T_{i_k,i_j}$ (mod 2, as always). Thus, $X_j = \sum_{k\leq j} Y_k$, where $Y_k = 1$ if $k \in K$ and $Y_k = 0$ otherwise, which implies that the X_j's are IIDs uniformly distributed in $\{0,1\}$. For sufficiently large c', we indeed have $\Pr[\sum_{j\in[c'r-1]} X_j > c'r/3] > 1 - 2^{-r}$, and (1) follows since $\sum_{j\in[c'r-1]}(i_{j+1} - i_j) \cdot X_j$ is greater than $(n/2c'r) \cdot \sum_{j\in[c'r-1]} X_j$. Turning to (2), consider a maximal set of independent vectors among the $v'_1, ..., v'_{c'r}$, and denote its set of indices by I. Then, $\Pr_K[\sum_{k\in K} v'_k = 0]$ can be computed by first selecting a random $K' \subseteq ([c'r] \setminus I)$, and then (for any outcome K') selecting a random $K'' \subseteq ([c'r] \cap I)$, which implies that this probability equals $2^{-|I|} \geq 2^{-r}$.

5.2.2 On the restricted complexity of almost all t-linear functions

Recall that for every t-linear function F, it holds that $\mathtt{RC}(F) = O(n^{t/2})$, by a circuit that merely adds all relevant monomials. We prove that for almost all t-linear functions this upper bound is tight up to a logarithmic factor.

Proposition 5.8 (a lower bound on $\mathtt{RC}(\cdot)$ for almost all t-linear functions): *For all $t = t(n)$, almost all t-linear functions $F : (\mathrm{GF}(2)^n)^t \to \mathrm{GF}(2)$ satisfy $\mathtt{RC}(F) = \Omega(n^{t/2}/\log n^t)$.*

Proof: We just upper bound the number of standard multilinear circuits of complexity m. Each such circuit corresponds to a DAG with m vertices, each representing either an addition gate or a (non-trivial) multiplication gate. In addition, each of these non-trivial gates may be fed by some variables or trivial multiplication gates (which are not part of this DAG), but the number of such feeds is at most m and each is selected among at most $(n+1)^t$ possibilities. Thus, the number of such circuits is at most

$$2^m \cdot 2^{\binom{m}{2}} \cdot \binom{(n+1)^t}{m}^m \tag{14}$$

where $2^{\binom{m}{2}}$ upper bounds the number of m-vertex DAGs, 2^m accounts for choice of the gate types, and $\binom{(n+1)^t}{m}$ accounts for the choice of DAG-external feeds to each gate. Clearly, Eq. (14) is upper bounded by $((n+1)^t)^{m^2} = \exp(tm^2 \log n)$, whereas the number of t-linear functions is 2^{n^t}. The claim follows. ∎

Acknowledgments

We are grateful to Or Meir for extremely helpful discussions.

References

[1] A.E. Andreev. On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes. *Moscow Univ. Math. Bull.*, Vol. 42 (1), pages 63–66, 1987.

[2] M. Ajtai. Σ_1^1-formulae on finite structures. *Ann. Pure Appl. Logic*, Vol. 24 (1), pages 1–48, 1983.

[3] L. Babai. Random oracles separate PSPACE from the Polynomial-Time Hierarchy. *IPL*, Vol. 26, pages 51–53, 1987.

[4] Z. Dvir, S. Saraf, and A. Wigderson. Improved rank bounds for design matrices and a new proof of Kelly's theorem. ECCC, TR12-138, 2012.

[5] P. Erdos and J. Spencer. *Probabilistic Methods in Combinatorics*. Academic Press, Inc., New York, 1974.

[6] M.L. Furst, J.B. Saxe, and M. Sipser. Parity, Circuits, and the Polynomial-Time Hierarchy. *Mathematical Systems Theory*, Vol. 17 (1), pages 13–27, 1984. Preliminary version in *22nd FOCS*, 1981.

[7] O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

[8] J. Hastad. Almost Optimal Lower Bounds for Small Depth Circuits. *Advances in Computing Research: a research annual*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 143–170, 1989. Extended abstract in *18th STOC*, 1986.

[9] J. Hastad. Computational Limitations for Small Depth Circuits. MIT Press, 1987.

[10] J. Hastad, S. Jukna. and P. Pudlak. Top-Down Lower Bounds for Depth-Three Circuits. *Computational Complexity*, Vol. 5 (2), pages 99–112, 1995.

[11] P. Hrubes and A. Rao. Circuits with Medium Fan-In. ECCC, TR14-020, 2014.

[12] S. Jukna. *Boolean Function Complexity: Advances and Frontiers*. Algorithms and Combinatorics, Vol. 27, Springer, 2012.

[13] M. Karchmer and A. Wigderson. Monotone Circuits for Connectivity Require Super-Logarithmic Depth. *SIAM J. Discrete Math.*, Vol. 3 (2), pages 255–265, 1990.

[14] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.

[15] S.V. Lokam. Complexity Lower Bounds using Linear Algebra. *Foundations and Trends in Theoretical Computer Science*, Vol. 4, pages 1–155, 2009.

[16] D. van Melkebeek. A Survey of Lower Bounds for Satisfiability and Related Problems. *Foundations and Trends in Theoretical Computer Science*, Vol. 2, pages 197-303, 2007.

[17] N. Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, Vol. 11 (1), pages 63–70, 1991.

[18] N. Nisan and A. Wigderson. Hardness vs Randomness. *Journal of Computer and System Science*, Vol. 49, No. 2, pages 149–167, 1994. Preliminary version in *29th FOCS*, 1988.

[19] N. Nisan and A. Wigderson. Lower Bound on Arithmetic Circuits via Partial Derivatives. *Computational Complexity*, Vol. 6, pages 217–234, 1996.

[20] R. Raz. Tensor-Rank and Lower Bounds for Arithmetic Formulas. Proceeding of the *42nd STOC*, pages 659–666, 2010.

[21] R. Raz and A. Yehudayoff. Lower Bounds and Separations for Constant Depth Multilinear Circuits. ECCC, TR08-006, 2008.

[22] A. Razborov. Lower bounds on the size of bounded-depth networks over a complete basis with logical addition. In *Matematicheskie Zametki*, Vol. 41, No. 4, pages 598–607, 1987 (in Russian). English translation in *Mathematical Notes of the Academy of Sci. of the USSR*, Vol. 41 (4), pages 333–338, 1987.

[23] W.J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *JCSS*, Vol. 4 (2), pages 177-192, 1970.

[24] R. Shaltiel and E. Viola. Hardness Amplification Proofs Require Majority. *SIAM J. Comput.*, Vol. 39 (7),pages 3122–3154, 2010. Extended abstract in *40th STOC*, 2008.

[25] R. Smolensky. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. In *19th STOC* pages 77–82, 1987.

[26] V. Strassen. Vermeidung von Divisionen. *J. Reine Angew. Math.*, Vol. 264, pages 182–202, 1973.

[27] L.G. Valiant. Graph-theoretic arguments in low-level complexity. Mathematical Foundations of Computer Science, Springer, Lecture Notes in Computer Science, Vol. 53, pages 162–176, 1977.

[28] L.G. Valiant. Exponential lower bounds for restricted monotone circuits. In *15th STOC*, pages 110–117, 1983.

[29] U.V. Vazirani. Efficiency Considerations in Using Semi-Random Sources. In *19th STOC*, pages 160-168, 1987.

[30] A.C. Yao. Separating the Polynomial-Time Hierarchy by Oracles. In *26th FOCS*, pages 1-10, 1985.

Appendices

Throughout the rest of this paper (i.e., in the appendices), by circuits we mean Boolean circuits. Since we are mainly interested in constant-depth circuits and in their size as being exponential in some parameters (while disregarding the constant factor in the exponent), the difference between formulas and circuits is immaterial here. Ditto with respect to the difference between the maximal fan-in of gates in the circuit (or formula) and the total size. Indeed, a depth-d formula (or circuit) of fan-in bound B, has size at most B^d, which is $\mathrm{poly}(B) = \exp(\log B)$, whenever d is constant. In other words, for a constant depth circuit (or formula) C it holds that $\log(\mathrm{size}(C))$ is linear in the logarithm of the fan-in bound.

Appendix A: On separating $\mathcal{NL}$ from $\mathcal{P}$

This appendix provides details for a comment made in the introduction regarding the effect of exponential lower bounds on the size of depth-three Boolean circuits on separating $\mathcal{NL}$ from $\mathcal{P}$.

We start by recalling a folklore result regarding the circuit complexity of $\mathcal{NL}$, which can be proved by a natural generalization of the well-known idea underlying Savitch's Theorem [23].

Theorem A.1 *Every set in $\mathcal{NL}$ has constant-depth Boolean circuits of sub-exponential size. That is, for every set $S \in \mathcal{NL}$, there exists a constant c such that for any constant $d > c$, the set S has depth-d circuits of $\exp(n^{c/d})$ size.*

(In his survey of lower bounds for Satisfiability [16], van Melkebeek describes this result and its proof in terms of alternating time; cf. [16, Sec. 3.2].)

