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

Oded Goldreich Avi Wigderson

September 29, 2019

Abstract

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 $\mathtt{AN}(\cdot)$, is the maximum of these two measures (i.e., the maximum between the **a**rity of the gates and the **n**umber 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 multilinear functions*. The resulting circuits have size that is exponential in the new complexity measure. Hence, lower bounds on the new complexity measure yield size lower bounds on a restricted class of depth-three Boolean circuits (for computing multilinear functions). Such lower bounds are a sanity check for our conjecture that multilinear 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 $\mathtt{AN}_2(\cdot)$ denote the complexity measure $\mathtt{AN}(\cdot)$, 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 $\mathtt{AN}(F) \leq \mathtt{AN}_2(F) = O((tn)^{t/(t+1)})$.
- For almost all t-linear function F, it holds that $\mathtt{AN}_2(F) \geq \mathtt{AN}(F) = \Omega((tn)^{t/(t+1)})$.
- There exists a bilinear function F such that $\mathtt{AN}(F) = O(\sqrt{n})$ but $\mathtt{AN}_2(F) = \Omega(n^{2/3})$.

The main open problem posed in this paper is proving that $\mathtt{AN}_2(F) \geq \mathtt{AN}(F) = \Omega((tn)^{t/(t+1)})$ holds for an explicit t-linear function F, with $t \geq 2$. 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.

An early version of this work appeared as TR13-043 of *ECCC*. The current revision is quite substantial (cf. [11]). In particular, the original abstract was replaced, the appendices were omitted, notations were changed, some arguments were elaborated, and updates on the state of the open problems were added (see, most notably, the progress made in [9]).

1 Introduction

The introduction contains an extensive motivation for the model of arithmetic circuits that is studied in the paper. Readers who are only interested in this 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 1.2.

1.1 The general context

Strong lower bounds on the size of constant-depth Boolean circuits computing parity and other explicit functions (cf., e.g., [34, 12] and [26, 29]) 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., [16, Chap. 11]). It seems that such lower bounds cannot be obtained by the standard interpretation of either the random restriction method [7, 12, 34] or the approximation by polynomials method [26, 29]. Many experts have tried other approaches (cf., e.g., [14, 17])[1], and some obtained encouraging indications (i.e., results that refer to restricted models, cf., e.g., [23]); 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, e.g., [11, Apdx A]) and presenting an explicit function that does not have linear-size circuits of logarithmic depth (see Valiant [32]). 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. [8, Sec. 8.3]), which in turn can be constructed based on strong average-case versions of circuit lower bounds; cf. [21, 22].

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 class of functions (i.e., multilinear functions of small degree), and on a restricted type of circuits that emerges rather naturally when considering the computation of such functions.

1.2 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

[1] The relevance of the Karchmer and Wigderson approach [17] to constant-depth circuits is stated explicitly in [18, Sec. 10.5].

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 [23]. 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 $t = t(n) \leq \mathrm{poly}(n)$.*

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* Boolean circuits of size $2^{tn/4}$ (see [11, Apdx C.3]).

Merely proving Conjecture 1.1 may not necessarily yield a major breakthrough in the state-of-art regarding circuit 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. (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. [23], which proves an exponential in t lower bound on the size of depth-three arithmetic circuits (when $n = 4$)).)

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).[2] The same holds for any function that corresponds either to a rectangular tensor (i.e., $T = I_1 \times \cdots \times I_t$, 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 [30]). On the other hand, it seems good to stick to as "simple" tensors as possible so

[2]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.

as to facilitate their analysis (let alone have the corresponding multilinear function be computable in exponential-time (i.e., in $\mathcal{E}$)).[3]

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

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

(having the corresponding tensor $T^{t,n}_{\mathtt{leq}} = \{(i_1, ..., i_t) \in [n]^t : i_1 \leq i_2 \leq \cdots \leq i_t\}$). Note that this function is polynomial-time computable (e.g., via dynamic programming),[4] and that $t = 1$ corresponds to `Parity`. Unfortunately, for every constant $t \geq 2$, 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 to 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).

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

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

(The shorthand `tet` was intended to stand for tetrahedon, since the geometric image of one eighth of $T^{3,n}_{\mathtt{tet}}$ resembles a "slanted tetrahedon". Indeed, $T^{3,n}_{\mathtt{tet}}$ as a whole looks more like a regular octahedon.)

Note that the functions $F^{t,n}_{\mathtt{tet}}$ and $F^{t,n}_{\mathrm{mod}\, p}$ are also computable in polynomial-time.[5] 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 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,

[3] 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 $n^t \ll \exp(tn)$ oracle calls.

[4] 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 $(F^{t'+1,i}_{\mathtt{leq}}(x^{(1)}_{[1,i]}, ..., x^{(t'+1)}_{[1,i]}))_{i \in [n]}$.

[5] Again, we use dynamic programming, but here we apply it to generalizations of these functions. Specifically, let $T^{t,n,d}_{\mathtt{tet}} = \{(i_1, ..., i_t) \in [n]^t : \sum_{j \in [n]} |i_j - (n/2)| \leq d\}$ and note that the associated 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 $F^{t,n,r}_{\mathrm{mod}\, p}(x^{(1)}, ..., x^{(t)}) = \sum_{i \in [n]} F^{t-1,n,r-i}_{\mathrm{mod}\, p}(x^{(1)}, ..., x^{(t-1)}) \cdot x^{(t)}_i$.

whereas the failure of $F^{2,n}_{\mathtt{tet}}$ is to the fact that $T^{2,n}_{\mathtt{tet}}$ is very similar to $T^{2,n}_{\mathtt{leq}}$ (i.e., each fourth of $T^{2,n}_{\mathtt{tet}}$ is isomorphic to $T^{2,n}_{\mathtt{leq}}$ (under rotation and scaling)). But these weaknesses do not seem to propagate to the trilinear versions (e.g., the eighthes of the tensor $T^{3,n}_{\mathtt{tet}}$ are not isomorphic to $T^{3,n}_{\mathtt{leq}}$).

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 1.3 and 1.4). Subsequent work [9] was more successful in that regard.

1.3 Design by 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 (denoted F) as a polynomial (denoted H) in some auxiliary multilinear functions (i.e., 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 *syntactically 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 *defined* 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.)[6]

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 1.4). Indeed, D-canonical circuits seem natural in the context of computing multilinear 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

[6] 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 [11, Apdx 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 (i.e., F_i's) level).

function 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 function 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 function be computed by composition of a polynomial and multilinear functions (and that this composition corresponds to a syntactically 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 fixed $t \geq 2$, 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 $t \leq \mathrm{poly}(n)$.*

We mention that subsequent work of Goldreich and Tal [9] proved an $\exp(\widetilde{\Omega}(n^{2/3}))$ lower bound on the size of D-canonical circuits computing some explicit trilinear functions (e.g., $F^{3,n}_{\mathtt{tet}}$).

1.4 Design by 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 (see Theorem 2.3). 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 Boolean circuit has depth-three.

Assuming we want to use s auxiliary functions or arity s, the basic idea is to use s non-deterministic guesses for the values of these s 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 multilinear functions, $F_1, ..., F_s$;
2. Express F as well as each of the other F_i's 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 *defined* to be $\exp(\Theta(s))$.

Note that, here (i.e., 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. (Hence, the composition is "nested".) 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 fixed $t \geq 2$, 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 $t \leq \mathrm{poly}(n)$.*

The subsequent work of Goldreich and Tal [9] establishes an $\exp(\widetilde{\Omega}(n^{0.6}))$ lower bound on the size of ND-canonical circuits computing the trilinear function $F^{3,n}_{\mathtt{tet}}$ and an $\exp(\widetilde{\Omega}(n^{2/3}))$ lower bound on the size of ND-canonical circuits computing some explicit 4-linear functions. It does so by following the path suggested in the original version of this work [11], where we wrote:

> 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).

1.5 The underlying models of arithmetic circuit and AN-complexity

Underlying the two models of canonical circuits (discussed in Section 1.3 and 1.4) is a new model of arithmetic circuits (for computing multilinear functions). Specifically, the expressions representing the value of (the target and auxiliary) functions in terms of the values of auxiliary functions and original variables correspond to gates in a circuit. These gates can compute arbitrary polynomials (as long as the multilinear condition is satisfied). In the case of D-canonical circuits, the corresponding arithmetic circuits have depth two (i.e., a top gate and at most one layer of intermediate gates), whereas for ND-canonical circuits the corresponding arithmetic circuits have unbounded depth. In both cases, the key complexity measure is the *maximum between the arity of the gates and their number*.

