

Small-depth Multilinear Formula Lower Bounds for Iterated Matrix Multiplication, with Applications*

Suryajith Chillara
Department of CSE, IIT Bombay.
suryajith@cse.iitb.ac.in

Nutan Limaye
Department of CSE, IIT Bombay.
nutan@cse.iitb.ac.in

Srikanth Srinivasan
Department of Mathematics, IIT Bombay.
srikanth@math.iitb.ac.in

October 8, 2018

Abstract

The complexity of Iterated Matrix Multiplication is a central theme in Computational Complexity theory, as the problem is closely related to the problem of separating various complexity classes within P. In this paper, we study the algebraic formula complexity of multiplying d many 2×2 matrices, denoted IMM_d , and show that the well-known divide-and-conquer algorithm cannot be significantly improved at any depth, as long as the formulas are multilinear.

Formally, for each depth $\Delta \leq \log d$, we show that any product-depth Δ multilinear formula for IMM_d must have size $\exp(\Omega(\Delta d^{1/\Delta}))$. It also follows from this that any multilinear circuit of product-depth Δ for the same polynomial of the above form must have a size of $\exp(\Omega(d^{1/\Delta}))$. In particular, any polynomial-sized multilinear formula for IMM_d must have depth $\Omega(\log d)$, and any polynomial-sized multilinear circuit for IMM_d must have depth $\Omega(\log d / \log \log d)$. Both these bounds are tight up to constant factors.

Our lower bound has the following consequences for multilinear formula complexity.

1. **Depth-reduction:** A well-known result of Brent (JACM 1974) implies that any formula of size s can be converted to one of size $s^{O(1)}$ and depth $O(\log s)$; further, this reduction continues to hold for multilinear formulas. On the other hand, our lower bound implies that any depth-reduction in the multilinear setting cannot reduce the depth to $o(\log s)$ without a superpolynomial blow-up in size.
2. **Circuits vs. formulas:** Any circuit of size s and product-depth Δ can be converted into a formula of product-depth Δ and size $s^{O(\Delta)}$. In the multilinear setting, we show that it is not possible to improve on this significantly for small depths. Formally, our results imply that for all large enough s and $\Delta = o(\log s / \log \log s)$, there is an explicit multilinear polynomial $P_{s,\Delta}$ that has a syntactic multilinear circuit of size s and is such that any multilinear formula of product-depth Δ computing $P_{s,\Delta}$ must have size $s^{\Omega(\Delta)}$.
3. **Separations from general formulas:** Shpilka and Yehudayoff (FnTTCS 2010) asked whether general formulas can be more efficient than multilinear formulas for computing multilinear polynomials. Our result, along with a non-trivial upper bound for IMM_d implied by a result of Gupta, Kamath, Kayal and Saptharishi (SICOMP 2016), shows

*A preliminary version of this work appeared in the proceedings of STACS 2018.

that for any size s and product-depth $\Delta = o(\log s)$, general formulas of size s and product-depth Δ cannot be converted to multilinear formulas of size $s^{O(1)}$ and product-depth Δ , when the underlying field has characteristic zero.

1 Introduction

Algebraic Complexity theory is the study of the complexity of those computational problems that can be described as computing a multivariate polynomial $f(x_1, \dots, x_N) \in \mathbb{F}[x_1, \dots, x_N]$ over elements $x_1, \dots, x_N \in \mathbb{F}$. Many central algorithmic questions such as the Determinant, Permanent, Matrix product etc. can be cast in this framework. The natural computational models that we consider in this setting are models such as *Algebraic circuits*, *Algebraic Branching Programs* (ABPs), and *Algebraic formulas* (or just formulas), all of which use the natural algebraic operations of $\mathbb{F}[x_1, \dots, x_N]$ to compute the polynomial f . These models have by now been the subject of a large body of work with many interesting upper bounds (i.e. circuit constructions) as well as lower bounds (i.e. impossibility results). (See, e.g. the surveys [SY10, Sap15] for an overview of many of these results.)

Despite this, many fundamental questions remain unresolved. An important example of such a question is that of proving lower bounds on the size of formulas for the *Iterated Matrix Multiplication* problem, which is defined as follows. Given d $n \times n$ matrices M_1, \dots, M_d , we are required to compute (an entry of) the product $M_1 \cdots M_d$; we refer to this problem as $\text{IMM}_{n,d}$. Proving superpolynomial lower bounds on the size of formulas for this problem is equivalent to separating polynomial-sized ABPs from polynomial-sized formulas, which is the algebraic analogue of separating the Boolean complexity classes NL and NC¹.