Proof: We shall show that directed st-connectivity can be solved by depth-d circuits of size $exp(\widetilde{O}(n^{2/d}))$, where n denotes the number of vertices (and factors that depend on d are hidden in the $\widetilde{O}$-notation). Let $\Phi(G, u, v, \ell)$ denote the predicate indicating that there is a path of length at most ℓ from u to v in the graph G. Observe that $\Phi(G, v_0, v_{m+1}, \ell)$ can be written as

$$\exists v_1, ..., v_m \forall i \in [m+1]\ \Phi(G, v_{i-1}, v_i, \lceil \ell/m \rceil).$$

Indeed, in the proof of Savitch's Theorem [23], one sets $m = 1$ (and recurses for $\log_2 n$ steps), but here we set $m = n^{2/d}$ and recurse for $d/2$ steps. We obtain the desired circuit by replacing the existential quantifiers with $2^{m \log_2 n}$-way OR-gates and the universal quantifiers with $((m+1)$-way) AND-gates. ∎

Corollary A.2 *If there is a function in the class $\mathcal{C}$ that has no constant-depth circuits of subexponential size, then $\mathcal{NL}$ is not contained in $\mathcal{C}$.*

Indeed, the same argument can be applied whenever the lower bound on the size of depth-d circuits (for the function in $\mathcal{C}$) is higher than $\exp(n^{O(1/d)})$. A simple case is when the lower bound is oblivious of the constant depth (or rather holds uniformly over all constant depths). In general, the lower bound may have the form $L_d(n)$, and in such a case it suffices that for every c there exists a d such that for sufficiently large n it holds that $L_d(n) > \exp(n^{c/d})$.

Appendix B: On worst-case vs average-case

The application of circuit lower bounds to derandomization (via the hardness-to-randomness connection of cf. [17, 18]) requires strong average-case bounds, not merely worst-case ones. Here average-case refers to the uniform distribution. Before continuing the discussion, let us clarify the above notions.

We say that a circuit C approximates the Boolean function F with error probability ϵ if $\Pr_x[C(x) \neq F(x)] \leq \epsilon$, where the probability is taken over the uniform distribution (over strings of adequate length), and its advantage (over a coin toss) is defined as $(1-\epsilon) - 0.5 = 0.5 - \epsilon$. The notion of worst case corresponds to

error probability 0, a mild level of average case may refer to some *constant* error probability $\epsilon \in (0, 0.5)$, whereas a strong level of average case may refer to error probability that has the form $\epsilon(n) = 0.5 - \mu(n)$, where μ is some negligible function (e.g., $\mu(n) = 2^{-\Theta(n)}$).

The point is that even if one obtains exponential lower bounds (on the size of constant-depth circuits) for computing some explicit function, these worst-case bounds do not necessarily yield average-case lower bounds. In other settings, hardness amplification can be used to bridge the gap, but in the context of constant-depth circuits generic hardness amplification to exponentially vanishing advantage is quite unlikely (cf. [24], which assert that such a black-box amplification implies a circuit for majority). Nevertheless, for t-linear function, hardness amplification to very moderate error rate is possible.[32]

Proposition B.1 (implicit in [2, 3]): *Let $F : (\{0,1\}^n)^t \to \{0,1\}$ be a t-linear function. If F cannot be computed by depth d circuits of size s, then F cannot be approximated with error probability at most $2^{-(t+2)}$ by depth $d-4$ circuits of size $s/(\exp(2^t) \cdot \mathrm{poly}(n))$.*

Proof: Suppose that C approximates F with error probability at most ϵ; that is,

$$\Pr_{x^{(1)},...,x^{(t)}}[C(x^{(1)},...,x^{(t)}) \neq F(x^{(1)},...,x^{(t)})] \leq \epsilon.$$

Then, following Babai [3], we can obtain a randomized circuit C' such that for every $(x^{(1)},...,x^{(t)}) \in (\{0,1\}^n)^t$ it holds that $\Pr[C'(x^{(1)},...,x^{(t)}) = F(x^{(1)},...,x^{(t)})] \geq 1 - 2^t\epsilon$. Specifically, C' selects uniformly $r^{(1)},...,r^{(t)} \in \{0,1\}^n$, and computes

$$C'(x^{(1)},...,x^{(t)};r^{(1)},...,r^{(t)}) \stackrel{\text{def}}{=} \sum_{(\sigma_1,...,\sigma_t)\in\{0,1\}^t} C(\sigma_1 x^{(1)} + r^{(1)},...,\sigma_t x^{(t)} + r^{(t)}).$$

Note that C' can be implemented in depth $\mathrm{depth}(C)+2$ and size $\exp(2^t)\cdot\mathrm{size}(C)$. Assuming that $\epsilon \leq 2^{-(t+2)}$, the error probability of C' is at most $1/4$, since $\sum_{(\sigma_1,...,\sigma_t)\in\{0,1\}^t} F(\sigma_1 x^{(1)} + r^{(1)},...,\sigma_t x^{(t)} + r^{(t)})$ equals $F(x^{(1)},...,x^{(t)})$.

We now apply Ajtai's amplification procedure [2]. First, we reduce the error probability to below n^{-3} by invoking C' for $\ell = O(\log n)$ times (with independent coins) and taking a majority vote; that is, $C''(x;\omega_1,...,\omega_\ell) = \mathtt{MAJ}(C'(x;\omega_i)_{i\in[\ell]})$. This yields a (randomized) circuit C'' of depth $\mathrm{depth}(C')+1$ and size $\mathrm{poly}(n)\cdot\mathrm{size}(C')$. Next, we constructs a (randomized) circuit C''' that on input x invokes $C''(x)$ for n^2 times, using coins $\omega_{1,1},...,\omega_{n,n}$, and outputs $\bigvee_{i\in[n]}\bigwedge_{j\in[n]} C''(x;\omega_{i,j})$. Note that C''' errs on x only if at least n invocations returned the wrong answer, which happens with probability at most $\binom{n^2}{n}\cdot(n^{-3})^n < 2^{-tn}$ (using $t < \log_2 n$ or else the claim holds vacuously). Fixing a sequence of coins that is good for all 2^{tn} possible inputs, we obtain a (deterministic) circuit of depth $\mathrm{depth}(C)+4$ and size $\exp(2^t)\mathrm{poly}(n)\cdot\mathrm{size}(C)$. The claim follows. ■

Comment. Indeed, the foregoing argument produces a non-canonical circuit. The first step (i.e., self-correction) would have been canonical if the $r^{(j)}$'s were considered input variables, but taking majority and computing a "weird" function (which corresponds to a "vast majority" promise problem) are not canonical. Surely, there are things that canonical circuits cannot do well, but the question is whether this matters when computing multilinear functions (rather than when doing mild hardness amplification).

Appendix C: On the size of DNFs and CNFs computing multilinear functions

We shall care both of the size of DNFs and CNFs computing various multilinear functions. The main motivation is to establish a lower bound that will be used in the sanity check for depth-three (and larger

[32] Indeed, this result falls short of obtaining a strong level of average-case hardness. Thus, it is our hope that exponential lower bounds for exact computation of multilinear functions will extend to approximation with error probability of the form $0.5 - \mu$ where μ is an exponentially vanishing function. Note that this was the case with respect to the parity lower bounds (cf., e.g., [9, Chap. 8]).

constant depth) circuits: The canonical rules for designing circuits, which are the core of these sanity checks, include the use of depth-two circuits for computing multilinear functions. (We shall actually need both DNFs and CNFs for computing each required multilinear function.) Additional motivation comes from the feeling that the depth-two case may teach us something about larger depth, but we actually doubt that feeling. Let us also warn that there may be a significant difference between the size of DNFs and CNFs, as indicated by the t-linear function $F^{t,n}_{\mathtt{all}}(x^{(1)},...,x^{(t)}) \stackrel{\rm def}{=} \sum_{i_1,...,i_t\in[n]} x^{(1)}_{i_1}\cdots x^{(t)}_{i_t}$.

Proposition C.1 (a gap between CNFs and DNFs): *The function $F^{t,n}_{\mathtt{all}}$ has CNFs of size $\widetilde{O}(2^n)$, but no DNFs of size smaller than 2^{tn-t}. In general, for any $d \geq 2$, the function $F^{t,n}_{\mathtt{all}}$ has depth-d circuits of size $\exp(n^{1/(d-1)})$.*

Proof: A depth-d circuit of size $\exp(n^{1/(d-1)})$ for computing $F^{t,n}_{\mathtt{all}}$ follows from the fact that $\sum_{i_1,...,i_t\in[n]} x^{(1)}_{i_1}\cdots x^{(t)}_{i_t}$ equals $\prod_{j\in[t]}\sum_{i\in[n]} x^{(j)}_i$. (Thus, $F^{t,n}_{\mathtt{all}}$ can be written as a conjunction of t (depth-d) parity circuits.)