In both cases, canonical Boolean circuits (for computing a multilinear function F) are obtained by presenting a Boolean circuit that emulates the computation of an arithmetic circuit (computing F). Specifically, the D-canonical circuits are obtained by a straightforward implementation of a depth-two circuit that computes F by applying a function H (in the top gate) to intermediate results computed by the intermediate gates (i.e., $F = H(F_1, ..., F_s)$, where F_i is computed by the i^{th} intermediate gate). The ND-canonical circuits are obtained by a Valiant-like (i.e., akin [32]) decomposition of the computation of the (unbounded depth) arithmetic circuit; that is, by guessing and verifying the values of all intermediate gates. In both cases, the size of the resulting Boolean circuit is exponential in the maximum between the *arity of these gates* the *number of gates*. Indeed, this parameter (i.e., the maximum of the two measures) restricts the power of the underlying arithmetic circuits or rather serves as their complexity measure, called *AN-complexity*, where "A" stands for arity and "N" for number (of gates). Let us spell out these two models of arithmetic circuit complexity.

The arithmetic circuits we refer to are directed acyclic graphs that are labeled by arbitrary multilinear functions and variables of the target function (i.e., F). These circuits are restricted to be syntactically multilinear; that is, each gate computes a function that is multilinear in the variables of the target function (i.e., that arguments that depend on variables in the same block are not multiplied by such gates). Specifically, a gate that is labelled by a function H_i and is fed by gates computing the auxiliary functions $F_{i_1}, ..., F_{i_{m'}}$ and m'' original variables, denoted $z_1, ..., z_{m''}$ (out of $x^{(1)}, x^{(2)}, .., x^{(t)}$), computes the function

$$F_i(x^{(1)}, x^{(2)}, ..., x^{(t)}) = H_i(F_{i_1}(x^{(1)}, x^{(2)}, ..., x^{(t)}), ..., F_{i_{m'}}(x^{(1)}, x^{(2)}, ..., x^{(t)}), z_1, ..., z_{m''}).$$

This holds also for the top gate that computes $F = F_0$. In case of depth-two circuits, the top gate is the only gate in the circuit that may be fed by intermediate gates (and we may assume, with no loss of generality, that it is not fed by any variable)[7] As we shall see later (see, e.g., Remark 3.5), the benefit of circuits of larger depth is that they may contain gates that are fed both by other gates and by variables. Let us summarize this discussion and introduce some notation.

- Following [23], 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.

[7] Since such directly fed variables can be replaced by dummy gates that are each fed by the corresponding variable.

- We say that the direct-composition complexity of F, denoted $\mathtt{AN}_2(F)$, is at most s if F *can be computed by a depth-two multilinear circuit with at most s gates that are each of arity at most s.*

- We say that the nested-composition complexity of F, denoted $\mathtt{AN}(F)$, is at most s if F *can be computed by a multilinear 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 $\mathtt{AN}_2(F) = \Theta(\mathtt{AN}(F))$, because all intermediate gates can feed directly to the top gate (since, in this case, all gates compute linear functions).[8] Also note that $\mathtt{AN}_2(F)$ equals the square root of the number of variables on which the *linear* function F depends. In general, $\mathtt{AN}(F) \geq \sqrt{tn}$ for any t-linear function F that depends on all its variables, and $\mathtt{AN}(F) \leq \mathtt{AN}_2(F) \leq tn$ 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 $\mathtt{AN}_2(F) = s$ (resp., $\mathtt{AN}(F) = s$). Thus, all results and open problems presented above (i.e., in Sections 1.3 and 1.4) in terms of canonical (Boolean) circuits are actually results and open problems regarding the complexity of (direct and nested) composition (i.e., $\mathtt{AN}_2(\cdot)$ and $\mathtt{AN}(\cdot)$). 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 $\mathtt{AN}(F) \leq \mathtt{AN}_2(F) = O((tn)^{t/(t+1)})$.

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

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

We stress that the foregoing lower bounds are existential, whereas we seek $\omega(\sqrt{n})$ lower bounds for explicit multilinear functions. (As noted above, this initial goal was achieved by the subsequent work of Goldreich and Tal [9], which establishes an $\mathtt{AN}(F) = \widetilde{\Omega}(n^{2/3})$ for some explicit 4-linear functions F.)

Summary and additional comments. 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). 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

[8] Doing so may increase the arity of the top gate, but this increase is upper-bounded by the number of gates. 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.

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., $\mathtt{AN}$) 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 $\mathtt{AN}_2(\cdot)$ and $\mathtt{AN}(\cdot)$ hold for every field, rather than merely for the binary field. Ditto for the open problems.

1.6 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 [23] 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).[9]

The multilinear circuit model was studied in subsequent works (cf., e.g., [25]); but, to the best of our knowledge, the complexity measure introduced in Section 1.5 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., [24]).

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 [23], 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.[10]

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) al-

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

[10] 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).

most 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.

Decoupling arity from the number of gates. In a work done independently (but subsequent to our initial posting[11]), Hrubes and Rao studied Boolean circuits with general gates [15]. 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. Lastly, we mention that a different relation between the arity and the number of gates is considered in a subsequent work [10] that extends the notion of canonical circuits to constant depth $d > 3$.

1.7 Subsequent work

The subsequent works of Goldreich and Tal [9, 10] were already mentioned several times in the foregoing. While [10] deals with an extension of the current models, the other work (i.e., [9]) is directly related to the current work; specifically, it resolves many of the specific open problems suggested in this work. As done so far, we shall report of the relevant progress whenever reproducing text (of our original work [11]) that raises such an open problem.

1.8 Various conventions

As stated up-front, throughout this paper, when we say that a function $f : \mathbb{N} \to \mathbb{N}$ is $\mathsf{exponential}$, 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 $\mathrm{GF}(2)$, and all arithmetic operations are over this field.[12]

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).

In the case of $t = 2$, the tensor (viewed as an array) is a matrix. In that case, we sometimes denote the variable-blocks by x and y (rather than $x^{(1)}$ and $x^{(2)}$).

1.9 Organization and additional highlights

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

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

[12] However, as stated in Section 1.5, our main results extend to other fields.

in terms of the arithmetic model outlined in Section 1.5; that is, of multilinear circuits with general multilinear gates and a complexity measure, termed AN-complexity, that accounts for both the arity of these gates and their number. The basic definitional issues are discussed in Section 2, upper bounds are presented in Section 3, and lower bounds in Section 4. These sections are the core of the current paper.

We now highlight a few aspects that were either not mentioned in the introducion or mentioned too briefly.

On the connection to matrix rigidity. As mentioned in Section 1.4, we show a connection between proving lower bounds on the AN-complexity of explicit functions and matrix rigidity. In particular, in Section 4.2, we show that $\mathtt{AN}(F^{3,n}_{\mathtt{tet}}) = \Omega(m)$ if there exists an n-by-n Toeplitz matrix that has rigidity m^3 with respect to rank m. This follows from Theorem 4.4, which asserts that *if T is an n-by-n matrix that has rigidity m^3 for rank m, then the corresponding bilinear function F satisfies* $\mathtt{AN}(F) > m$. In Section 4.3 we show that the same holds for a relaxed notion of rigidity, which we call *structured rigidity*. We also show that structured rigidity is strictly separated from the standard notion of rigidity. All these connections were used in the subsequent work of Goldreich and Tal [9].

On further-restricted models. In Section 5, we consider two restricted models of multilinear circuits, which are obtained by imposing constraints on the models outlined in Section 1.5.

1. In Section 5.1, we consider circuits that compute functions without relying on cancellations. We show that such circuits are weaker than the multilinear 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 some explicit functions (i.e., $F^{3,n}_{\mathtt{tet}}$ and $F^{2,n}_{\mathtt{had}}$) without cancellation, whereas one of these functions has AN-complexity $\widetilde{O}(\sqrt{n}$ (i.e., $\mathtt{AN}_2(F^{2,n}_{\mathtt{had}}) = \widetilde{O}(\sqrt{n})$.

2. In Section 5.2 we study a restricted multilinear 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" $F^{2,n}_{\mathtt{leq}}$.

Note that in both these restricted models, we are able to prove a non-trivial lower bound on an explicit function.

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 [23]. 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, and does not depend on variables of other blocks.

- 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 $\left(\bigcup_{i\in[m]} J_i\right)$-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 $\left(\bigcup_{i\in[m]} J_i\right)$-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.*[13] *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 J* (i.e., $J_{i_1} \cap J_{i_2} = \emptyset$ for all $i_1 \neq i_2$ and $\bigcup_{i\in I} J_i = J$). *The function computed by the gate is J'-linear if J' is the union of all the sets that define the linearity of the functions that correspond to the different monomials in H.*

[13] Clearly, w.l.o.g., H is multilinear in its m inputs, since we are considering multiplication over GF(2). However, what we consider next is not the dependency of H on its own inputs, but rather its dependency on the inputs of the circuits as reflected in the composed function $H(F_1, ..., F_m)$. Furthermore, we do not consider this function *per se*, but rather its syntactic form (before cancellations).