A standard divide-and-conquer algorithm yields the best-known formulas for $\text{IMM}_{n,d}$. More precisely, for any $\Delta \leq \log d$, this approach yields a formula of product-depth¹ Δ and size $n^{O(\Delta d^{1/\Delta})}$ for $\text{IMM}_{n,d}$ and choosing $\Delta = \log d$ yields the current best formula upper bound of $n^{O(\log d)}$, which has not been improved in quite some time. On the other hand, separating the power of ABPs and formulas is equivalent to showing that $\text{IMM}_{n,d}$ does not have formulas of size $\text{poly}(nd)$.

The Iterated Matrix Multiplication problem has many nice features that render its complexity an interesting object to study. For one, it is the algebraic analogue of the Boolean reachability problem, and thus any improved formula upper bounds for $\text{IMM}_{n,d}$ could lead to improved Boolean circuit upper bounds for the reachability problem, which would resolve a long-standing open problem in that area. For another, this problem has strong self-reducibility properties, which imply that improving on the simple divide-and-conquer approach to obtain formulas of size $n^{O(\log d)}$ for any d would lead to improved upper bounds for all $D > d$; this implies that the lower-degree variant is no easier than the higher-degree version of the problem, which can be very useful (e.g. for homogenization [Raz13]). Finally, the connection to the Reachability problem imbues $\text{IMM}_{n,d}$ with a rich combinatorial structure via its graph theoretic interpretation, which has been used extensively in lower bounds for depth-4 arithmetic circuits [FLMS14, KLSS14, KS14, KNS16, KST16].

We study the formula complexity of this problem in the *multilinear* setting, which restricts the underlying formulas to only compute multilinear polynomials at intermediate stages of computation. Starting with the breakthrough work of Raz [Raz06], many lower bounds have been

¹The *product-depth* of an arithmetic circuit or formula is the maximum number of product gates on a path from output to input. If the product-depth of a circuit or formula is Δ , then its depth can be assumed to be at least $2\Delta - 1$ and at most $2\Delta + 1$.

proved for multilinear models of computation [RY08, RY09, RSY08, DMPY12, AKV17]. Further, it is known by a result of Dvir, Malod, Perifel and Yehudayoff [DMPY12] that multilinear ABPs are in fact superpolynomially more powerful than multilinear formulas. Unfortunately, however, this does not imply any non-trivial lower bound for Iterated Matrix Multiplication (see the Related Work section below), and as far as we know, it could well be the case that there are multilinear formulas that beat the divide-and-conquer approach in computing this polynomial.

Here, we are able to show that this is not the case for the problem of multiplying 2×2 matrices (and by extension $c \times c$ matrices for any constant c) at any product-depth. Our main theorem is the following (stated more formally as Theorem 12 later).

Theorem 1. *For $\Delta \leq \log d$, any product-depth Δ multilinear formula that computes $\text{IMM}_{2,d}$ must have size $2^{\Omega(\Delta d^{1/\Delta})}$.*

This lower bound strengthens a result of Nisan and Wigderson [NW97] who prove a similar lower bound in the more restricted *set-multilinear* setting. Our result is also qualitatively different from the previous lower bounds for multilinear formulas [Raz06, Raz04, RY08] since $\text{IMM}_{2,d}$ does in fact have polynomial-sized formulas of product-depth $O(\log d)$ (via the divide-and-conquer approach), whereas we show a superpolynomial lower bound for product-depth $o(\log d)$. This observation leads to interesting consequences for multilinear formula complexity in general, which we now describe.

Depth Reduction. An important theme in circuit complexity is the interplay between the size of a formula or circuit and its depth [Bre74, Spi73, VSBR83, AV08, Tav15]. In the context of algebraic formulas, a result of Brent [Bre74] says that any formula of size s can be converted into another of size $s^{O(1)}$ and depth $O(\log s)$. Further, the proof of this result also yields the same statement for multilinear formulas.

Can the result of Brent be improved? Theorem 1 implies that the answer is no in the multilinear setting (Corollary 15). More precisely, since the $\text{IMM}_{2,d}$ polynomial (over $O(d)$ variables) has formulas of size $\text{poly}(d)$ and depth $O(\log d)$ but no formulas of size $\text{poly}(d)$ and depth $o(\log d)$ (by Theorem 1), we see that any multilinear depth-reduction procedure that reduces the depth of a size- s formula to $o(\log s)$ must incur a superpolynomial blow-up in size. This strengthens a result of Raz and Yehudayoff [RY09], whose results imply that any depth-reduction of multilinear formulas to depth $o(\sqrt{\log s}/\log \log s)$ should incur a superpolynomial blow-up in size. It is also an analogue in the algebraic setting of some recent results proved for Boolean circuits [Ros15, RS17].

Circuits vs Formulas. Our next application deals with the relative power of Algebraic formulas and circuits².

It can be seen that any circuit of size s and product-depth Δ can be converted into a formula of product-depth Δ and size $s^{O(\Delta)}$ by replicating gates where necessary. A natural question to ask is if this is tight. More precisely, is there a procedure that converts circuits of size s and product-depth Δ to algebraic formulas of product-depth Δ and size $s^{o(\Delta)}$?