The lower bound on the size of DNFs follows by observing that (1) each (non-trivial) term in such DNF must contain an occurrence of each variable, and (2) the probability that $F^{t,n}_{\mathtt{all}}$ evaluates to 1 is 2^{-t}. Specifically, regarding (1), assume towards the contradiction that some term ϕ lacks an occurrence of variable $x^{(j)}_i$, and consider an arbitrary assignment that satisfies this term. Then, flipping the value of $x^{(j)}_i$ keeps this term satisfied, whereas $F_{\mathtt{all}}$ cannot evaluate to 1 under both assignments. (Here, as well as for verifying (2), it is useful to write $F^{t,n}_{\mathtt{all}}(x^{(1)},...,x^{(t)})$ as $\prod_{j\in[t]}\sum_{i_j\in[n]} x^{(j)}_{i_j}$.) Combining (1) with (2), we infer that the number of terms, denoted M, must satisfy $M\cdot 2^{-tn} \geq 2^{-t}$. ∎

The rest of this appendix. This appendix consists of a very basic study of the size of depth-two circuits computing various multilinear functions. To set the stage, recall that a generic t-linear function F has the form $F(x^{(1)},...,x^{(t)}) = \sum_{(i_1,...,i_t)\in T} x^{(1)}_{i_1}\cdots x^{(t)}_{i_t}$, where $T \subseteq (I_1\times\cdots\times I_t) \subseteq [n]^t$. Needless to say, we shall consider the smallest possible rectangle $I_1\times\cdots\times I_t$ that contains T, which means that $I_j = \{i\in[n] : \exists(i_1,...,i_{j-1},i,i_{j+1},...,i_t)\in T\}$.

It turns out that the size of depth-two circuits for such a function F may range between exponential in $2^{-t}\cdot\sum_{j\in[t]}|I_j|$ and exponential in $\sum_{j\in[t]}|I_j|$. We shall consider both cases, as well as the intermediate case in which the size is exponential in $\max_{j\in[t]}\{|I_j|\}$.

Clearly, $\exp(\sum_{j\in[t]}|I_j|)$ is an obvious upper bound on the size of DNFs and CNFs computing F. We shall see that in some cases there exists a matching lower bound (of the form $\exp(\Omega(\sum_{j\in[t]}|I_j|))$, which means that we discard polynomial relations in size). But we first turn to lower bounds that hold in all cases, which have the weaker form of $\exp(2^{-t}\cdot\sum_{j\in[t]}|I_j|)$.

C.1 A lower bound that hold for all t-linear functions

As will be proved next, a lower bound that holds for all t-linear functions $F : (\{0,1\}^n)^t \to \{0,1\}$ has the form $\exp(2^{-t}\cdot\sum_{j\in[t]}|I_j|)$, where the $I_1\times\cdots\times I_t$ is the smallest rectangle that contains the corresponding tensor T. We shall also see that this lower bound is the best possible (with respect to lower bounds that are stated in terms of t and $\sum_{j\in[t]}|I_j|$).

Proposition C.2 (on the size of DNFs computing any multilinear function):

(general lower bound) *For every $T\subseteq[n]^t$, if $(I_1\times\cdots\times I_t)$ is the minimal rectangle that contains T, then $F(x^{(1)},...,x^{(t)}) = \sum_{(i_1,...,i_t)\in T} x^{(1)}_{i_1}\cdots x^{(t)}_{i_t}$ has neither a DNF nor a CNF of size smaller than $\exp(\exp(-t)\cdot\sum_{j\in[t]}|I_j|)$.*

(matching upper bound) *For every $n\geq m\geq 3^{t-1}$, there exists a non-empty $T\subseteq(I_1\times\cdots\times I_t)\subseteq[n]^t$ such that $\sum_{j\in[t]}|I_j|\in[m,m+O(\log m)]$ and the corresponding F has DNFs and CNFs of size $\exp(\exp(-t)\cdot\sum_{j\in[t]}|I_j|)$.*

Proof: In proving the lower bound, we assume, w.l.o.g, that $|I_1| = \max_j\{|I_j|\}$. Note that $F(x^{(1)},...,x^{(t)})$ can be written as $\sum_{i\in I_1} F_i(x^{(2)},...,x^{(t)}) \cdot x_i^{(1)}$, where each $F_i(x^{(2)},...,x^{(t)})$ is a (non-trivial) $(t-1)$-linear function. Hence, by the Schwartz–Zippel Lemma (for small fields), it holds that, for every $i \in I_1$, the probability that $F'(x^{(1)}) = F(x^{(1)}, r^{(2)},...,r^{(t)})$ depends on $x_i^{(1)}$, for uniformly chosen $r^{(2)},...,r^{(t)} \in \{0,1\}^n$, is at least $2^{-(t-1)}$. It follows that there exists $r^{(2)},...,r^{(t)} \in \{0,1\}^n$ such that $F'(x^{(1)})$ is a linear function of at least $v \stackrel{\text{def}}{=} |I_1|/2^{t-1}$ variables, and thus has no DNF or CNF of size smaller than 2^{v-1}. The lower bound follows, since $v > |I_1|/2^t = \Omega(\exp(-t)\cdot\sum_{j\in[t]}|I_j|)$.

For proving the upper bound, we first consider the case of $m = 3^{t-1}$. Associate $[m] \subseteq [n]$ with the set, denoted $3^{[t-1]}$, of all 3-way (ordered) *partitions* of $[t-1]$, and consider the function

$$F(x^{(1)},...,x^{(t)}) = \sum_{(A,B,C)\in 3^{[t-1]}} \left(\prod_{j\in A} x_1^{(j)}\right)\cdot\left(\prod_{j\in B} x_2^{(j)}\right)\cdot\left(\prod_{j\in C}(x_1^{(j)}+x_2^{(j)})\right)\cdot x_{(A,B,C)}^{(t)}$$

Indeed, this function is t-linear (since each $j\in[t-1]$ appears in exactly one part of any 3-partition $(A,B,C)\in 3^{[t-1]}$) and it depends on the variables $x_1^{(1)}, x_2^{(1)},..., x_1^{(t-1)}, x_2^{(t-1)}$ and $x^{(t)}$ (i.e., $x_{(A,B,C)}^{(t)}$ for all $(A,B,C)\in 3^{[t-1]}$). Thus, the corresponding tensor is minimally bounded by the rectangle $\{1,2\}^{t-1}\times 3^{[t-1]}$.

We show that, for any possible assignment to $x^{(1)},...,x^{(t-1)}$, at most 2^{t-1} of the 3^{t-1} variables of $x^{(t)}$ are influential. First note that for each $j\in[t-1]$ it cannot hold that $r_1^{(j)} = r_2^{(j)} = r_1^{(j)} + r_2^{(j)} = 1$. Thus, for every $r^{(1)},...,r^{(t-1)} \in \{0,1\}^n$, it holds that $|\{(A,B,C)\in 3^{[t-1]} : M_{(A,B,C)}(r^{(1)},...,r^{(t-1)}) = 1\}| \le 2^{t-1}$, where $M_{(A,B,C)}(r^{(1)},...,r^{(t-1)}) = (\prod_{j\in A} r_1^{(j)})\cdot(\prod_{j\in B} r_2^{(j)})\cdot(\prod_{j\in C}(r_1^{(j)}+r_2^{(j)}))$. This established the foregoing claim.

Now, we can write a disjunction over all $2^{2(t-1)}$ assignments to $x_1^{(1)}, x_2^{(1)},..., x_1^{(t-1)}, x_2^{(1)}$ and for each such assignment write a DNF on 2^{t-1} influential variables. That is, letting $r = (r_1^{(1)}, r_2^{(1)},..., r_1^{(t-1)}, r_2^{(t-1)})$, we write

$$F(x^{(1)},...,x^{(t)}) = \bigvee_{r\in\{0,1\}^{2(t-1)}} \phi_r(x_1^{(1)}, x_2^{(1)},..., x_1^{(t-1)}, x_2^{(t-1)}, x_{I(r)}^{(t)})$$

where $x_{I(r)}^{(t)}$ denote the sequence of variables in $x^{(t)}$ that are influential under the assignment r. (Indeed, the DNF ϕ_r computes the Boolean function $(\bigwedge_{j\in[t],i\in\{1,2\}} x_i^{(j)} = r_i^{(j)}) \wedge \mathtt{PAR}(x_{I(r)}^{(t)})$, which we do not bother to write in DNF.) Thus, we obtained a DNF of size $\exp(t+2^t) = \exp((2/3)^{t-1}\cdot m)$, since $m = 3^{t-1}$. (The same can be done with a top conjunction and CNFs, yielding a CNF.)[33] In general, when $m > 3^{t-1}$, the claim follows by partitioning $[m]$ into $\lfloor m/3^{t-1}\rfloor$ blocks of length 3^{t-1} and treating each block as above. ■

Corollary C.3 (lower bounds on the size of DNFs computing any multilinear function): *Every t-linear function that depends on all its variables has no depth-two circuits of size* $\exp(o(\exp(-t)\cdot n))$. *Furthermore, the claim hold even if the function depends only on $\Omega(n)$ of its tn variables.*

C.2 The intermediate range: a parity-level lower bound

For many natural t-linear functions, it is easy to obtain an exponential in n lower bound by reducing $\Omega(n)$-way parity to the t-linear function $F : (\{0,1\}^n)^t \to \{0,1\}$ at hand. Such a reduction amounts to showing that fixing $nt - n'$ of the input bits of F results in the parity of the remaining $n' = \Omega(n)$ bits. Using such reductions, one can easily show the following.