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 gates belong to two sets of blocks with empty intersection).

Definition 2.2 (the AN-complexity of multilinear circuits with general gates): *The* arity *of a multilinear circuit is the maximum arity of its* (general) *gates, and the* number of gates *counts only the general gates and not the leaves* (variables). *The* AN-complexity *of a multilinear circuit is the maximum between its arity and the number of its* (general) *gates.*

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

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

Clearly, $\mathtt{AN}_2(F) \geq \mathtt{AN}(F)$ for every multilinear function F. For linear functions F, it holds that $\mathtt{AN}_2(F) \leq 2 \cdot \mathtt{AN}(F)$, 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 (while increasing its arity by at most $\mathtt{AN}(F) - 1$ units). This is no longer the case for bilinear functions; that is, there exists bilinear functions F such that $\mathtt{AN}_2(F) \gg \mathtt{AN}(F)$.

Theorem 2.3 (separating $\mathtt{AN}_2$ from $\mathtt{AN}$): *There exist bilinear functions $F : (\mathrm{GF}(2)^n)^2 \to \mathrm{GF}(2)$ such that $\mathtt{AN}(F) = O(\sqrt{n})$ but $\mathtt{AN}_2(F) = \Omega(n^{2/3})$. 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, it holds that $\mathtt{AN}_{t+1}(F) = O(\mathtt{AN}(F))$.

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 $L_i(y) = \sum_{k=(i-1)s+1}^{si} y_k$. That is, f is obtained from g by replacing each variable z_i (of g) by the linear function $L_i(y)$; in the sequel, we shall refer to this f as being derived from g.

Clearly, $\mathtt{AN}(f) \leq 2s + 1$ 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 $\mathtt{AN}_2(f) = \Omega(n^{2/3})$.

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 AN-complexity at most $m = 0.9n^{2/3}$ that computes f. Given such a circuit, using a suitable (random) restriction, we obtain a circuit that computes the underlying

function g such that the resulting circuit belongs to a set containing at most $2^{0.9sn}$ circuits. 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 AN-complexity at most $m = 0.9n^{2/3}$; that is, for almost all such functions f, it holds that $\mathtt{AN}_2(f) > 0.9n^{2/3}$. The actual argument follows.

Consider an arbitrary *depth-two* multilinear circuit of AN-complexity m that computes a generic f (derived as above from a generic g). (We shall assume, w.l.o.g., that the top gate of this circuit is not fed directly by any variable, which can be enforced by replacing such variables with singleton linear functions.)[14] 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, and each of these functions depends on at most m variables. Furthermore, $m' + m'' + m''' < m$. (That is, Eq. (5) corresponds to a generic description of a depth-two multilinear circuit of AN-complexity m that computes a bilinear function. The top gate computes the sum of a bilinear function of $m' + m''$ intermediate linear gates and a sum of m''' intermediate bilinear gates, whereas all intermediate gates are fed by variables only.)

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).

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 $(2m^2\log_2 n) + (m^3/s)$ 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 result 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 $\sum_i B_i(x,y')$. Note that, for $m \geq n^{\Omega(1)}$, the quantity in Eq. (6) is upper-bounded by $2^{m^2+\widetilde{O}(m^2)+(m^3/s+\widetilde{O}(m^2))}$, and for $m > \widetilde{O}(n^{1/2})$ the dominant term in the exponent is m^3/s. In particular, 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 (i.e., 2^{sn}). Hence, for $m = 0.9n^{2/3}$, not every function f can be computed as in Eq. (5), and the theorem follows. ■

[14] Actually, this may increase m by one unit. The reason is that if the top gate if fed by i variables, then the number of intermediate gates in the circuit is at most $m-i$. So introducing intermediate singleton gates yields a depth-two circuit with at most $(m-i)+i$ intermediate gates.

Digest. The proof of the lower bound of Theorem 2.3 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 relatively small description (i.e., description length at most $0.9n^{1.5}$). Using the random restriction, we show that if f has depth-two AN-complexity at most $0.9n^{2/3}$, then the underlying g is (always) 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 depth-two AN-complexity greater than $0.9n^{2/3}$. (A more appealing abstraction, which requires a slightly more refined proof, is obtained by letting G contains all functions g that have depth-two multilinear circuits of AN-complexity at most $0.9n^{2/3}$ such that each gate is fed by at most $n^{1/6}$ variables from the short block.)[15]

Remark 2.4 (on the depth of multilinear circuits achieving $\mathtt{AN}$): *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 $\mathtt{AN}(F) = s$ for any t-linear function F, then there is a depth $t+1$ circuit with arity and size $O(s)$ computing F as well; that is, for any t-linear F, it holds that $\mathtt{AN}_{t+1}(F) = O(\mathtt{AN}(F))$. This observation is proved in Proposition 4.5.*

Remark 2.5 (waiving the multilinear restriction): *We note that arbitrary arithmetic circuits* (with general gates) *that compute t-linear functions can be simulated by multilinear circuits of the same depth, while increasing their AN-complexity measure by a factor of at most 2^t. This can be done by replacing each* (intermediate) *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 multilinear are not computed, and this is OK because their influence must cancel out at the top gate.*[16] *Indeed, the top gate performs the $2^t - 1$ computations that corresponds to the different I-linear sums, and sums-up the $2^t - 1$ results.*

2.2 Relation to canonical circuits

As outlined in Section 1.5, the direct and nested AN-complexity of multilinear functions (i.e., $\mathtt{AN}_2$ and $\mathtt{AN}$) 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 $\exp(\mathtt{AN}_2)$ and ND-canonical circuits of size $\exp(\mathtt{AN})$).

Construction 2.6 (D-canonical circuits of size $\exp(\mathtt{AN}_2)$): *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.)[17] *Then,*

[15] The point is that this alternative class G does not refer to the "description length" but rather to the complexity measures defined in this section. In this case, we may show that a random restriction of the type used in the original proof leaves m/s live variables in each G_i, in expectation, just as it holds for the B_i's. Using $m = 0.9n^{2/3}$, it holds that, with high probability, none of the gates exceeds this expectation by a factor of $1/0.9$. Next, we upper-bound the size of G, very much as done in the foregoing proof, where here the crucial fact is that each B_i has only $m \cdot n^{1/6}$ live terms, whereas $m^2 \cdot n^{1/6} = 0.81 \cdot n^{3/2}$.

[16] Here, we assume (as is standard in the area) that the cancellations must hold over any extension field of GF(2); that is, the polynomial x^i equals the polynomial $(2k+1) \cdot x^j$ if and only if $i = j$.

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

the following depth-three Boolean circuit 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. *Compose C_H with the various C_i's and C'_i's such that a positive occurrence of the* ith *variable of C_H is replaced by C_i and a negative occurrence is replaced by C'_i.*

   Collapsing the two adjacent levels of OR*-gates* (resp., AND-gates)*, yields 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. In particular, we have

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

It turns out that the upper bound provided in Proposition 2.7 is not tight; that is, D-canonical circuits do not provide the smallest depth-three Boolean circuits for all multilinear functions. In particular, there exists multilinear functions that have depth-three Boolean circuits of size $\exp(\mathtt{AN}_2(F)^{3/4})$. This follows by combining Theorem 2.3 and Proposition 2.9, where Theorem 2.3 asserts that for some bilinear functions F it holds that $\mathtt{AN}(F) = O(\sqrt{n}) = O(n^{2/3})^{3/4} = O(\mathtt{AN}_2(F))^{3/4}$, and Proposition 2.9 asserts that every multilinear function F has depth-three Boolean circuits of size $\exp(\mathtt{AN}(F))$. The latter is proved by using ND-canonical circuits, which leads us to their general construction.

Construction 2.8 (ND-canonical circuits of size $\exp(\mathtt{AN})$): *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 $x_k^{(\ell)}$).

   Indeed, each $C_{\overline{v}}$ is a CNF of size $\widetilde{O}(2^m)$.

3. We obtain a depth-three Boolean circuit C by letting

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

Hence, C has size $2^{m-1}\cdot\widetilde{O}(2^m)$.