In the setting of Boolean circuits, answering the question negatively for $\Delta = O(\log s)$ would separate the Boolean circuit class NC^1 from the class NC^2 (or even the smaller class SAC^1). Recently, Rossman [Ros15] and Rossman and Srinivasan [RS17] proved such results for Boolean circuits of depth Δ , where $\Delta \leq \varepsilon \log s / \log \log s$ for some fixed $\varepsilon > 0$.

²Mrinal Kumar, Rafael Oliveira and Benjamin Rossman pointed out this application of our result.

In the algebraic multilinear setting, the question above has already been answered negatively by the work of Raz [Raz04] when $\Delta = O(\log s)$. However, it is unclear how to use his results to prove such a statement for smaller depths. Using Theorem 1, we are able to prove such a result for $\Delta \leq \varepsilon \log s / \log \log s$ for some fixed $\varepsilon > 0$.

Multilinear vs. general formulas. Shpilka and Yehudayoff [SY10] ask the question of whether general formulas can be more efficient at computing multilinear polynomials than multilinear formulas. This is an important question, since we have techniques for proving lower bounds for multilinear formulas, whereas the same question for general formulas (or even depth-3 formulas over large fields) remains wide open.

We are able to make progress towards this question here by showing a separation between the two models for small depths when the underlying field has characteristic zero (Corollary 17). We do this by using Theorem 1 in conjunction with a (non-multilinear) formula *upper bound* for $\text{IMM}_{2,d}$ over fields of characteristic zero due to Gupta et al. [GKKS16]. In particular, the result of Gupta et al. [GKKS16] implies that for any product-depth $\Delta \leq \log d$, the polynomial $\text{IMM}_{2,d}$ has formulas of product-depth Δ and size $2^{O(\Delta d^{1/2\Delta})}$, which is considerably smaller than our lower bound in the multilinear case for small Δ . From this, it follows that for any size parameter s and product-depth $\Delta = o(\log s)$, general formulas of size s and product-depth Δ cannot be converted to multilinear formulas of size $s^{O(1)}$ and product-depth Δ . Improving our result to allow for $\Delta = O(\log s)$ would resolve the question entirely.

Related Work. The multilinear formula model has been the focus of a large body of work on Algebraic circuit lower bounds. Nisan and Wigderson [NW97] proved some of the early results in this model by showing size lower bounds for small-depth *set-multilinear*³ circuits computing $\text{IMM}_{2,d}$. They showed that any product-depth Δ circuit for $\text{IMM}_{2,d}$ must have a size of $2^{\Omega(d^{1/\Delta})}$ matching the upper bound from the divide-and-conquer algorithm for $\Delta = o(\log d / \log \log d)$. Our lower bounds for multilinear formulas imply similar lower bounds for multilinear circuits of product-depth Δ .

Raz [Raz06] proved the first superpolynomial lower bound for multilinear formulas by showing an $n^{\Omega(\log n)}$ lower bound for the $n \times n$ Determinant and Permanent polynomials. This was further strengthened by the results of Raz [Raz04] and Raz and Yehudayoff [RY08] to a similar lower bound for an explicit polynomial family that has polynomial-sized multilinear *circuits*. Raz and Yehudayoff [RY08] also showed that the depth reduction of size s , degree d circuits to circuits of size $s^{O(1)}$ and depth $O(\log d)$ [VSB83] continues to hold for multilinear circuits; it is easily seen that such a circuit yields a multilinear formula of size $s^{O(\log d)}$ that computes the same polynomial. In particular, these results show the tightness of the depth-reduction procedure for algebraic *circuits* in the multilinear setting [VSB83, RY08].

Similar polynomial families were also used in the work of Raz and Yehudayoff [RY09] to prove *exponential* lower bounds for multilinear constant-depth circuits. By proving a tight lower bound for depth- Δ circuits computing an explicit polynomial (similar to the construction of Raz [Raz04]), Raz and Yehudayoff [RY09] showed superpolynomial separations between multilinear circuits of different depths.

In particular, the result of Raz and Yehudayoff [RY09] implies that the polynomial families

³Set-multilinear circuits are further restrictions of multilinear circuits. A set-multilinear circuit for $\text{IMM}_{n,d}$ is defined by the property that each intermediate polynomial computed must be a linear combination of monomials that contain exactly one variable from each matrix M_i ($i \in S$), for some choice of $S \subseteq [d]$.

of [Raz04, RY08], which have formulas of size $n^{O(\log n)}$, cannot be computed by formulas of size less than some $s(n) = n^{\omega(\log n)}$ if the product-depth Δ is $o(\log n / \log \log n)$. This yields the superpolynomial separation between formulas of size s and depth $o(\sqrt{\log s} / \log \log s)$ alluded to above. Unfortunately, these polynomials also have nearly optimal formulas of depth just $O(\log n) = O(\sqrt{\log s})$, so they cannot be used to obtain the optimal size s vs depth $o(\log s)$ separation we obtain here.