Proposition C.4 (reductions from parity): *Almost all t-linear function $F : (\{0,1\}^n)^t \to \{0,1\}$ cannot be computed by depth-two circuits of size $2^{0.49n}$. The t-linear functions $F_{\mathtt{leq}}^{t,n}$, $F_{\mathtt{tet}}^{t,n}$, and $F_{\mathrm{mod}\,p}^{t,n}$ for $p \le n$, cannot be computed by depth-two circuits of size 2^{n-2}.*

[33] Indeed, a corresponding CNF ψ_r computes the function $(\bigvee_{j\in[t],i\in\{1,2\}} x_i^{(j)} \neq r_i^{(j)}) \wedge \mathtt{PAR}(x_{I(r)}^{(t)})$.

The first part (i.e., regarding almost all multilinear function) is stated merely for sake of demonstrating the technique. We shall see a stronger results for almost all functions in Section C.3.

Proof: When considering a random t-linear function $F : (\{0,1\}^n)^t \to \{0,1\}$, we consider the corresponding tensor T, which is uniformly distributed among all subsets of $[n]^t$. Hence, $F(x^{(1)}, 1^{(t-1)\cdot n}) = \sum_{i\in[n]} \sigma_i \cdot x_i^{(1)}$, where $\sigma_i = |\{(i, i_2, ..., i_t) \in T : i_2, ..., i_t \in [n]\}| \bmod 2$. Thus, with overwhelmingly high probability over the choice of T, at least $0.49n$ of the σ_i's will be set to 1, which means that we can use any depth-two circuit computing T in order to compute the parity of $0.49n$ bits.

Turning to $F^{t,n}_{\mathtt{leq}}$, note that $F^{t,n}_{\mathtt{leq}}(x^{(1)}, ..., x^{(t)})$ equals $\sum_{i\in[n]} F_i(x^{(2)}, ..., x^{(t)}) \cdot x_i^{(1)}$, where $F_i(x^{(2)}, ..., x^{(t)}) = \sum_{i\le i_2\le\cdots\le i_t\le n} x^{(2)}_{i_2} \cdots x^{(t)}_{i_t}$. Hence, for every $j \in [2,t]$, setting $r^{(j)}_n = 1$ and $r^{(j)}_k = 0$ for every $k \in [n-1]$, we get $F_i(r^{(2)}, ..., r^{(t)}) = 1$ for every $i \in [n]$ (since $r^{(2)}_{i_2} \cdots r^{(t)}_{i_t} = 1$ if and only if $i_2 = \cdots = i_t = n$). Thus, $F^{t,n}_{\mathtt{leq}}(x^{(1)}, r^{(2)}, ..., r^{(t)})$ equals $\sum_{i\in[n]} x_i^{(1)}$. A similar argument applies to $F^{t,n}_{\mathtt{tet}}$ except that here (for every $j \in [2,t]$) we set $r^{(j)}_{n/2} = 1$ and $r^{(j)}_k = 0$ for every other $k \in [n]$.

Lastly, considering $F^{t,n}_{\bmod p}$, for every $j \in [3,t]$, we set $r^{(j)}_1 = 1$ and $r^{(j)}_k = 0$ for every $k \in [2,n]$, whereas $r^{(2)}_k = 1$ iff $k \in [p]$. Note that $F^{t,n}_{\bmod p}(x^{(1)}, ..., x^{(t)})$ equals $\sum_{i\in[n]} F_i(x^{(2)}, ..., x^{(t)}) \cdot x_i^{(1)}$, where here $F_i(x^{(2)}, ..., x^{(t)}) = \sum_{(i_2,...,i_t):(i,i_2,...,i_t)\in T^{t,n}_{\bmod p}} x^{(2)}_{i_2} \cdots x^{(t)}_{i_t}$, and the only term that contributes to $F_i(r^{(2)}, ..., r^{(t)})$ is the one that satisfies $i_3 = \cdots = i_t = 1$ and $i_2 \in [p]$ such that $i_2 \equiv -(i + (t-2)) \pmod p$. Indeed, for each $i \in [n]$ there exists exactly one such term. Thus, $F^{t,n}_{\bmod p}(x^{(1)}, r^{(2)}, ..., r^{(t)})$ equals $\sum_{i\in[n]} x_i^{(1)}$. ■

C.3 Lower bounds that are exponential in tn

Some indication towards the non-triviality of such lower bounds comes from looking at bilinear functions (i.e., $t = 2$). In contrast to what one may think, the size of depth-two circuits that compute $F^{2,n}_{\mathtt{leq}}$ is significantly below 2^{2n} (i.e., it is at most $2^{1.6n}$). The same holds for almost all bilinear functions. Also, for $p \le n$, the size of depth-two circuits that compute $F^{2,n}_{\bmod p}$ is at most $2^{2n-\Omega(p))}$. (In the context of $t = 2$, it makes no sense to consider $F^{t,n}_{\bmod p}$ for $p > 2n$.)

Proposition C.5 (upper bounds for some bilinear functions): *The bilinear function $F^{2,n}_{\mathtt{leq}}(x,y) = \sum_{i\le j\le n} x_i y_j$ has depth-two circuits of size $2^{1.6n}$. The same upper bound holds for almost all bilinear functions. The bilinear function $F^{2,n}_{\bmod p}$ has depth-two circuits of size $2^{\max(1.51n, 2n-\Omega(p))}$.*

Proof: In all cases, the key observation is that, for all but a small fraction of the settings of the y-variables, the number of relevant x-variables is significantly smaller than n. Thus, the DNF can consist of the disjunction of DNFs that correspond to each of the possible assignments to y, and most of these DNFs will be significantly smaller than 2^n. (For implementation details, see the proof of the upper bound in Proposition C.2.)

Starting with $F^{2,n}_{\mathtt{leq}}(x,y)$, we write $F^{2,n}_{\mathtt{leq}}(x,y) = \sum_{i\in[n]} c_i x_i$, where $c_i = \sum_{j=i}^n y_j$. Note that the c_i's are obtained by a full-rank linear transformation of the y_j's. Thus, the number of relevant x-variables for a random assignment to the y-variables (represented by k below) behaves like the Binomial distribution on n events (with success probability 1/2), and so the size of the final DNF will be

$$\begin{aligned} \mathrm{poly}(n) \cdot \sum_{k=0}^{n} \binom{n}{k} \cdot 2^k &= \mathrm{poly}(n) \cdot \max_{k\in[n]} \left\{ \binom{n}{k} \cdot 2^k \right\} \\ &= \mathrm{poly}(n) \cdot 2^{\max_{\alpha\in[0,5,1]}\{\alpha + H_2(\alpha)\}\cdot n} \end{aligned}$$

where H_2 denotes the binary entropy function. Noting that $\max_{\alpha\in[0,5,1]}\{\alpha + H_2(\alpha)\} < 1.6$, we are done.

In the case of an arbitrary bilinear function F that is associated with the tensor $T \subseteq [n]^2$, we have $F(x,y) = \sum_{i\in[n]} c_i x_i$, where $c_i = \sum_{j\in[n]:(i,j)\in T} y_j$. For a random tensor $T \subseteq [n]^2$, with very high probability, the c_i's are obtained by a rank $(n - o(n))$-rank linear transformation of the y_j's. In such a case, the number of relevant x-variables for a random assignment to the y-variables behaves like $B_{n-o(n)} + o(n)$, where $B_{n'}$

is the Binomial distribution on n' events (with success probability 1/2). Thus, the size of the DNF will be smaller than $2^{\max_{\alpha\in[0,5,1]}\{\alpha+H_2(\alpha)\}\cdot n+o(n)}$, which is smaller than $2^{1.599n+o(n)}$.

Turning to $F^{2,n}_{\mathrm{mod}\,p}$, we write $F^{2,n}_{\mathrm{mod}\,p}(x,y)=\sum_{r\in[p]}L_r(x)\cdot L_{p-r}(y)$, where $L_k(z)\stackrel{\mathrm{def}}{=}\sum_{j\in[n]:j\equiv k \pmod p} z_j$. Thus, the number of relevant x-variables for a random assignment to the y-variables behaves like $(n/p)\cdot B_p$, where B_p is the Binomial distribution on p events (which here reflect the values of the linear functions $L_{p-r}(y)$); that is, here the size of the DNF will be $\mathrm{poly}(n)\cdot 2^n\cdot \mathrm{E}[2^{B_p\cdot(n/p)}]$, which is upper bounded by $2^{1.51n}+\Pr[B_p>0.51p]\cdot 2^{2n}=2^{1.51n}+2^{2n-\Omega(p)}$. ∎

Lower bounds for bilinear functions. We do not know whether the bilinear functions $F^{2,n}_{\mathtt{leq}}$ and $F^{2,n}_{\mathrm{mod}\,p}$ (for $p<n$) require depth-two circuits of size that is significantly larger than 2^n. This is quite annoying but of no real significance, since we seek lower bounds of the form $\exp(\Omega(tn))$ for t-linear functions. Still, the following problem is of natural interest.