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.*

Note that $C(x^{(1)},...,x^{(t)}) = 1$ if and only if there exists $\overline{v} = (v_1, v_2, ..., v_m) \in \mathrm{GF}(2)^m$ 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. In particular, we have

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

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 [11, Apdx C.1], 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 ($\mathtt{AN}$ 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 $\exp(\Omega(\exp(-t)\cdot\mathtt{AN}(F)))/\mathrm{poly}(n)$.*

When combined with adequate lower bounds on $\mathtt{AN}$ (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 $t = \sqrt{\log n}$. Furthermore, in some special cases (see [11, Apdx C.3]), multilinear functions that depends on m variables requires depth-two Boolean circuits of size at least $\exp(\Omega(m))$. This suggests making a bolder conjecture, 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 $\exp(\Omega(\mathtt{AN}(F)))/\mathrm{poly}(n)$. The same holds for almost all t-linear functions.*

When combined with adequate lower bounds on $\mathtt{AN}$ (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

In Section 3.1 we present a generic upper bound on the direct AN-complexity of any t-linear function; that is, we show that $\mathtt{AN}_2(F) = O((tn)^{t/(t+1)})$, for every t-linear function F. This bound, which is obtained by a generic construction, is the best possible for almost all multilinear functions (see Theorem 4.1). Obviously, one can do better in some cases, even when this may not be obvious at first glance. In Section 3.2, we focus on two such cases (i.e., $F^{t,n}_{\mathtt{leq}}$ and $F^{2,n}_{\mathrm{mod}\,p}$).

3.1 A generic upper bound

The following upper bound on the AN-complexity of multilinear circuits that compute a generic t-linear function is derived by using a depth-two circuit with a top gate that computes addition (i.e., a linear function). This implies that the intermediate gates in this curcuit, which are fed by variables only, must all be t-linear gates. While the overall structure of the circuit is oblivious of the t-linear function that it computes, the latter function determines the choice of the t-linear gates.

Theorem 3.1 (an upper bound on $\mathtt{AN}_2(\cdot)$ 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, $\mathtt{AN}_2(F) = O((tn)^{t/(t+1)})$.*

Proof: We partition $[n]^t$ into m equal-sized subcubes such that the number of subcubes (i.e., m) equals the number of variables that correspond to each subcube (i.e., $t \cdot \sqrt[t]{n^t/m}$); that is, the side-length of each subcubes is $\ell \stackrel{\mathrm{def}}{=} n/m^{1/t}$ and m is selected such that $m = t \cdot \ell$. 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 $((n/\ell)^t = m)$ cubes, and the sum of the lengths of each cube is $t\ell$. Let T be the tensor corresponding to F. Then,

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

Each $F_{k_1,...,k_t}$ is computed by a single $[t]$-linear gate of arity $t \cdot \ell$, and it follows that $\mathtt{AN}_2(F) \leq \max(t\ell, m+1)$, since $(n/\ell)^t = m$. Using $m = t\ell$ and recalling that $\ell = n/m^{1/t}$, we get $\mathtt{AN}_2(F) \leq m+1$ and $m/t = n/m^{1/t}$, which yields $\mathtt{AN}_2(F) = O((tn)^{t/(t+1)})$, since $m = (tn)^{\frac{1}{1+(1/t)}}$. ■

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. Examples include (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) $F^{t,n}_{\mathtt{diag}}(x^{(1)}, ..., x^{(t)}) =$

$\sum_{i\in[n]} x_i^{(1)} \cdots x_i^{(t)}$. Specifically, we can easily get $\mathtt{AN}_2(F^{t,n}_{\mathtt{all}}) \leq t\sqrt{n}+1$ and $\mathtt{AN}_2(F^{t,n}_{\mathtt{diag}}) \leq t\sqrt{n}$. 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 AN-complexity $t\sqrt{n}+1$ that use only ($\sqrt{n}$-way) addition and (t-way) multiplication gates.[18] While such simple 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 $\mathtt{AN}_2(F^{2,n}_{\mathtt{leq}}) = O(\sqrt{n})$; that is, as in the proof of Theorem 3.1, the following conmstruction also uses general multilinear gates.

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

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 $\bigcup_{i<j} S_{i,j} \subset T^{2,n}_{\mathtt{leq}} \subset \bigcup_{i\leq j} S_{i,j}$. 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 off 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. Thus, the contribution of each diagonal square will be computed by a designated bilinear gate, whereas the contribution of the off-diagonal squares will be computed by the top gate (which is fed by $2s$ linear gates, each computing the sum of s variables, and computes a suitable bilinear function of these $2s$ sums). Details follow.

- 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 $x^{(2)}_{(i-1)s+1}, ..., x^{(2)}_{is}$.

  Indeed, $Q_i(x^{(1)}, x^{(2)})$ computes the contribution of the i^{th} diagonal square (i.e., $S_{i,i}$). In contrast, the following linear functions will be used to compute the contribution of the off-diagonal squares.

- 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 $L'_i(x^{(2)}) = \sum_{j\in[s]} x^{(2)}_{(i-1)s+j}$.

Observing that

$$F^{2,n}_{\mathtt{leq}}(x^{(1)}, x^{(2)}) = \sum_{i\in[s]} Q_i(x^{(1)}, x^{(2)}) + \sum_{1\leq i<j\leq s} L_i(x^{(1)}) \cdot L'_j(x^{(2)}), \tag{7}$$

the claim follows. Specifically, we use intermediate gates that compute the Q_i's, L_i's and L'_j's (and let the top gate compute their combination (per Eq. (7))). ■

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).

[18] Depth-two circuits can be derived by combining the t-way multiplication gate with the $\sqrt{n}$-way addition gates feeding it (resp., each $\sqrt{n}$-way addition gate with the t-way multiplication gate feeding it).

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

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,

$$F^{2,n}_{\bmod p}(x^{(1)}, x^{(2)}) = \sum_{r \in \mathbb{Z}_p} L_r(x^{(1)}) \cdot L'_{p-r}(x^{(2)}).$$

Each of the foregoing $p \leq s$ linear functions depend on n/p variables, which is fine if $p = \Omega(s)$. Otherwise (i.e., for $p = o(s)$), 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). Then, the top gate just computes the suitable (biliear) combination of these $O(s)$ linear functions.

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

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

for every $i \in [s]$. Clearly, $F^{2,n}_{\bmod p}(x^{(1)}, x^{(2)}) = \sum_{i \in [s]} Q_i(x^{(1)}, x^{(2)})$, 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 $\mathtt{AN}_2(F^{t,n}_{\mathtt{leq}})$): *For every t, it holds that $\mathtt{AN}_2(F^{t,n}_{\mathtt{leq}}) = O(\exp(t) \cdot \sqrt{n})$.*

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 an 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

$$F^{t,[i,n]}_{\mathtt{leq}}(x^{(1)}, ..., x^{(t)}) \stackrel{\text{def}}{=} \sum_{(i_1,...,i_t) \in T^{t,n}_{\mathtt{leq}} : i_1 \geq i} x^{(1)}_{i_1} \cdots x^{(t)}_{i_t}. \tag{8}$$

Indeed, $F^{t,n}_{\mathtt{leq}} = F^{t,[1,n]}_{\mathtt{leq}}$. The *inductive claim*, indexed by $t \in \mathbb{N}$, asserts 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 $(i_1,...,i_t) \in T^{t,[(k-1)s+1,n]}_{\mathtt{leq}}$ uniquely determines $j \in [t]$ and $k_1 \in [k,n]$ 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

$$F^{t,[(k-1)s+1,n]}_{\mathtt{leq}}(x^{(1)},...,x^{(t)}) = \sum_{j\in[t-1]} \sum_{k_1 \geq k} P^{(j)}_{k_1}(x^{(1)},...,x^{(j)}) \cdot F^{t-j,[k_1 s+1,n]}_{\mathtt{leq}}(x^{(j+1)},...,x^{(t)})$$

$$\text{where } P^{(j)}_{k_1}(x^{(1)},...,x^{(j)}) \stackrel{\text{def}}{=} \sum_{(i_1,...,i_j)\in(T^{j,n}_{\mathtt{leq}}\cap(I_{k_1})^j)} x^{(1)}_{i_1} \cdots x^{(j)}_{i_j}.$$

It follows that all $F^{t,[(k-1)s+1,n]}_{\mathtt{leq}}$'s are simultaneously expressed in terms of $(t-1)\cdot s$ new functions (i.e., the $P^{(j)}_{k_1}$'s), each depending on at most $t \cdot s$ inputs, and $(t-1)\cdot s$ functions (i.e., the $F^{t-j,[k_1 s+1,n]}_{\mathtt{leq}}$'s) that by the induction hypothesis can be expressed using $\sum_{j\in[t-1]}(t-j)2^{t-j}\cdot s$ multilinear functions (although with different variable names for different j's).[19] So, in total, we expressed all $F^{t,[(k-1)s+1,n]}_{\mathtt{leq}}$'s using less than $ts+\sum_{j\in[t-1]}(t-j)2^{t-j}\cdot s$ 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 $t2^t s$, the induction claim follows. This establishes that $\mathtt{AN}(F^{t,n}_{\mathtt{leq}}) \leq t2^t \cdot \sqrt{n}$.