Dvir et al. [DMPY12] showed that there is an explicit polynomial on n variables that has multilinear ABPs of size $\text{poly}(n)$ but no multilinear formulas of size less than $n^{\Omega(\log n)}$. One might hope that this yields a superpolynomial lower bound for multilinear formulas computing $\text{IMM}_{N,d}$ for some N, d but this unfortunately does not seem to be the case. The reason for this is that while any polynomial f on n variables that has an ABP of size $\text{poly}(n)$ can be reduced via variable substitutions to $\text{IMM}_{N,d}$ for $N, d = n^{O(1)}$, this reduction might substitute different variables in the $\text{IMM}_{N,d}$ polynomial by the same variable x of f and in the process destroy multilinearity.

Gupta et al. [GKKS16] showed the surprising result that general (i.e. non-multilinear) formulas of depth-3 can beat the divide-and-conquer approach for computing $\text{IMM}_{N,d}$, when the underlying field has characteristic zero. Their result implies that, in this setting, $\text{IMM}_{N,d}$ has product-depth 1 formulas of size $N^{O(\sqrt{d})}$, as opposed to the $N^{O(d)}$ -sized formula that is obtained from the traditional divide-and-conquer approach.⁴ Using the self-reduction properties of $\text{IMM}_{N,d}$, this can be easily seen to imply the existence of $N^{O(\Delta d^{1/2\Delta})}$ -sized formulas of product-depth Δ . This construction uses the fact that the formulas are allowed to be non-multilinear. Our result shows that this cannot be avoided.

Proof Overview. The proof follows a two-step process as in [SY10, DMPY12].

The first step is a “product lemma” where we show that any multilinear polynomial f on n variables that has a small multilinear formula can also be computed as a sum of a small number of polynomials each of which is a product of many polynomials on disjoint sets of variables; if such a term is the product of t polynomials, we call it a t -product polynomial.⁵ It is known [SY10, Lemma 3.5] that if f has a formula of size s , then we can ensure a decomposition into a sum of at most $s+1$ many $\Omega(\log n)$ -product polynomials. We show that if the formula further is known to have depth Δ then the number of factors can be increased to $\Omega(\Delta n^{1/\Delta})$. In particular, note that this is $\omega(\log n)$ as long as $\Delta = o(\log n)$: this allows us to obtain superpolynomial lower bounds for up to this range of parameters.

Similar lemmas were already known in the small-depth setting [RY09], but they do not achieve the parameters of our decomposition. However, the lemma of [RY09] satisfies the additional condition that every factor of each t -product polynomial in the decomposition depends on a “large” number of variables. Here, we only get that each factor depends on a non-zero number of variables, but this is sufficient to prove the lower bound we want.

The second step is to use this decomposition to prove a lower bound. Specifically, we would like to say that the polynomial $\text{IMM}_{2,d}$ has no small decomposition into terms of the above form. This is via a rank argument as in Raz [Raz06]. Specifically, we partition the variables X in our polynomial into two sets Y and Z and consider any polynomial $f(X)$ as a polynomial in the variables in Y with coefficients from $\mathbb{F}[Z]$. The dimension of the space of coefficients (as vectors over the base field \mathbb{F}) is considered as a measure of the complexity of f .

⁴For product-depth 1, this is the brute-force expansion into monomials.

⁵The polynomials in our decomposition can also have a different form which we choose to ignore for now.

It is easy to come up with a partition of the underlying variable set X into Y, Z so that the complexity of $\text{IMM}_{2,d}$ is as large as possible. Unfortunately, we also have simple multilinear formulas that have maximum dimension w.r.t. this partition. Hence, this notion of complexity is not by itself sufficient to prove a lower bound. At this point, we follow an idea of Raz [Raz06] and show something stronger for $\text{IMM}_{2,d}$: we show that its complexity is quite *robust* in the sense that it is full rank w.r.t. many different partitions.

More precisely, we carefully design a large space of *restrictions* $\rho : X \rightarrow Y \cup Z \cup \mathbb{F}$ such that for any restriction ρ , the resulting substitution of $\text{IMM}_{2,d}$ continues to have high complexity w.r.t. the measure defined above. These restrictions are motivated by the combinatorial structure of the underlying polynomial, specifically the connection to Graph Reachability.

The last step is to show that, for any t -product polynomial f , a random restriction from the above space of restrictions transforms it with high probability into a polynomial whose measure is small. Once we have this result, it follows by a simple union bound that given a small multilinear formula, there is a restriction that transforms each term in its decomposition (obtained from the product lemma) into a small complexity polynomial. The subadditivity of rank then shows that the entire formula now has small complexity, and hence it cannot be computing $\text{IMM}_{2,d}$ which by the choice of our restriction has high complexity.