Problem C.6 (does $F^{2,n}_{\mathtt{leq}}$ require significantly larger CNFs than parity?) *Is it true that the bilinear function $F^{2,n}_{\mathtt{leq}}(x,y)=\sum_{i\le j\le n}x_iy_j$ has no depth-two circuits of size smaller than $2^{1.5n}$? Ditto for the bilinear function $F^{2,n}_{\mathrm{mod}\,p}$, when $p\le n$.*[34]

In fact, the same may be asked of almost all bilinear functions.

Lower bounds for t-linear functions. Turning to larger values of t,we proceed in two steps. The first step (captured by Proposition C.7) reduces establishing lower bounds on the size of depth-two circuits computing a t-linear function F to establishing lower bounds on the number of variables that influence the linear function that is obtained from F by fixing random values to all other $t-1$ blocks of variables. The second step (represented by the subsequent propositions) establishes lower bounds of the latter form for various t-linear functions.

Proposition C.7 (exponential lower bounds for some multilinear functions): *Let $F:(\{0,1\}^n)^t\to\{0,1\}$ be a t-linear function, $n_1,...,n_t\ge 0$ and $\epsilon_1,...,\epsilon_t\ge 0$ such that $\sum_{j\in[t]}n_j\ge 1$ and $\sum_{j\in[t]}\epsilon_j<1/4$. Suppose that for each $j\in[t]$, with probability at least $1-\epsilon_j$ over the choice of $r^{(1)},...,r^{(t)}\in\{0,1\}^{t\cdot n}$, the residual function $F(r^{(1)},...,r^{(j-1)},x^{(j)},r^{(j+1)},...,r^{(t)})$ depends on at least n_j variables. Then, every depth-two circuit computing F has size at least 2^{m-3}, where $m\stackrel{\mathrm{def}}{=}\sum_{j\in[t]}n_j$.*

The hypothesis holds for almost all t-linear functions with $n_j=(0.5-o(1))n$ and $\epsilon_j=1/5t$ for all $j\in[t]$ (provided $t=\exp(o(n))$, see Proposition C.8). On the other hand, the hypothesis does not hold for t-linear functions F that can be presented as a product of a pair of a multilinear functions (i.e., a t'-linear function and a $(t-t')$-function).[35]

Proof: It will be more convenient to show that neither F nor $F+1$ has a DNF of size smaller than 2^{m-3}. For any $\sigma\in\{0,1\}$, suppose that $F(x^{(1)},...,x^{(t)})+\sigma=\bigvee_{k\in[M]}\phi_k(x^{(1)},...,x^{(t)})$, where each ϕ_k is a non-trivial conjunction. We shall show that $M\ge 2^{m-3}$.

For each $k\in[M]$ and $j\in[t]$, let $D^{(j)}_k$ denote the set of variables in $x^{(j)}$ on which ϕ_k depends, and let $G\stackrel{\mathrm{def}}{=}\{k\in[M]:(\forall j\in[t])\,|D^{(j)}_k|\ge n_j\}$ (denote the set of good ϕ_k's). Letting $F'(x^{(1)},...,x^{(t)})\stackrel{\mathrm{def}}{=}\sigma+\bigvee_{k\in G}\phi_k(x^{(1)},...,x^{(t)})$, we shall prove that $F'+\sigma$ is ϵ-close to $F+\sigma$, where $\epsilon=\sum_{j\in[t]}\epsilon_j$.

Assume, towards the contradiction, that $F'+\sigma=\bigvee_{k\in G}\phi_k$ is ϵ-far from $F+\sigma=\bigvee_{k\in[M]}\phi_k$, and let $B_j\stackrel{\mathrm{def}}{=}\{k\in[M]:|D^{(j)}_k|<n_j\}$ (denote the set of ϕ_k that are bad for j). Note that $G=[M]\setminus(\cup_{j\in[t]}B_j)$, and that a random assignment to all the variables satisfies $\bigvee_{k\in([M]\setminus G)}\phi_k$ with probability greater

[34] Note that $T^{t,n}_{\mathrm{mod}\,p}=\emptyset$ for $p>tn$.

[35] That is, if $F(x^{(1)},...,x^{(t)})$ equals $F_1(x^{(1)},...,x^{(t')})\cdot F_2(x^{(t'+1)},...,x^{(t)})$, then for every $j\in[t']$ with probability at least 1/2 the function F_2 evaluates to 0 under a random assignment to $x^{(t'+1)},...,x^{(t)}$, and in this case the value of the residual F does not depend only any variable in $x^{(j)}$.

than ϵ (since $\Pr[\bigvee_{k\in([M]\setminus G)}\phi_k(r^{(1)},...,r^{(t)})=1]$ equals $\Pr[F(r^{(1)},...,r^{(t)})+\sigma\neq F'(r^{(1)},...,r^{(t)})+\sigma]$, which is greater than ϵ by the contradiction hypothesis). Then, there exists $j\in[t]$ such that, with probability greater than ϵ_j, a random assignment to all the variables satisfies $\bigvee_{k\in B_j}\phi_k$. Fixing such a j and recalling the main hypothesis, it follows that there exists an assignment $r^{(1)},...,r^{(t)}$ such that $\bigvee_{k\in B_j}\phi_k(r^{(1)},...,r^{(t)})=1$ and $f(x^{(j)})\stackrel{\text{def}}{=}F(r^{(1)},...,r^{(j-1)},x^{(j)},r^{(j+1)},...,r^{(t)})$ depends on at least n_j variables (and is linear). Fix this assignment as well as a $k\in B_j$ such that $\phi_k(r^{(1)},...,r^{(t)})=1$. Recalling that $\phi(x^{(j)})\stackrel{\text{def}}{=}\phi_k(r^{(1)},...,r^{(j-1)},x^{(j)},r^{(j+1)},...,r^{(t)})$ depends on less than n_j variables and is not identically 0, we reach a contradiction (because we can set n_j-1 variables of $\phi(x^{(j)})$ such that ϕ is determined to the value 1, and then set the remaining variables such that $f+\sigma$ is 0.[36]

Having established that $F'(x^{(1)},...,x^{(t)})+\sigma=\bigvee_{k\in G}\phi_k(x^{(1)},...,x^{(t)})$ is ϵ-close to $F+\sigma$, where $\epsilon<1/4$, we note that $F+\sigma$ evaluates to 1 with probability at least $\max_{j\in[t]:n_j\geq 1}\{(1-\epsilon_j)\}\cdot 1/2>3/8$, where the first factor lower bounds the probability that assigning random values to the variables $x^{(1)},..,x^{(j-1)},x^{(j+1)},...,x^{(t)}$ of F yields a non-trivial linear function in $x^{(j)}$. It follows that $F'+\sigma$, which is $1/4$-close to $F+\sigma$, evaluates to 1 with probability at least $(3/8)-(1/4)=1/8$. This implies that $\sum_{k\in G}2^{-\ell_k}\geq 1/8$, where $\ell_k=\sum_{j\in[t]}|D_k^{(j)}|\geq\sum_{j\in[t]}n_j=m$ for every $k\in G$. Hence, $|G|\geq 2^{m-3}$. ■

Proposition C.8 (almost all multilinear functions satisfy the hypothesis of Proposition C.7 with linear n_j's): *For every $\epsilon>0$, for almost all t-linear functions $F:(\{0,1\}^n)^t\to\{0,1\}$, it holds that for each $j\in[t]$, with probability at least $1-\exp(-\Omega(\epsilon^2 n))$ over the choice of $r^{(1)},...,r^{(t)}\in\{0,1\}^{t\cdot n}$, the residual function $F(r^{(1)},...,r^{(j-1)},x^{(j)},r^{(j+1)},...,r^{(t)})$ depends on at least $(0.5-\epsilon)\cdot n$ variables.*

Thus, for any $t=\exp(o(n))$, almost all t-linear functions $F:(\{0,1\}^n)^t\to\{0,1\}$, satisfy the hypothesis of Proposition C.7 with $n_j=(0.5-o(1))\cdot n$ and $\epsilon_j=1/5t$ for every $j\in[t]$.