In order to prove $\mathtt{AN}_2(F^{t,n}_{\mathtt{leq}}) \leq t2^t \cdot \sqrt{n}$, we take a closer look at the foregoing expressions. Specifically, note that all $F^{t,[(k-1)s+1,n]}_{\mathtt{leq}}$ are expressed in terms of $t2^t s$ functions such that each function is either a polynomial in the input variables or another function of the form $F^{t-j,[k_1 s+1,n]}_{\mathtt{leq}}$. In terms of multilinear circuits, this means that each gate is fed either only by variables or only by other gates (rather than being fed by a mix of both types). It follows that the top gate is a function of all gates that are fed directly by variables only, and so we can obtain a depth-two multilinear circuit with the same (or even slightly smaller) number of gates and the same (up to a factor of 2) gate arity. ■

Remark 3.5 (circuits having no mixed gates yield depth-two circuits): *The last part of the proof of Proposition 3.4 relied on the fact that if no intermediate gate of the circuit is fed by both variables and other gates, then letting all intermediate gates feed directly to the top gate yields a depth-two circuit of AN-complexity that is at most twice the AN-complexity of the original circuit. As can be seen in the proof of Theorem 2.3, the benefit of feeding a gate by both intermediate gates and varaibles is that it may multiply these two types of inputs. Such a* mixed gate*, which may apply an arbitrary multilinear function to its inputs, can be split into two non-mixed gates* only if *it sums a function of the variables and a function of the other gates. It is also* not *feasible to feed the top gate with all variables that are fed to mixed gates, becuase this may square the AN-complexity.*

4 Lower Bounds

We believe that the generic upper bound established by Theorem 3.1 (i.e., every t-linear function F satisfies $\mathtt{AN}(F) \leq \mathtt{AN}_2(F) = O((tn)^{t/(t+1)})$ is tight for many explicit functions. However, we were

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

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 AN-complexity at least $\sqrt{n}$. Of course, $\mathtt{AN}(P_n) = \Omega(\sqrt{n})$ follows by combining Proposition 2.9 with either [12] or [14], but the foregoing proof is much simpler (*to say the least*) and yields a better constant in the Ω-notation.

4.1 On the AN-complexity of almost all multilinear functions

Theorem 4.1 (a lower bound on the AN-complexity of 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 $\mathtt{AN}(F) = \Omega(tn^{t/(t+1)})$. Furthermore, such a t-linear function can be found in $\exp(n^t)$ time.*

Combined with Theorem 3.1, it follows that almost all t-linear functions satisfy $\mathtt{AN}(F) = \Theta(tn^{t/(t+1)})$. Here (and elsewhere), we use the fact that $t^{t/(t+1)} = \Theta(t)$.

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 $\mathtt{AN}(F) \leq m$. 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, this is an over-specification in many ways.)[20] Thus, the number of such choices is upper-bounded by

$$\binom{tn}{m} \cdot t^m \cdot 2^{((2m/t)+1)^t} \tag{9}$$

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.[21] Note that Eq. (9) 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 $m \leq c \cdot (tn)^{t/(t+1)})$, for some $c > 0$. The main claim follows.

[20] For starters, we allowed each gate to be feed by m original variables and m auxiliary functions, whereas the arity bound is m. Furthermore, we allowed each gate to be fed by all other gates, whereas the circuit should be acyclic. Moreover, the choice of the t-partition can be the same for all gates, let alone that the various t-partitions must be consistent among gates and adheres to the multilinearity condition of Definition 2.1.

[21] Denoting by m_j the number of variables and/or 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 $\sum_{j\in[t]} m_j \leq 2m$.

The furthermore claim follows by observing 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 similar lower bounds for some explicit multilinear functions. In the original version of this work [11], we suggrested the following "modest start":

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

This challenge was met by Goldreich and Tal [9], who showed that $\mathtt{AN}(F^{3,n}_{\mathtt{tet}}) = \Omega(n^{0.6})$ and that $\mathtt{AN}(F) = \widetilde{\Omega}(n^{2/3})$ holds for some explicit 4-linear F. Referring to Problem 4.2, their work leaves open the case of $t = 2$ (for any $c > 1/2$) as well as obtaining $c > 2/3$ (for any $t > 2$). The more ambitious goal set in [11] remains far from reach, since the techniques of [9] (which are based on the "rigidity connection" made in Section 4.2) cannot yield $c > 2/3$.

Problem 4.3 (the ultimate goal regarding lower bounds regarding AN): *For every* $t \geq 2$, *prove that* $\mathtt{AN}(F) = \Omega((tn)^{t/(t+1)})$ *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 $\mathtt{AN}(F) = \Omega((tn)^{\epsilon t/(\epsilon t+1)})$, for some fixed constant $\epsilon > 0$, will also allow to derive exponential lower bounds when setting $t = O(\log n)$.

4.2 The AN-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 AN-complexity 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 [31] and surveyed in [19]). 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 $\omega(n^{1/2})$).[22] Anyhow, here is its basic connection to our model.

Theorem 4.4 (reducing AN-complexity 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* $\mathtt{AN}(F) > m$.

[22] Added in Revision: Interestingly, a subsequent work of Dvir and Liu [3, 4] shows that no Toeplitz matrix is rigid in the Valiant range of parameters. Specifically, they show that, for any constant $c > 1$, no Toeplitz matrix has rigidity n^c with respect to rank $n/\log n$ (see [4], which builds upon [3]). In contrast, the subsequent work of Goldreich and Tal [9] shows that almost all Toeplitz matrix have rigidity $\widetilde{\Omega}(n^3)$ with respect to rank $r \in [\sqrt{n}, n/32]$.

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 $\mathtt{AN}(F) > m$.*

Proof: As a warm-up, we first prove that $\mathtt{AN}_2(F) > m$; that is, we prove a lower bound referring to depth-two multilinear circuits rather than to general multilinear circuits. Suppose towards the contradiction that $\mathtt{AN}_2(F) \leq m$, and consider the multilinear circuit that guarantees this bound. Without loss of generality,[23] it holds that $F(x,y) = H(F_1(x,y),...,F_{m-1}(x,y))$, 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 or y). Furthermore, each F_i depends on at most m variables. Since $H(F_1(x,y),...,F_{m-1}(x,y))$ is a syntactically bilinear polynomial (in x and y), it follows that it has the form

$$\sum_{i\in[m']} Q_i(x,y) + \sum_{(j_1,j_2)\in P} L_{j_1}(x)L_{j_2}(y), \tag{10}$$

where $P \subset [m'+1,m''] \times [m''+1,m-1]$ (for some $m'' \in [m'+1,m-2]$) and each Q_i and L_j depends on at most m variables. (Indeed, the same form was used in the proof of Theorem 2.3 (see Eq. (5)).) Furthermore, each of the L_j's is one of the auxiliary functions F_i's, which means that the second sum (in Eq. (10)) depends on at most $m-1$ different (linear) functions.

The key observation is that bilinear functions correspond to matrices; that is, the bilinear function $B : \mathrm{GF}(2)^{n+n} \to \mathrm{GF}(2)$ corresponds to the n-by-n matrix M such that the $(k,\ell)^{\text{th}}$ entry of M equals 1 if and only if the monomial $x_k y_\ell$ is included in $B(x,y)$ (i.e., iff $B(0^{k-1}10^{n-k}, 0^{\ell-1}10^{n-\ell}) = 1$).[24] Now, observe that the matrix that corresponds to the first sum in Eq. (10) 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. (10) has rank at most $m-1$ (since the sum $\sum_{(j_1,j_2)\in P} L_{j_1}L_{j_2}$, viewed as $\sum_{j_1\in[m-1]} L_{j_1} \cdot \sum_{j_2:(j_1,j_2)\in P} L_{j_2}$, corresponds to the sum of $m-1$ rank-1 matrices).[25] But this contradicts the hypothesis that T has rigidity m^3 for rank m, and so $\mathtt{AN}_2(F) > m$ follows.

Turning to the actual proof (of $\mathtt{AN}(F) > m$), 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 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; cf., Remark 3.5). 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 (since products of linear F_i's can be moved to the top gate). Thus, the expression for $F(x,y)$ is as in Eq. (10), 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).[26] This suffices for completing the argument. Details follow.

Suppose towards the contradiction that $\mathtt{AN}(F) \leq m$, and consider a multilinear circuit that supports this bound. Each of the $m' \leq m$ gates in this circuit computes a bilinear (or linear)

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

[24] In terms of Eq. (1), letting T denote the set of one-entries of M, it holds that $B(x,y) = \sum_{(k,\ell)\in T} x_k y_\ell$.

[25] That is, letting $L'_j(y) = \sum_{j_2:(j,j_2)\in P} L_{j_2}(y)$, we consider the sum $\sum_{j_1\in[m-1]} L_{j_1}(x) \cdot L'_{j_1}(y)$, and note that each term corresponds to a rank-1 matrix (i.e., the $(k,\ell)^{\text{th}}$ entry of the j_1^{th} matrix equals $L_{j_1}(0^{k-1}10^{n-k}) \cdot L'_{j_1}(0^{\ell-1}10^{n-\ell})$).