2 Preliminaries

2.1 Basic setup

Unless otherwise stated, let \mathbb{F} be an arbitrary field. Let $d \in \mathbb{N}$ be a growing integer parameter. We define $X^{(1)}, \dots, X^{(d)}$ to be disjoint sets of variables where each $X^{(i)} = \{x_{j,k}^{(i)} \mid j, k \in [2]\}$ is a set of four variables that we think of as forming a 2×2 matrix. Let $X = \bigcup_{i \in [d]} X^{(i)}$.

A polynomial $P \in \mathbb{F}[X]$ is called *multilinear* if the degree of P in each variable $x \in X$ is at most 1. We define the multilinear polynomial $\text{IMM}_d \in \mathbb{F}[X]$ as follows. Consider the matrices $M^{(1)}, \dots, M^{(d)}$ where the entries of $M^{(i)}$ are the variables of $X^{(i)}$ arranged in the obvious way. Define the matrix $M = M^{(1)} \cdots M^{(d)}$; the entries of M are multilinear polynomials over the variables in X . We define

$$\text{IMM}_d = M(1, 1) + M(1, 2),$$

i.e. the sum of the $(1, 1)$ th and $(1, 2)$ th entries of M . Formally,

$$\text{IMM}_d = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{1,1}^{(1)} & x_{1,2}^{(1)} \\ x_{2,1}^{(1)} & x_{2,2}^{(1)} \end{bmatrix} \cdot \begin{bmatrix} x_{1,1}^{(2)} & x_{1,2}^{(2)} \\ x_{2,1}^{(2)} & x_{2,2}^{(2)} \end{bmatrix} \cdots \begin{bmatrix} x_{1,1}^{(d)} & x_{1,2}^{(d)} \\ x_{2,1}^{(d)} & x_{2,2}^{(d)} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

(Note, in particular, that the polynomial IMM_d does not depend on the variables $x_{2,1}^{(1)}$ and $x_{2,2}^{(1)}$.)

This is a slight variant of the Iterated Matrix Multiplication polynomial seen in the literature, as it is usually defined to be either the matrix entry $M(1, 1)$ or the trace $M(1, 1) + M(2, 2)$. Our results can easily be seen to hold for these variants, but we use the definition above for some technical simplicity.

Another standard way of defining the polynomial IMM_d is via graphs. Define the edge-labelled directed acyclic graph $G_d = (V, E, \lambda)$ as follows: the vertex set V is defined to be the disjoint union of vertex sets $V^{(0)}, \dots, V^{(d)}$ where $V^{(i)} = \{v_1^{(i)}, v_2^{(i)}\}$. The edge set E is the set of

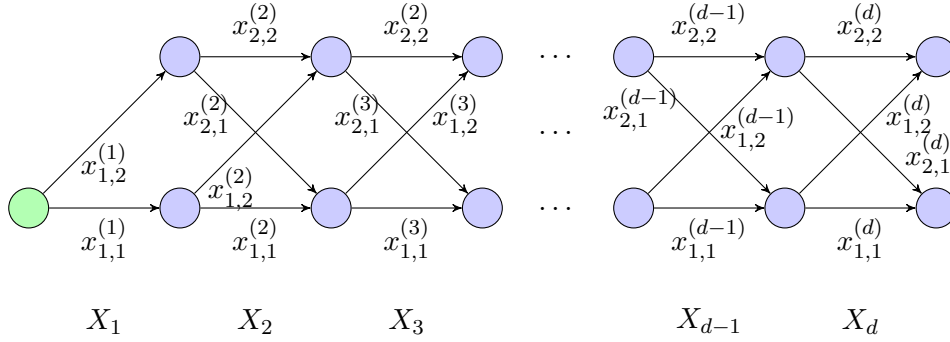


Figure 1: The directed acyclic graph G_d that defines the polynomial IMM_d with its labeling.

all possible edges from some set $V^{(i)}$ to $V^{(i+1)}$ (for all $i < d$). The labelling function $\lambda : E \rightarrow X$ is defined by $\lambda((v_j^{(i)}, v_k^{(i+1)})) = x_{j,k}^{(i+1)}$. See Figure 1 for a depiction of this graph.

Given a path π in the graph G_d , $\lambda(\pi)$ is defined to be the product of all the labels of edges in π . In this notation, IMM_d can be seen to be the following.

$$\text{IMM}_d = \sum_{\substack{\text{paths } \pi \text{ from } v_1^{(0)} \\ \text{to } v_1^{(d)} \text{ or } v_2^{(d)}}} \lambda(\pi) = \sum_{\pi_1, \dots, \pi_d \in \{1,2\}} x_{1,\pi_1}^{(1)} x_{\pi_1,\pi_2}^{(2)} \cdots x_{\pi_{d-1},\pi_d}^{(d)} \quad (1)$$

2.2 Multilinear formulas and circuits

We refer the reader to the standard resources (e.g. [SY10, Sap15]) for basic definitions related to algebraic circuits and formulas. Having said that, we do make a few remarks.