Proof: For simplicity of notation, let $j=1$. For a generic t-linear function F associated with the tensor $T\subseteq[n]^t$ and a generic assignment $r^{(2)},...,r^{(t)}$, we have $F(x^{(1)},r^{(2)},...,r^{(t)})=\sum_{i\in[n]}v_i x_i^{(1)}$, where $v_i=\sum_{(i,i_2,...,i_t)\in T}r_{i_2}^{(2)}\cdots r_{i_t}^{(t)}$. Defining $I_k=\{i\in[n]:r_i^{(k)}=1\}$, it follows that $v_i=|T\cap(\{i\}\times R)|$, where $R\stackrel{\text{def}}{=}I_2\times\cdots\times I_t$. Note that, with probability $1-\exp(-n)$ over the choice of $r^{(2)},...,r^{(t)}\in\{0,1\}^{(t-1)\cdot n}$, the $(t-1)$-dimensional rectangle R is non-empty, and in such a case $v_i=|T\cap(\{i\}\times R)|$ will be odd with probability $1/2$ when T is selected at random. Thus, with probability $1-\exp(-\epsilon^2 n)$ over the random choice of both $r^{(1)},...,r^{(t)}\in\{0,1\}^{t\cdot n}$ and a t-linear function $F:[n]^t\to\{0,1\}$, the residual function $F(x^{(1)},r^{(2)},...,r^{(t)})$ will depend on at least $(n/2)-\epsilon n$ variables. The same holds to any other $j\in[t]$, and the claim follows by an averaging argument. ■

Proposition C.9 (F_{mod} satisfies the hypothesis of Proposition C.7 with linear n_j's): *Let $p\leq n$, and suppose that p is a prime such that 2 is a primitive root modulo p* (i.e., 2 generates $\mathbb{Z}_p^*$). *Then, for any $\epsilon>0$ and each $j\in[t]$, with probability at least $1-4(t-1)\cdot 2^{-p}-2\cdot e^{-2\epsilon^2 p}$ over the choice of $r^{(1)},...,r^{(t)}\in\{0,1\}^{t\cdot n}$, the residual function $F_{\text{mod}\,p}^{t,n}(r^{(1)},...,r^{(j-1)},x^{(j)},r^{(j+1)},...,r^{(t)})$ depends on at least $(0.5-\epsilon)\cdot n-((p-n)\bmod p)$ variables.*[37]

In particular, using $p=p(n)$ and $t=o(2^p)$, with probability at least $1-\exp(-\sqrt{p})$ over the choice of $r^{(1)},...,r^{(t)}\in\{0,1\}^{t\cdot n}$, the residual function $F_{\text{mod}\,p}^{t,n}(r^{(1)},...,r^{(j-1)},x^{(j)},r^{(j+1)},...,r^{(t)})$ depends on at least $(0.5-o(1))\cdot n$ variables.

Proof: Since the function is symmetric, it suffices to establish the claim for $j=1$. We shall also start by considering the case that $p=n$.

[36] That is, we can set $r^{(j)}$ such that $\phi(r^{(j)})=1$ but $f(r^{(j)})=\sigma$, which means that $\phi_k(r^{(1)},...,r^{(j-1)},r^{(j)},r^{(j+1)},...,r^{(t)})=1$ whereas $F(r^{(1)},...,r^{(j-1)},r^{(j)},r^{(j+1)},...,r^{(t)})=\sigma$, which contradicts the hypothesis that $F(x^{(1)},...,x^{(t)})+\sigma=\bigvee_{k\in[M]}\phi_k(x^{(1)},...,x^{(t)})$.

[37] Here, e denotes the natural logarithm base.

For simplicity of notation, we shall replace $[n]$ by $\mathbb{Z}_n$. For every $i \in \mathbb{Z}_n$, let $F^{t,n}_{\bmod n,i}$ denote the function associated with the tensor $T^{(t,n)}_i = \{(i_1,...,i_t) \in \mathbb{Z}^t_n : \sum_{j\in[t]} i_j \equiv i \pmod n\}$. Indeed, $T^{(t,n)}_0 \equiv T^{t,n}_{\bmod n}$, and we can write $F^{t,n}_{\bmod n,i}$ as

$$\sum_{i\in[n]} \sum_{(i_2,...,i_t)\in T^{(t-1,n)}_{n,n-i}} x^{(2)}_{i_2} \cdots x^{(t)}_{i_t} \cdot x^{(1)}_i. \tag{15}$$

We shall prove, by induction on $t \geq 2$, that, with probability at least $1-4(t-1)\cdot 2^{-n}$ over the uniform choice of $r^{(2)},...,r^{(t)} \in \{0,1\}^n$, setting $R^{(t)}_i = \sum_{(i_2,...,i_t)\in T^{(t-1,n)}_{n,n-i}} r^{(2)}_{i_2} \cdots r^{(t)}_{i_t}$ (for each $i \in \mathbb{Z}_p$) yields a distribution $(R^{(t)}_1,...,R^{(t)}_n)$ of min-entropy at least $n-1$ (i.e., no outcome occurs with probability greater than $2^{-(n-1)}$). Observing that the residual function $F^{t,n}_{\bmod n,i}(x^{(1)},r^{(2)},...,r^{(t)})$ depends on $x^{(1)}_i$ if and only if $R^{(t)}_i = 1$, we conclude that, with probability at least $1-4(t-1)\cdot 2^{-n} - 2e^{-2\epsilon^2 n}$, the residual function depends on at least $(0.5-\epsilon)\cdot n$ variables.

The base of the induction (at $t=2$) holds, since in that case $R^{(2)}_1,...,R^{(2)}_n$ is merely a permutation of the sequence $r^{(2)}$ (i.e., $i_2 \in T^{(1,n)}_{n,n-i}$ iff $i_2 \equiv n-i \pmod n$). In the induction step, we use the fact that $R^{(t+1)}_i$ can be written as $\sum_{k\in\mathbb{Z}_n} r^{(2)}_k F^{t-1,n}_{\bmod n,n+i-k}(r^{(3)},...,r^{(t+1)})$, which is distributed identically to $\sum_{k\in\mathbb{Z}_n} B_k R^{(t)}_{i-k}$, where the B_k's are IIDs each uniformly distributed in $\{0,1\}$. Letting R denote an n-by-n matrix in which the i^{th} column is the result of i downward rotations of $(R^{(t)}_1,...,R^{(t)}_n)^\top$, it holds that $(R^{(t+1)}_1,...,R^{(t+1)}_n)$ is distributed identically to bR, where $b=(B_1,...,B_n)$.

We now invoke a result of [29] that states that if R (which is a shifted matrix (of dimension n with 2 generating $\mathbb{Z}^*_n$)) is neither identically zero nor identically one, then it has rank at least $n-1$. Recalling that the induction hypothesis asserts that, with probability at least $1-4(t-1)2^{-n}$, the vector $(R^{(t)}_1,...,R^{(t)}_n)$ has min-entropy at least $n-1$, it follows that with probability $1-4(t-1)2^{-n}-2\cdot 2\cdot 2^{-n}$ the matrix R has rank at least $n-1$. In that case bR has min-entropy at least $n-1$ (since b us uniformly distributed in $\{0,1\}^n$).

This complete the proof for the case that $p=n$. The case of $p<n$ is treated by observing that $F^{t,n}_{\bmod p}(x^{(1)},...,x^{(t)})$ equals $F^{t,p}_{\bmod p}(y^{(1)},...,y^{(t)})$, where $y^{(j)}_r = \sum_{i\in[n]:i\equiv r \pmod p} x^{(j)}_i$, for every $j\in[t]$ and $r\in[p]$. Thus, fixing the values of $x^{(2)},...,x^{(t)}$ at random, means doing so to $y^{(2)},...,y^{(t)}$, and if the residual function $F^{t,p}_{\bmod p}$ depends on k of the variables $y^{(1)}$, then the corresponding residual function $F^{t,n}_{\bmod p}$ depends on at least $k\cdot\lfloor n/p\rfloor$ of the variables $x^{(1)}$. Noting that $k\cdot\lfloor n/p\rfloor = (k/p)\cdot n - ((p-n) \bmod p)$, the claim follows. ■

Proposition C.10 ($F_{\mathtt{tet}}$ satisfies the hypothesis of Proposition C.7 with linear n_j's): *For each $j\in[t]$, with probability at least $1-n^{t-2}\cdot\exp(-\Omega(n))$ over the choice of $r^{(1)},...,r^{(t)} \in \{0,1\}^{t\cdot n}$, the residual function $F^{t,n}_{\mathtt{tet}}(r^{(1)},...,r^{(j-1)},x^{(j)},r^{(j+1)},...,r^{(t)})$ depends on at least $n/5$ variables.*

Proof: Since the function is symmetric, it suffices to establish the claim for $j=1$. For simplicity, we shall consider the related functions $F^{(t,n)}_m$, for $m=0,1,...,n/2$, that correspond to the tensors $T^{(t,n)}_m = \{(i_1,...,i_t)\in\{0,1,...,m\}^t : \sum_{j\in[t]} i_j \leq m\}$. Clearly, if $F^{(t,n)}_{n/2}$ satisfies the hypothesis of Proposition C.7 with linear n_j's, then so does $F^{t,n}_{\mathtt{tet}}$. For every $m\in\{0,1,...,n/2\}$, we write the function $F^{(t,n)}_m(x^{(1)},x^{(2)},...,x^{(t)})$ as

$$\sum_{i\in[n]} F^{(t-1,n)}_{m-i}(x^{(2)},...,x^{(t)})\cdot x^{(1)}_i. \tag{16}$$

We shall prove, by induction on $t\geq 2$, setting $X^{(t)}_i = F^{(t-1,n)}_i(r^{(2)},...,r^{(t)})$ (for each $i\in\{0,1,...,n/2\}$), where $r^{(2)},...,r^{(t)}\in\{0,1\}^n$ are uniformly distributed, yields a distribution $(X^{(t)}_0,...,X^{(t)}_{n/2})$ of min-entropy at least $(n/2)+1-(t-2)\cdot\log_2 n$ (i.e., no outcome occurs with probability greater than $n^{t-2}\cdot 2^{-((n/2)+1)}$). Observing that the residual function $F^{(t,n)}_{n/2}(x^{(1)},r^{(2)},...,r^{(t)})$ depends on $x^{(1)}_i$ if and only if $X^{(t)}_i = 1$, we conclude that, with probability at least $1-n^{t-2}\cdot\exp(-\epsilon^2 n)$, the residual function depends on at least $(0.5-\epsilon)\cdot(n/2)$ variables.