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

function of its feeding-inputs, which are a possible mix of (up to m) original variables and (up to $m-1$) outputs of other gates. This bilinear (or linear) function of the feeding-inputs can be expressed as a sum of monomials of the following three types, where F_i denotes the auxiliary function computed by the i^{th} internal gate (and $F_0 = F$ is the function computed by the top gate).

1. Mixed monomials that consist of the product of a linear auxiliary function (i.e., an F_j) and an original variable. Such monomials cannot exist in the computation of linear functions.

2. Monomials that consist only of auxiliary functions F_j's: Such a monomial may be either a single bilinear (or linear) function or a product of two linear functions.[27]

   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 fed by this gate (without effecting the number of variables that feed directly to these gates). Note that the arity of gates in the resulting circuit is at most $m+m$, where one term is due to the number of variables that feed directly into the gate and the other term is due to the total number of gates in the circuit.

   For example, if the monomial $F_k(x)F_\ell(y)$ appears in the expression computed by the j^{th} internal gate (which computes $F_j(x,y)$) that feeds the i^{th} gate (which computes $F_i(x,y)$), where possiblly $i=0$ (i.e., the j^{th} gate feeds the top gate), then we can remove the monomial F_kF_ℓ from F_j and add it to F_i, which may require adding F_k and F_ℓ to the list of gates (or rather functions) that feed F_i. Ditto if $F_k(x,y)$ is a monomial of F_j. The process may be repeated till no internal gate contains a monomial that consists only of auxiliary functions.

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

Hence, we obtain the general form for the computations of the top gate (which computes F) and the intermediate gates (which compute the auxiliary functions F_i's):

$$\begin{aligned}
F(x,y) &= \sum_{(k,\ell)\in P_{0,1}} F_k(x)y_\ell + \sum_{(k,\ell)\in P_{0,2}} x_kF_\ell(y) \\
&\quad + \sum_{i\in S} F_i(x,y) + \sum_{(i,j)\in P_3} F_i(x)F_j(y) + \sum_{(i,j)\in P_{0,4}} x_iy_j \\
F_i(x,y) &= \sum_{(k,\ell)\in P_{i,1}} F_k(x)y_\ell + \sum_{(k,\ell)\in P_{i,2}} x_kF_\ell(y) + \sum_{(k,j)\in P_{i,4}} x_ky_j \\
F_i(z) &= \sum_{k\in S_i} z_k
\end{aligned}$$

where the P's are subsets of $[m]^2$ (resp., the S's are subsets of $[m]$), and the F_i's (of arity at most $2m$) replace the original F_i's (per the 'w.l.o.g."-cluase of Item 2). Indeed, as asserted in Item 2, only the top gate contains monomials that are either auxiliary bilinear functions (corresponding to S) or products of auxiliary linear functions (corresponding to P_3).

Summing together all mixed monomials, *regardless of the gate to which they belong*, we obtain at most $m-1$ quadratic forms, where each quadratic form is the product of one of the auxiliary

[27] Since, as argued next, such monomials exist only in the top gate, it follows that (w.l.o.g.) they cannot be a single linear function, because the top gate must compute a homogeneous polynomial of degree 2.

(linear) functions F_i and a linear combination (of an arbitrary number) of the original variables. Let us denote this sum by σ_1; that is,

$$\begin{aligned}\sigma_1 &= \sum_{i\in\{0,1,...,m-1\}} \left(\sum_{(k,\ell)\in P_{i,1}} F_k(x)y_\ell + \sum_{(k,\ell)\in P_{i,2}} x_k F_\ell(y) \right) \\ &= \sum_k F_k(x) \cdot \sum_i \sum_{\ell:(k,\ell)\in P_{i,1}} y_\ell \;+\; \sum_\ell F_\ell(y) \cdot \sum_i \sum_{k:(k,\ell)\in P_{i,2}} x_k\end{aligned}$$

Adding to this sum (i.e., σ_1) the sum, denoted σ_2, of all monomials (computed by the top gate) that are a product of two linear F_i's (i.e., $\sigma_2 = \sum_{(i,j)\in P_3} F_i(x)F_j(y)$), 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. (This uses 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.) These sums leave out the monomials that are a product of two original varaibles (i.e., the sum $\sum_{i\in\{0,1,...,m-1\}} \sum_{(k,j)\in P_{i,4}} x_k y_j$). We stress that sum $\sum_{i\in S} F_i(x,y)$ is not included here, since the monomials computed by these F_i's are already accounted by one of the foregoing three types (i.e., they either appear in the sum $\sigma_1+\sigma_2$ or were left out as products of two variables).

Let T' denote matrix that corresponds to the $F' = \sigma_1 + \sigma_2$. Note that T' has rank at most $m-1$ (since it is the sum of at most $m-1$ rank-1 matrices, which correspond to the products of the different linear F_i's with arbitrary linear functions). Lastly, note that $F - F'$ equals $\sum_{i\in\{0,1,...,m-1\}} \sum_{(k,j)\in P_{i,4}} x_k y_j$, which means that T' differs from T on at most m^3 entries. (Actually, the disagreement is smaller, since $|P_{i,4}| \leq \max_{m'\in[m-1]}\{m'\cdot(m-m')\leq (m/2)^2$.) This implies that $T = T' + (T - T')$ does not have rigidity m^3 for rank m, and the claim follows. ■

A short detour. 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 the AN-complexity): *Let F be ay t-linear function. Then, there exists a depth $t+1$ circuit with arity and size $\mathtt{AN}(F)$ that computes F. That is, for any t-linear F, it holds that $\mathtt{AN}_{t+1}(F) = O(\mathtt{AN}(F))$.*

Proof: Generalizing an observation made in the proof of Theorem 4.4, note that monomials in the expression for F_j that contain *only* auxiliary functions can be moved to the expressions of all functions that depend on F_j (while at most doubling the AN-complexity of the circuit). Thus, without loss of generality, each auxiliary function F_j (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_j). Hence, using induction on $i \geq 0$, it holds that gates that are at distance i from the top gate are fed by auxiliary functions of degree at most $t-i$. It follows that gates at distance t from the top are only fed by variables. Thus, the depth of multilinear circuits computing a t-linear function needs not exceed $t+1$. ■

Implications of the "rigidity connection" on $\mathtt{AN}(F^{3,n}_{\mathtt{tet}})$. In the original version of this work [11], we suggested to try to obtain an improved lower bound on the AN-complexity of the trilinear function $F^{3,n}_{\mathtt{tet}}$ (see Eq. (3)) via a reduction to proving a rigidity lower bound for a *random* (or actually

any) Toeplitz matrix. 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 $\mathtt{AN}(F^{3,n}_{\mathtt{tet}})$ to proving lower bounds on the AN-complexity 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 $\mathtt{AN}(F) \geq m$, then $\mathtt{AN}(F^{3,n}_{\mathtt{tet}}) = \Omega(m)$.*

Indeed, a striking feature of of this reduction is that a lower bound on an explicit function follows from a lower bound on *any* function in a natural class that contained exponentially many different functions.

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 $T_3 = \{(i_1, i_2, i_3) \in [[n']]^3 : \sum_j i_j \leq n'\}$, where $[[n']] \stackrel{\mathrm{def}}{=} \{0, 1, ..., n'\}$ (and $n' = \lfloor n/2 \rfloor$). Indeed, T_3 is a lightly padded version of one eighth of $T^{3,n}_{\mathtt{tet}}$. Observe that multilinear circuits for $F^{3,n}_{\mathtt{tet}}$ yield circuits of similar AN-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).[28]

The main observation is that *if $F_3(x, y, z) = \sum_{(i,j,k)\in T_3} x_i y_j z_k$ satisfies $\mathtt{AN}(F_3) \leq m$, 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 $I \subseteq [[n']]$, 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 (i.e., given a triangular Toeplitz matrix $(t_{j,k})_{j,k\in[[n']]}$, let $I = \{i \in [[n']] : t_{0,n'-i} = 1\}$). (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 AN-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 $\mathtt{AN}(F^{3,n}_{\mathtt{tet}})$ reduces to establishing this bound for some Toeplitz matrix. This gives rise to the following open problems posed in [11] and resolved in [9].

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

(This was proved for $c = 0.6$ in [9, Cor. 1.4].) 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.*

[28] 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 $\{\pm 1\}^3$.

(This was proved for $c = 0.6 - o(1)$ in [9, Thm. 1.2], whereas the improved bound for $c = 0.6$ (in [9, Cor. 1.4]) was established via "structured rigidity" as defined next.)

4.3 On structured rigidity

The proof of Theorem 4.4 shows that if a bilinear function F has AN-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.