- All the gates in our formulas and circuits will be allowed to have *unbounded* fan-in.
- The size of a formula or circuit will refer to the number of gates (including input gates) in it, and depth of the formula or circuit will refer to the number of gates on the longest path from an input gate to the output gate.
- Further, the *product-depth* of the formula or circuit (as in [RY08]) will refer to the maximum number of product gates on a path from an input gate to the output gate. Note that the product-depth of a formula or circuit can be assumed to be within a factor of two of the overall depth (by collapsing sum gates if necessary).

Multilinear circuits and formulas. An algebraic formula F (resp. circuit C) computing a polynomial from $\mathbb{F}[X]$ is said to be *multilinear* if each gate in the formula (resp. circuit) computes a multilinear polynomial⁶. Moreover, a formula F (resp. circuit C) is said to be *syntactic multilinear* if for each multiplication gate Φ of F (resp. C) with children Ψ_1, \dots, Ψ_t , we have $\text{Supp}(\Psi_i) \cap \text{Supp}(\Psi_j) = \emptyset$ for each $i \neq j$, where $\text{Supp}(\Phi)$ denotes the set of variables that appear in the subformula (resp. subcircuit) rooted at Φ . Finally, for $\Delta \geq 1$, we say that a multilinear formula (resp. circuit) is a $(\Sigma\Pi)^\Delta\Sigma$ formula (resp. circuit) if the output gate is a sum

⁶It is important to note that multilinear polynomials can also be computed by a non-multilinear formulas and circuits.

gate and along any path, the sum and product gates alternate, with each product gate appearing exactly Δ times and the bottom gate being a sum gate. We can define $(\Sigma\Pi)^\Delta$, $\Sigma\Pi\Sigma$, $\Sigma\Pi\Sigma\Pi$ formulas and circuits similarly.

For a gate Φ in a syntactically multilinear formula, we define a set of variables $\text{Vars}(\Phi)$ in a top-down fashion as follows.

Definition 2. *Let F be a syntactically multilinear formula computing a polynomial on the variable set X . For the output gate Φ , we define $\text{Vars}(\Phi) = X$. If Φ is a sum gate with children Ψ_1, \dots, Ψ_k and $\text{Vars}(\Phi) = S \subseteq X$, then for each $1 \leq i \leq k$, $\text{Vars}(\Psi_i) = S$. If Φ is a product gate with children Ψ_1, \dots, Ψ_k and $\text{Vars}(\Phi) = S \subseteq X$, then $\text{Vars}(\Psi_i) = \text{Supp}(\Psi_i)$ for $1 \leq i \leq k-1$ and $\text{Vars}(\Psi_k) = S \setminus \left(\bigcup_{i=1}^{k-1} \text{Vars}(\Psi_i) \right)$.*

It is easy to see that $\text{Vars}(\cdot)$ satisfies the properties listed in the following proposition.

Proposition 3. *For each gate Φ in a syntactically multilinear formula F , let $\text{Vars}(\Phi)$ be defined as in Definition 2 above.*

1. *For any gate Φ in F , $\text{Supp}(\Phi) \subseteq \text{Vars}(\Phi)$.*
2. *If Φ is a sum gate, with children $\Psi_1, \Psi_2, \dots, \Psi_k$, then $\forall i \in [k]$, $\text{Vars}(\Psi_i) = \text{Vars}(\Phi)$.*
3. *If Φ is a product gate, with children $\Psi_1, \Psi_2, \dots, \Psi_k$, then $\text{Vars}(\Phi) = \bigcup_{i=1}^k \text{Vars}(\Psi_i)$ and the sets $\text{Vars}(\Psi_i)$ ($i \in [k]$) are pairwise disjoint.*

We will use the following structural results that convert general multilinear circuits (resp. formulas) to $(\Sigma\Pi)^\Delta\Sigma$ circuits (resp. formulas).

Lemma 4 (Raz and Yehudayoff [RY09], Claims 2.3 and 2.4). *For any multilinear formula F of product-depth at most Δ and size at most s , there is a syntactic multilinear $(\Sigma\Pi)^\Delta\Sigma$ formula F' of size at most $(\Delta + 1)^2 \cdot s$ computing the same polynomial as F .*

Lemma 5 (Raz and Yehudayoff [RY09], Lemma 2.1). *For any multilinear circuit C of product-depth at most Δ and size at most s , there is a syntactic multilinear $(\Sigma\Pi)^\Delta\Sigma$ formula F of size at most $(\Delta + 1)^2 \cdot s^{2\Delta+1}$ computing the same polynomial as C .*

We will also need the following structural result.