The base of the induction (at $t = 2$) holds, since in that case $X_0^{(2)}, X_1^{(2)}, ..., X_{n/2}^{(2)}$ is uniformly distributed (since, $X_i^{(t)} = \sum_{k=0}^{i} r_k^{(2)}$). In the induction step, we use the fact that $X_i^{(t+1)}$ can be written as $\sum_{k=0}^{i} X_k^{(t)} r_{i-k}^{(t+1)}$. For $m = (n/2) + 1$, letting R denote an m-by-m matrix in which the i^{th} row equals $(X_0^{(t)}, X_1^{(t)}, ..., X_{i-1}^{(t)}, 0, ..., 0)$, it holds that $(X_0^{(t+1)}, X_1^{(t+1)}, ..., X_m^{(t+1)})$ is distributed identically to Rb, where $b = (r_0^{(t+1)}, r_1^{(t+1)}, ..., r_m^{(t+1)})$. We now make the following observations:

1. If $X_0^{(t)}, X_1^{(t)}, ..., X_{n/2}^{(t)}$ were uniformly distributed, then for every $i \in [m]$, the matrix R would have had rank $m - i + 1$ with probability 2^{-i}. This is because R has rank $m + 1 - i$ if and only if $(X_0^{(t)}, X_1^{(t)}, ..., X_{i-1}^{(t)}) = 0^{i-1}1$.
2. Since $X_0^{(t)}, X_1^{(t)}, ..., X_{n/2}^{(t)}$ has min-entropy $m - (t-2) \cdot \log_2 n$, the matrix R has rank $m + 1 - i$ with probability at most $n^{t-2} \cdot 2^{-i}$.
3. If R has rank r, then Rb has min-entropy r, which implies that $\Pr[Rb{=}v] \leq 2^{-r}$ for every v.

Thus, for any $v \in \{0,1\}^{m+1}$,

$$\begin{aligned}\Pr[Rb = v] &= \sum_{i=0}^{m} \Pr[\text{rank}(R) = m+1-i] \cdot \Pr[Rb = v \mid \text{rank}(R) = m+1-i] \\ &\leq \sum_{i=0}^{m} (n^{t-2} \cdot 2^{-i}) \cdot 2^{-(m+1-i)}\end{aligned}$$

which is upper bounded by $n^{t-1} \cdot 2^{-(m+1)}$, and the induction claim follows. ■

Proposition C.11 ($F_{\texttt{leq}}$ satisfies the hypothesis of Proposition C.7 with linear n_j's): *For $t < o(\sqrt{\log n})$ and each $j \in [t]$, with probability at least $1 - o(1/t)$ over the choice of $r^{(1)}, ..., r^{(t)} \in \{0,1\}^{t \cdot n}$, the residual function $F_{\texttt{leq}}^{t,n}(r^{(1)}, ..., r^{(j-1)}, x^{(j)}, r^{(j+1)}, ..., r^{(t)})$ depends on at least $(0.25 - o(1)) \cdot n$ variables. Furthermore, for $j \in \{1, t\}$, a lower bound of $(0.5 - o(1)) \cdot n$ holds.*

Combined with Proposition C.7, the furthermore claim implies that for any $t \in [3, o(\sqrt{\log n})]$, the t-linear function $F_{\texttt{leq}}^{t,n}$ has no depth-two circuits of size $2^{n+0.24 \cdot (t-2) \cdot n}$. On the other hand, for $t > 4n$, the function $F_{\texttt{leq}}^{t,n}$ evaluates to 1 with exponentially vanishing probability, and (by Eq. (17) (below)) this implies that $F_{\texttt{leq}}^{t,n}$ violates the hypothesis of Proposition C.7.

Proof: The key observation is that for every $j \in [t]$, it holds that

$$F_{\texttt{leq}}^{t,n}(x^{(1)}, ..., x^{(t)}) = \sum_{i \in [n]} F_{\texttt{leq}}^{j-1,i}(x_{[1,i]}^{(1)}, ..., x_{[1,i]}^{(j-1)}) \cdot F_{\texttt{leq}}^{t-j,n-i+1}(x_{[i,n]}^{(j+1)}, ..., x_{[i,n]}^{(t)}) \cdot x_i^{(j)}, \tag{17}$$

where $x_{[a,b]}^{(k)} = (x_a^{(k)}, ..., x_b^{(k)})$. It will also be useful to let $x_{[a,b]}^{([c,d])}$ denote the variable sequence $x_{[a,b]}^{(c)}, ..., x_{[a,b]}^{(d)}$. Thus, Eq. (17) can be written as

$$F_{\texttt{leq}}^{t,n}(x_{[1,n]}^{([1,t])}) = \sum_{i \in [n]} F_{\texttt{leq}}^{j-1,i}(x_{[1,i]}^{([1,j-1])}) \cdot F_{\texttt{leq}}^{t-j,n-i+1}(x_{[i,n]}^{([j+1,t])}) \cdot x_i^{(j)}, \tag{18}$$

and we are interested in the distribution of the pair of n-bit long sequences $(F_{\texttt{leq}}^{j-1,i}(x_{[1,i]}^{([1,j-1])}))_{i \in [n]}$ and $(F_{\texttt{leq}}^{t-j,n-i+1}(x_{[i,n]}^{([j+1,t])}))_{i \in [n]}$, when $x_{[1,n]}^{([1,t])}$ are assigned random values. Note that these two sequences are independent of one another (since the first depends only on $x^{(1)}, ..., x^{(j-1)}$ whereas the second depends only on $x^{(j+1)}, ..., x^{(t)}$). Hence, if *in each of these two sequences almost each element is 1 with probability approximately* $1/2$ *and this holds also conditioned on the value of almost each other element in the sequence*, then the fraction of influential variables in $x^{(j)}$ would be approximately $1/4$.

In general, for every $t' \geq 1$, we shall be interested in the behavior of the distribution of the (n-bit long) sequence $(F_{\mathtt{leq}}^{t',i}(x_{[1,i]}^{([1,t'])})_{i\in[n]}$, when $x^{(1)},...,x^{(t')}$ are uniformly and independently distributed in $\{0,1\}^n$. Note that this corresponds directly to the sequence $(F_{\mathtt{leq}}^{j-1,i}(x_{[1,i]}^{([1,j-1])}))_{i\in[n]}$ (by setting $t'=j-1$), and also represents the sequence $(F_{\mathtt{leq}}^{t-j,n-i+1}(x_{[i,n]}^{([j+1,t])}))_{i\in[n]}$ (by setting $t'=t-j$ and replacing $[i,n]$ with $[i]$ (and $n-i+1$ with i)).

We first observe that for $t'=1$ the foregoing sequence is uniformly distributed in $\{0,1\}^n$, since $(F_{\mathtt{leq}}^{1,i}(x_{[1,i]}^{([1,1])})_{i\in[n]}$ equals $(\sum_{k\in[i]} x_k^{(1)})_{i\in[n]}$. A key observation regarding $t'>1$ is that

$$F_{\mathtt{leq}}^{t',i}(x_{[1,i]}^{([1,t'])}) = \sum_{k\in[i]} F_{\mathtt{leq}}^{t'-1,k}(x_{[1,k]}^{([1,t'-1])}) \cdot x_k^{(t')}. \tag{19}$$

In general, it is useful to realize that we are dealing with a sequence of sequences of random variables, which are each defined on top of the previous one. That is, let $X_i^{[t']} \stackrel{\text{def}}{=} F_{\mathtt{leq}}^{t',i}(x_{[1,i]}^{([1,t'])})$, and note that Eq. (19) asserts that $X_i^{[t']} = \sum_{k\in[i]} X_k^{[t'-1]} \cdot R_k^{(t')}$, where $(R_1^{(t')},...,R_n^{(t')})$ is uniformly distributed in $\{0,1\}^n$ independently of anything else. (Indeed, we may also introduce dummy $X_k^{[0]}$'s set to 1, and write $X_i^{[1]} = \sum_{k\in[i]} X_k^{[0]} \cdot R_k^{(1)}$.)