Theorem 4.10 (reducing AN-complexity 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 $\mathtt{AN}(F) \geq m$.*

(As stated above, Theorem 4.10 follows by the very proof of Theorem 4.4.) 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 $\mathtt{AN}(F) \geq m$.* 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.*

Indeed, the lower bound of $\Omega(n^{0.6})$ on the AN-complexity of (the bilinear functions that correspond to) most n-by-n Toeplitz matrices has been proved in [9] by establishing a similar lower bound on the *structured rigidity* of these matrices, improving over a lower bound of $\widetilde{\Omega}(n^{0.6})$ established in [9] via a similar lower bound on the standrad notion of rigidity (see [9, Thm. 1.2] versus [9, Thm. 1.3]). This provides some weak empirical evidence for the speculation, made in the original version of this work [11], by which Problem 4.11 may be easier than Problem 4.8. This speculation was supported in [11] by the following separation result.

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.*

Note that M does not have rigidity $3nm \ll m^3$ 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 $\frac{m^3}{3nm} = \Omega(m^2/n)$.

Proof: For each sequence $R, S_1, ..., S_m$ such that R has rank m and each $S_i \subseteq [n] \times [n]$ is an m-by-m square (generalized) submatrix (i.e., has the form $I_i \times J_i$ such that $|I_i|, |J_i| \leq m$), we shall show that

$$\Pr_{M \in \mathrm{GF}(2)^{n \times n} : |M| = 3mn} \left[M - R \subseteq \bigcup_{i \in [m]} S_i \right] \leq 2^{-3nm}, \tag{11}$$

where M is a uniformly selected n-by-n matrix with exactly $3mn$ ones (and $M - R \subseteq S$ means that all non-zero entries of the matrix $M - R$ reside in the set $S \subseteq [n] \times [n]$). The theorem follows since the number of such sequences (i.e., a rank m matrix R and small submatrices $S_1, ..., S_m$) is smaller than $(2^{2n})^m \cdot \binom{n}{m}^{2m} \ll 2^{2nm + 2m^2 \log n}$, where we specify a rank-m matrix by a sequence of m rank-1 matrices (equiv., pairs of subsets of $[n]$). Using $m^2 \log n < nm/4$ (equiv., $m = o(n/\log n)$), the foregoing quantity is upper-bounded by $2^{2.5nm}$. We shall also use $m \leq n^{2/3}/2$, which implies $m^3 \leq n^2/8$ and $3nm = o(n^2)$. In order to prove Eq. (11), we consider two cases

Case 1:* R *has at least* $n^2/3$ *one-entries. Since $3nm = o(n^2)$, it follows that $M - R$ has at least $n^2/4$ non-zero entries, but these cannot be covered by the $\bigcup_i S_i$, since the latter has at most $m^3 \leq n^2/8$ elements. Hence, $M - R \subseteq \bigcup_{i \in [m]} S_i$ never holds in this case, which means that the l.h.s. of Eq. (11) is zero.

Case 2:* R *has at most* $n^2/3$ *one-entries. In this case the union of the one-entries of R and $\bigcup_i S_i$, denoted U, covers at most half of a generic n-by-n matrix. Now, selecting $3nm$ random entries in the matrix, the probability that all entries reside in U at most $(1/2)^{3nm}$. But if some one-entry of M does not reside in U, then this entry is non-zero in $M - R$ but does not reside in $\bigcup_i S_i$. In this case, $M - R \not\subseteq \bigcup_{i \in [m]} S_i$ holds. Hence, Eq. (11) holds.

To rec-cap: Having established Eq. (11), and recalling the upper bound on the number of $(R, S_1, ..., S_m)$-sequences, we conclude that with probability at least $1 - 2^{2.5nm} \cdot 2^{-3nm} = 1 - 2^{-nm/2}$, the matrix M has structural rigidity (m, m, m) 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 $T \neq R + S$. 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.* [5], 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*[29] rank $n/100$, the matrix S must have at least $n^{1.1}$ ones.

[29] 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 [5] refers to the rank of the matrix over the real.

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*.[30] 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 that do not use it. In fact, 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 AN-complexity $\Omega(n^{2/3})$, where the AN-complexity of circuits is as defined in Definition 2.2. In contrast, $F^{2,n}_{\mathtt{had}}$ can be computed by a circuit of AN-complexity $\widetilde{O}(\sqrt{n})$* with cancellation*; actually, $\mathtt{AN}_2(F^{2,n}_{\mathtt{had}}) = O(\sqrt{n \log n})$.*

Proof: We first prove the lower bound. Suppose that $F^{2,n}_{\mathtt{had}}$ can be computed by a circuit of AN-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

$$F^{2,n}_{\mathtt{had}}(x,y) = F_0(x,y) = \sum_{i=0}^{m'} Q_i(x,y) + \sum_{i=m'+1}^{m-1} L_i(x,y) F_i(x,y)\,, \tag{12}$$

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 or y).[31] 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.

The punchline is that, by the non-cancellation hypothesis, these rectangles (i.e., the $L_i F_i$'s) 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., [6, 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 $(m'+1) \cdot m^2 + (m-m') \cdot n < m^3 + mn$, whereas $T^{2,n}_{\mathtt{had}}$ has $n^2/2$ one-entries. The main claim (i.e., $m = \Omega(n^{2/3})$) follows.

[30] This means that one considers the syntactic polynomial computed by the circuit (over a generic field) and requires that it equals the target polynomial when the field remains unspecified.

[31] 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. (12).

The secondary claim (i.e., $\mathtt{AN}(F^{)}_{\mathtt{had}} = \widetilde{O}(\sqrt{n})$) follows by the fact that $T^{2,n}_{\mathtt{had}}$ has rank $\ell = \log_2 n$. The point is that any bilinear 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 can be computed as the product of two linear functions, and each linear function can be computed as the sum of $\sqrt{n/2r}$ functions that compute the sum of at most $\sqrt{2rn}$ variables. All in all, we use $1 + 2r \cdot \sqrt{n/2r}$ gates, which are each are each of arity $\sqrt{2rn}$. This yields a depth-two circuit of AN-complexity $\sqrt{2rn} + 1$, where the top gate is a quadratic expression in $\sqrt{2rn}$ linear functions. ■

Computing $F^{3,n}_{\mathtt{tet}}$ without cancellation. While we were unable to prove that $\mathtt{AN}(F^{3,n}_{\mathtt{tet}}) = \omega(\sqrt{n})$, 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 AN-complexity $\Omega(n^{2/3})$.*

(Again, recall that the AN-complexity of circuits is defined exactly 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

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

where $m' + m'' + m''' \leq m - 1$, the cubic functions 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) functions (which may depend on an arbitrary number of variables (from adequate variable-blocks)). and the other F_i's are either quadratic (for $i \in [m' + 1, m' + m'']$) or linear (for $i \in [m' + m'' + 1, m' + m'' + m''']$).[32] Combining the two last summations in Eq. (13), we obtain

$$F_0 = \sum_{i=0}^{m'} C_i + \sum_{i=m'+1}^{m-1} L'_i Q'_i \tag{14}$$

where the C_i's are as in Eq. (13), and the L'_i's (resp., Q'_i's) are arbitrary linear (resp., quadratic) functions (which may depend on an arbitrary number of variables (from adequate variable-blocks)). Note that C_i corresponds to a tensor with one-entries that are confined to a m-by-m-by-m box, and each $L'_i Q'_i$ 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 $T^{3,n}_{\mathtt{tet}}$.*

[32] 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. (13).

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 $i'_1 \in I'_j$, 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 restricted 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 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, because the model seems restricted enough as is. Note that for every t-linear function F, it holds that $\mathtt{AN}(F) \leq t \cdot \mathtt{RC}(F)$, 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.[33]

[33] 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.

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 s with respect to rank $r > 1$, then $\mathtt{RC}(F) \geq \min(r, \sqrt{s})$.*

As shown in Proposition 5.5, the tensor $T^{2,n}_{\mathtt{leq}}$ has rigidity $\Omega(n^2/r)$ with respect to rank r, so letting $r = n^{2/3}$, we obtain $\mathtt{RC}(F^{2,n}_{\mathtt{leq}}) = \Omega(n^{2/3})$, since $\sqrt{n^2/r} = n^{(2-(2/3))/2}$. Also, since a random n-by-n matrix has rigidity $\Omega(n^2)$ with respect to rank $\Omega(n)$, it follows that for almost all bilinear functions $F : \mathrm{GF}(2)^{n+n} \to \mathrm{GF}(2)$ 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 $n^{t/2}$ addition gates, each of arity $n^{t/2}$, that sum-up all the relevant monomials).