Lemma 6 (Raz, Shpilka and Yehudayoff [RSY08], Claim 5.6). *Let F be a syntactic multilinear formula computing a polynomial f and let Φ be any gate in F computing a polynomial g . Then f can be written as $f = Ag + B$, where $A \in \mathbb{F}[X \setminus \text{Vars}(\Phi)]$, $B \in \mathbb{F}[X]$ and B is computed by replacing Φ by 0 in F .*

A standard divide-and-conquer approach (see [AK10, Proposition 3.10]) yields the best-known multilinear formulas and circuits for IMM_d for all depths.

Lemma 7. *For each $\Delta \leq \log d$,⁷ IMM_d is computed by a syntactic multilinear $(\Sigma\Pi)^\Delta$ circuit C_Δ of size at most $d^{O(1)} \cdot 2^{O(d^{1/\Delta})}$ and a syntactic multilinear $(\Sigma\Pi)^\Delta$ formula F_Δ of size at most $2^{O(\Delta d^{1/\Delta})}$.*

⁷All our logarithms will be to base 2.

Proof sketch. Let us recall that the IMM_d polynomial is defined over the matrices $M^{(1)}, M^{(2)}, \dots, M^{(d)}$. Define the IMM'_d polynomial to be the polynomial corresponding to, say, the $(1, 1)$ th entry of the product $M = M^{(1)} \dots M^{(d)}$.⁸ It suffices to prove the statement of the lemma for IMM'_d in place of IMM_d since the polynomial IMM_d is a sum of two copies of IMM'_d .

We show how to inductively construct the circuit C'_Δ for IMM'_d . For the base case $\Delta = 1$, we note that the brute force $\Sigma\Pi$ formula for IMM'_d gives us a circuit of the required form of size $2^{O(d)}$. Now, assume that $\Delta > 1$. Let us divide the matrices $M^{(1)}, \dots, M^{(d)}$ into $t = d^{1/\Delta}$ contiguous blocks of size d/t each, say B_1, B_2, \dots, B_t . Let M'_i denote the product of the matrices in the i th block and let $P_{u,v}^{(i)}$ (for $u, v \in [2]$) denote the polynomial that is the (u, v) th entry of M'_i . Note that each such polynomial $P_{u,v}^{(i)}$ is a copy of the polynomial $\text{IMM}'_{d/t}$.

Since the matrix M is the product of M'_1, \dots, M'_t , we see that the polynomial IMM'_d can now be expressed in terms of M'_1, \dots, M'_t as follows.

$$\text{IMM}'_d = \text{IMM}'_t(M'_1, \dots, M'_t) = \sum_{u_1, \dots, u_{t-1} \in \{1, 2\}} P_{1, u_1}^{(1)} \cdot P_{u_1, u_2}^{(2)} \dots P_{u_{t-1}, 1}^{(t)}, \quad (2)$$

Now, for each entry $P_{u,v}^{(i)}$ ($u, v \in [2]$) of M'_i ($i \in [t]$), we have an inductively constructed $(\Sigma\Pi)^{(\Delta-1)}$ syntactic multilinear circuit. Using these and the trivial $\Sigma\Pi$ formula for IMM'_t (as shown in (2)) of size $2^{O(t)}$, we obtain a $(\Sigma\Pi)^\Delta$ syntactic multilinear circuit for IMM'_d . This completes the description of the circuit.

We get the following recursive formula for the size $s(d, \Delta)$ of the $(\Sigma\Pi)^\Delta$ circuit thus constructed. For $\Delta = 1$, we have $s(d, \Delta) = 2^{O(d)}$ as noted above. For $\Delta > 1$, we get

$$s(d, \Delta) \leq O(t) \cdot s(d/t, \Delta - 1) + 2^{O(t)}.$$

Solving this recursion gives us the claimed bound of $d^{O(1)} \cdot 2^{O(d^{1/\Delta})}$.

We can obtain a $(\Sigma\Pi)^\Delta$ syntactic multilinear formula F'_Δ for IMM'_d by simply expanding out the circuit C'_Δ , replicating gates as and when necessary. It can be checked that each sum gate in the above construction has fan-in $2^{O(t)}$ and each product gate has fan-in t .

The total number of gates in the formula F'_Δ at depth i is equal to the number of paths of length i from a gate at depth i in C'_Δ to the output gate, which is at most $2^{O(it)}$. In particular, this implies that the number of gates in F'_Δ is at most $2^{O(\Delta t)} = 2^{O(\Delta d^{1/\Delta})}$. \square

We will show that the above bounds are nearly tight in the multilinear setting. If we remove the multilinear restriction on $(\Sigma\Pi)^\Delta\Sigma$ formulas computing IMM_d , we can get better upper bounds, as long as the underlying field has characteristic zero. The proof of this is very similar to the proof of Lemma 7, with the difference being that we use a highly non-trivial result of Gupta et al. [GKKS16] for the base case of the induction.