We shall prove that, for adequate functions $\ell_{t'} : (0,1] \to \mathbb{N}$, it holds that *for every* $i_2 \geq i_1 + \ell_{t'}(\epsilon)$ *and every* $\sigma,\tau \in \{0,1\}$:

$$\Pr\left[X_{i_2}^{[t']} = X_{i_1}^{[t']} + \tau \,\middle|\, X_{i_1}^{[t']} = \sigma \right] \leq 0.5 + \epsilon \tag{20}$$

In particular, this means that $\Pr[X_{i_2}^{[t']}=1] \in [0.5 \pm \epsilon]$. We note that the case of $\tau=1$ (in Eq. (20)) is actually trivial, since by Eq. (21)-(22), it suffices to show that $\Pr[\sum_{k=i_1+1}^{i_2} X_k^{[t'-1]} \cdot R_k^{(t')}=1] \leq 1/2$, which just holds by the independence and uniformity of the $R_k^{(t')}$'s. Also note that we have already shown that Eq. (20) holds for $t'=1$ with $\ell_1 \equiv 1$. We shall proceed by induction on t'. The key observation is that, for any $i_2 > i_1$, it holds that

$$\begin{aligned} X_{i_2}^{[t']} &= \sum_{k=1}^{i_2} X_k^{[t'-1]} \cdot R_k^{(t')} &(21)\\ &= X_{i_1}^{[t']} + \sum_{k=i_1+1}^{i_2} X_k^{[t'-1]} \cdot R_k^{(t')}. &(22) \end{aligned}$$

Thus, if the sequence $X_{i_1+1}^{[t'-1]},...,X_{i_2}^{[t'-1]}$ is not all zeros, then $\Pr[X_{i_2}^{[t']} = X_{i_1}^{[t']}] = 1/2$. So we will be done if i_2 is sufficiently larger than i_1 such that the former condition holds with probability at least $1-\epsilon$. For $t'=2$, this happens whenever $i_2 > i_1 + \log_2(1/\epsilon)$, since the sequence $X_{i_1+1}^{[1]},...,X_{i_2}^{[1]}$ is uniformly distributed in $\{0,1\}^{i_2-i_1}$. For general $t'>2$, we use the induction hypothesis regarding the sequence $X_{i_1+1}^{[t'-1]},...,X_{i_2}^{[t'-1]}$, which asserts that, in intervals of length $\ell_{t'-1}(\epsilon)$, value-changes occur with probability at least $0.5-\epsilon > 0.4$. Intuitively, this means that $\ell_{t'}(\epsilon) = O(\ell_{t'-1}(\epsilon) \cdot \log(1/\epsilon))$ should do, but this intuition is based on the false assumption that what happens within disjoint intervals (of length $\ell_{t'-1}(\epsilon)$) is statistically independent. Yet, as shown next, picking $\ell_{t'}(\epsilon) = O(\ell_{t'-1}(\epsilon/O(1)) \cdot \epsilon^{-1})$ will do; that is, in this case, with probability at least $1-\epsilon$, the sequence $X_{i_1+1}^{[t'-1]},...,X_{i_2}^{[t'-1]}$ will not be the all-zero sequence. This is a special case of the following more general claim.[38]

Technical Claim: For $\gamma < \delta/4$ and $m' < \gamma m/3$, let $Z_1,...,Z_m$ be an arbitrary sequence of (possibly dependent) 0-1 random variables. Suppose that, for every $j_2 \geq j_1 + m'$ and every $\sigma,\tau \in \{0,1\}$, it holds that $\Pr[Z_{j_2} = Z_{j_1} + \tau | Z_{j_1} = \sigma] \leq 0.5 + \gamma$. Then,

$$\Pr\left[\sum_{i\in[m]} Z_i \notin [(0.5 \pm \delta) \cdot m] \right] < \frac{12\gamma}{\delta^2}$$

[38] Use $\gamma = \epsilon/50$ and $\delta = 0.49$, and set $m = i_2 - i_1$, $m' = \ell_{t'-1}(\epsilon/50) < \epsilon m/150$, and $Z_i = X_{i_1+i}^{[t'-1]}$ for every $i \in [m]$.

Proof: Let $S = \sum_{i\in[m]} Z_i$ and $\mu = \mathrm{E}[S]$, and note that $|\mu - (m/2)| \leq \gamma m + m' < \delta m/2$. Applying Chebyshev's Inequality, we have

$$\begin{aligned}\Pr[|S-\mu| > \delta m/2] &\leq \frac{\mathrm{Var}[S]}{\delta^2 m^2/4} \\ &= \frac{4}{\delta^2 m^2} \cdot \sum_{j_1,j_2\in[m]} \left(\mathrm{E}[Z_{j_1} Z_{j_2}] - \mathrm{E}[Z_{j_1}] \cdot \mathrm{E}[Z_{j_2}]\right). \end{aligned} \tag{23}$$

The contribution of the pairs that are at distance at most m' apart totals in less than $m \cdot (2m'+1) < 3m'm$. As for the other (j_1, j_2) pairs, each has a contribution of

$$\begin{aligned}\Pr[Z_{j_1} Z_{j_2} = 1] - \Pr[Z_{j_1} = 1] \cdot \Pr[Z_{j_2} = 1] &= \Pr[Z_{j_1} = 1] \cdot (\Pr[Z_{j_2} = 1 | Z_{j_1} = 1] - \Pr[Z_{j_2} = 1]) \\ &< 2\gamma \end{aligned}$$

Thus, Eq. (23) is upper bounded by $\frac{4\cdot(3m'm+2\gamma m^2)}{\delta^2 m^2} < \frac{12\gamma}{\delta^2}$, and the claim follows. ■

Let us re-cap: By the Technical Claim, if $i_2 \geq i_1 + \ell_{t'}(\epsilon)$, where $\ell_{t'}(\epsilon) = O(\ell_{t'-1}(\epsilon/O(1))/\epsilon)$, then, with probability at least $1-\epsilon$, the sequence $X_{i_1+1}^{[t'-1]}, ..., X_{i_2}^{[t'-1]}$ will not be the all-zero sequence, and conditioned on that event $X_{i_2}^{[t]} = X_{i_1}^{[t]}$ with probability 1/2. This establishes the induction claim for t'; that is, for every $i_2 \geq i_1 + \ell_{t'}(\epsilon)$, the probability that $X_{i_2}^{[t']} = X_{i_1}^{[t']}$, conditioned on any value of $X_{i_1}^{[t']}$, is between $0.5-\epsilon$ and $0.5+\epsilon$. Note that we are using $\ell_{t'}(\epsilon) = O(\ell_{t'-1}(\epsilon/O(1))/\epsilon)$, which solves to $\ell_{t'}(\epsilon) = \exp(O(t')^2 + O(t'\log(1/\epsilon)))$.

We are almost done. Applying the Technical Claim with $\gamma = \epsilon/20$, $\delta = \epsilon^{1/3}$, while setting $m = n$, $m' = \ell_{t'}(\epsilon/20) < \epsilon n$ (which is possible for $t = o(\sqrt{n})$ and some $\epsilon = o(1)$), and $Z_i = X_i^{[t']}$ for every $i \in [n]$, we conclude that, with probability at least $1 - \epsilon^{1/3}$, the sequence $X_1^{[t'-1]}, ..., X_n^{[t'-1]}$ has at least $(0.5 - \epsilon^{1/3}) \cdot n$ ones. This does not quite finish the entire proof, because it could hypothetically be that (when $x_{[1,n]}^{([1,t])}$ are assigned random values) the ones in the sequences $(F_{\mathtt{leq}}^{j-1,i}(x_{[1,i]}^{([1,j-1])}))_{i\in[n]}$ and $(F_{\mathtt{leq}}^{t-j,n-i+1}(x_{[i,n]}^{([j+1,t])}))_{i\in[n]}$ (almost) always reside in different locations. Of course, this cannot be the case, but proving this fact requires a small generalization of the Technical Claim: Specifically, under the same conditions as in the claim, one can show that for every set $S \subseteq [m]$,

$$\Pr\left[\sum_{i\in S} Z_i \notin [(0.5 \pm \delta) \cdot |S|]\right] < \frac{12\gamma m}{\delta^2 |S|}$$

Now, we can first fix at random the sequence $(F_{\mathtt{leq}}^{j-1,i}(x_{[1,i]}^{([1,j-1])}))_{i\in[n]}$, let S denote the set of indices assigned the value 1, and now set at random the sequence $(F_{\mathtt{leq}}^{t-j,n-i+1}(x_{[i,n]}^{([j+1,t])}))_{i\in[n]}$. Finally, the proposition follows. ■

Problem C.12 (Improving Proposition C.11): *Does a statement analogous to Proposition C.11 holds for higher values of t? Specifically, does it hold for $t = \Omega(\log n)$ rather than for all $t = o(\sqrt{\log n})$?*

Corollary C.13 (exponential lower bounds for almost all functions and for $F_{\mathtt{leq}}, F_{\mathtt{tet}}$ and F_{mod}): *For $t = \exp(o(n))$, almost all t-linear functions $F : (\{0,1\}^n)^t \to \{0,1\}$ require depth-two circuits of size at least $2^{(0.5-o(1))\cdot tn}$. Ditto for $F_{\mathrm{mod}\, n}^{t,n}$ if $t = o(2^n)$ and n is a prime with 2 as a primitive root modulo n. Likewise, if $t = o(\sqrt{\log n})$, then $F_{\mathtt{leq}}^{t,n}$ require depth-two circuits of size at least $2^{0.5+(0.25-o(1))\cdot tn}$, and if $t < n/\log_2 n$, then $F_{\mathtt{tet}}^{t,n}$ require depth-two circuits of size at least $2^{tn/5}$.*