Proof: Assuming that T has rigidity s with respect to rank $r > 1$, and that $m \stackrel{\mathrm{def}}{=} \mathtt{RC}(F) < \sqrt{s}$, we shall 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. Also note that there exists exactly one multiplication gate on each path from the top gate to a variable, and that this gate is trivial if and only if it is the last gate on this path. 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 s with respect to rank r, since $|T + T'| \leq m^2 < s$ (by our hypothesis that $m < \sqrt{s}$).
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.*

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 $I_k = \{(k-1)n/r+1, ..., kn/r\}$ and $J_k = \{kn/r+1, ..., n\}$, for every $k \in [r]$. (This is the case since $\sum_{k\in[r]}(I_k \times J_k) \subseteq T^{2,n}_{\texttt{leq}} \subseteq \sum_{k\in[r]}(I_k \times J_{k-1})$, and $\sum_{k\in[r]} |I_k \times (J_{k-1} - J_k)| = n^2/r$.)

Proof: For a constant $c > 1$ to be determined later, we consider any $r < n/c$. We shall prove that any matrix $R = (R_{i,j})_{i,j\in[n]}$ of rank r is $\Omega(1/r)$-far from $T \stackrel{\text{def}}{=} T^{2,n}_{\texttt{leq}}$; that is, $|R + T| = \Omega(n^2/r)$.

Let R be an arbitrary matrix of rank at most r. We say that $i \in [n]$ is good if $|\{j \in [n] : R_{i,j} \neq T_{i,j}\}| < n/cr$. The claim of the proposition reduces to proving that at least half of $i \in [n]$ are not good, since in this case R 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 R by w_k. Then, for a random non-empty set $K \subseteq [c'r]$, the following two conditions hold:

1. With probability greater than $1 - 2^{-r}$, the vector $\sum_{k\in K} v_k$ has weight greater than $n/6$.

   This follows from the structure of T (i.e., $v_k = 0^{i_k-1}1^{n-i_k+1}$) and the distance between the i_k's. Specifically, for a random K, the weight of the vector $\left(\sum_{k\in K} v_k \bmod 2\right)$ is distributed as $\sum_{j\in[c'r]}(i_{j+1} - i_j) \cdot X_j$, where $i_{c'r+1} = n+1$ and $X_j = \sum_{k\in K} T_{i_k,i_j} \bmod 2$ indicates the parity of the elements selected in column i_j (which equals the parity in all columns in $[i_j, i_{j+1} - 1]$). Thus, $X_j = \left(\sum_{k\leq j} Y_k \bmod 2\right)$, where $Y_k = 1$ if $k \in K$ and $Y_k = 0$ otherwise, which implies that the X_j's are uniformly and indentially distributed in $\{0, 1\}$. For sufficiently large c', we have $\Pr\left[\sum_{j\in[c'r-1]} X_j > c'r/3\right] > 1 - 2^{-r}$, and the claim follows since $\sum_{j\in[c'r]}(i_{j+1} - i_j) \cdot X_j$ is greater than $(n/2c'r) \cdot \sum_{j\in[c'r]} X_j$ (and $(n/2c'r) \cdot (c'r/3) = n/6$).

2. With probability at least 2^{-r}, the vector $\left(\sum_{k\in K} w_k \bmod 2\right)$ has weight 0.

   This follows from the rank of R. Specifically, consider a maximal set of independent vectors among the $w_1, ..., w_{c'r}$, and denote the corresponding set of indices by I. Then, $\Pr_K\left[\sum_{k\in K} w_k = 0\right] = 2^{-|I|} \geq 2^{-r}$, which can be seen by first selecting a random $K' \subseteq ([c'r] \setminus I)$, and then (for any outcome K') selecting a random $K'' \subseteq ([c'r] \cap I)$.

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 $\sum_{k\in K} w_k$ has weight 0. But this is impossible because, by the hypothesis that all i_k's are good, the distance between these two vectors is at most $|K| \cdot n/(cr) \leq c'r \cdot n/(cr) < n/6$, 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 $s = \Omega(n^2/r)$. 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 $s = n^{1/3}$: For $i, j \in [n/s]$, let $S_{i,j} = [(i-1)s+1, is] \times [(j-1)s+1, js]$, 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 $n/s = n^{2/3}$ diagonal squares and the contribution of the squares that are above the diagonal; that is,

$$F^{2,n}_{\mathtt{leq}}(x,y) = \sum_{i \in [n^{2/3}]} \sum_{(k,\ell) \in S_{i,i} : k \leq \ell} x_k y_\ell + \sum_{i<j} \sum_{(k,\ell) \in S_{i,j}} x_k y_\ell .$$

The contribution of the square $S_{i,i}$ can be computed as the sum of its relevant $r \stackrel{\text{def}}{=} \binom{s}{2} + s < n^{2/3}$ entries, which means that the sum of the contribution of all $n^{2/3}$ 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 $n^{2/3}$. (We also use $n^{2/3} \cdot r < n^{4/3}/2$ trivial multiplication gates, but these are not counted.)

The contribution of the above-diagonal squares can be computed by writing $\cup_{i<j} S_{i,j}$ as $\sum_{i \in [n/s]} R_i$, where $R_i = [(i-1)s+1, is] \times [(i-1)s]$. The contribution of each of the $n/s = n^{2/3}$ rectangles (i.e., R_i's) can be computed by multiplying two linear expressions (see next). Hence, the total contribution of the off-diagonal squares is

$$\sum_{i<j} \sum_{(k,\ell) \in S_{i,j}} x_k y_\ell = \sum_{i \in [n/s]} \sum_{(k,\ell) \in R_i} x_k y_\ell = \sum_{i \in [n^{2/3}]} \left(\sum_{k \in [(i-1)s+1, is]} x_k \right) \cdot \left(\sum_{\ell \in [(i-1)s]} y_\ell \right) .$$

The point is that there are $n^{2/3}$ linear expressions each involving $s = n^{1/3}$ 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 $[(i-1)s] = [(i-2)s] \cup [(i-2)s+1, (i-1)s]$ (i.e., the i^{th} addition gate sums the result of the $i-1^{\text{st}}$ addition gate and s new variables). (We also use $n^{2/3}$ multiplication gates, each of arity 2.) The claim follows. ■

Added in revision: A lower bound on the restricted complexity of $F^{3,n}_{\mathtt{tet}}$. Combining [9, Thm. 1.2] with Theorem 5.4, we get $\mathtt{RC}(F^{3,n}_{\mathtt{tet}}) = \widetilde{\Omega}(n^{3/4})$. This follows because by [9, Thm. 1.2] almost all n-by-n Toeplitz matrices have rigidity $\widetilde{\Omega}(n^3/r^2)$ with respect to rank $r \in [\sqrt{n}, n/32]$, and (by Theorem 5.4) each corresponding bilinear function F satisfies $\mathtt{RC}(F) \geq \min(r, \widetilde{\Omega}(n^{3/2}/r) = \widetilde{\Omega}(n^{3/4}$ (using $r = n^{3/4}$). The bound for $F^{3,n}_{\mathtt{tet}}$ follows analogously to Proposition 4.6.

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 the restricted complexity of 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 gate-entries is at most m and each is selected among at most $(n+1)^t$ possibilities (since there are $(n+1)^t$ possible multilinear monomials). Thus, the number of such circuits is at most

$$2^m \cdot 2^{\binom{m}{2}} \cdot \binom{(n+1)^t}{m}^m \tag{15}$$

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. (15) 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, and to Avishay Tal for many suggestions for improving the presentation. Research was partially done while O.G. visited the IAS.

References

[1] M. Ajtai. Σ^1_1-formulae on finite structures. *Ann. Pure Appl. Logic*, Vol. 24 (1), pages 1–48, 1983.

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

[3] Z. Dvir and A. Liu. Fourier and Circulant Matrices Are Not Rigid. In *34th CCC*, pages 17:1–17:23, 2019. See also `arXiv:1902.07334 [math.CO]`, Feb. 2019.

[4] Z. Dvir and A. Liu. Fourier and Circulant Matrices Are Not Rigid. To appear in *TOC*, special issue of *34th CCC*. See also *ECCC*, TR19-129, Sept. 2019.

[5] 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.

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

[7] 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.

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

[9] O. Goldreich and A. Tal. Matrix rigidity of random Toeplitz matrices. *Computational Complexity*, Vol. 27 (2), pages 305–350, 2018. Preliminary versions in *48th STOC* (2016) and *ECCC* TR15-079 (2015).

[10] O. Goldreich and A. Tal. On Constant-Depth Canonical Boolean Circuits for Computing Multilinear Functions. *ECCC*, TR17-193, 2017.

[11] O. Goldreich and A. Wigderson. On the Size of Depth-Three Boolean Circuits for Computing Multilinear Functions. *ECCC*, TR13-043, 2013.

[12] 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.

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

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

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

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

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

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

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

[20] 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.

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

[22] 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.

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

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

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

[26] 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.

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

[28] 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.

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

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

[31] 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.

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

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

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