Lemma 8 (follows from [GKKS16]). *Let \mathbb{F} be a field of characteristic zero. For each $\Delta \leq \log d$, IMM_d has a $(\Sigma\Pi)^\Delta\Sigma$ non-multilinear formula F_Δ of size at most $2^{O(\Delta d^{1/(2\Delta)})}$.*

Proof sketch of Lemma 8. As in the proof of Lemma 7, we show the lemma statement for IMM'_d . We will proceed by induction on Δ .

For the base case, we need the following claim (implicit in [GKKS16]).

⁸Clearly, the polynomial corresponding to the (u, v) th entry (for any $u, v \in [2]$) is the same polynomial as IMM'_d , up to a renaming of variables.

Claim 9. For $t > 1$, IMM_t has a $\Sigma\Pi\Sigma$ formula of size at most $c^{\sqrt{t}}$ for some fixed constant $c > 1$.

Proof of Claim 9. Applying Lemma 7 with $\Delta = 2$ yields a $\Sigma\Pi\Sigma\Pi$ formula F for IMM_d of size $2^{O(\sqrt{d})}$. It can be checked from the proof of Lemma 7 that this formula satisfies the additional property that all the product gates in the formula have fan-in $O(\sqrt{t})$.

Over any field \mathbb{F} of characteristic zero⁹, Gupta et al. [GKKS16] showed that any $\Sigma\Pi\Sigma\Pi$ formula of size s where all product gates have fan-in at most k can be converted into a $\Sigma\Pi\Sigma$ formula of size $\text{poly}(s) \cdot 2^{O(k)}$. Applying this result to the formula F obtained above, we get that IMM_t can indeed be computed by a $\Sigma\Pi\Sigma$ formula of size at most $2^{O(\sqrt{t})}$, over any field \mathbb{F} of characteristic zero. This proves the claim. \square

The above immediately yields the statement of the lemma for $\Delta = 1$. For $\Delta > 1$, we proceed as in the proof of Lemma 7. We divide the matrices $M^{(1)}, \dots, M^{(d)}$ into $t = d^{1/\Delta}$ blocks of size d/t each. Let M'_i denote the product of the matrices in the i th block. Using the identity

$$\text{IMM}'_d = \text{IMM}'_t(M'_1, \dots, M'_t), \quad (3)$$

we can construct a $(\Sigma\Pi)^\Delta\Sigma$ circuit for IMM'_d as follows. We inductively construct a $(\Sigma\Pi)^{\Delta-1}\Sigma$ circuit for each entry of each M'_i (which is a copy of $\text{IMM}'_{d/t}$) and combine them with a $2^{O(\sqrt{t})}$ -sized $\Sigma\Pi\Sigma$ formula for IMM'_t (given to us by Claim 9). Collapsing two adjacent levels of sum gates gives us a $(\Sigma\Pi)^\Delta\Sigma$ circuit for IMM'_d .

It can be checked that this yields a $(\Sigma\Pi)^\Delta\Sigma$ circuit C of size $d^{O(1)} \cdot 2^{O(d^{1/2\Delta})}$. Further, it can also be checked inductively that the output gate of the circuit has fan-in at most $c^{\sqrt{t}}$ and every other gate in the circuit has fan-in at most $c^{2\sqrt{t}}$.

We expand out this circuit C to obtain the formula F_Δ . Since the fan-in of each gate is $2^{O(\sqrt{t})}$, we see that the size of the formula F_Δ is at most $2^{O(\Delta\sqrt{t})} = 2^{O(\Delta d^{1/2\Delta})}$ as required. \square

2.3 Some probabilistic estimates

The following version of the Chernoff bound [Che52, Hoe63] is from the book of Dubashi and Panconesi [DP09, Theorem 1.1].

Theorem 10 (Chernoff bound). *Let W_1, \dots, W_n be independent $\{0, 1\}$ -valued random variables and let $W = \sum_{i \in [n]} W_i$. Then we have the following.*

1. For any $\varepsilon > 0$,

$$\Pr[W > (1 + \varepsilon) \mathbf{E}[W]] \leq \exp\left(-\frac{\varepsilon^2}{3} \mathbf{E}[W]\right) \quad \text{and} \quad \Pr[W < (1 - \varepsilon) \mathbf{E}[W]] \leq \exp\left(-\frac{\varepsilon^2}{2} \mathbf{E}[W]\right).$$

2. For any $t > 2e \mathbf{E}[W]$,

$$\Pr[W > t] \leq 2^{-t}.$$

Here we state a useful proposition that we shall later use in our analysis.

Proposition 11. *Let m be a positive integer. Let a_1, a_2, \dots, a_m be m independent Boolean random variables such that for all $i \in [m]$, $\Pr[a_i] = 1/2$. Then, $\Pr[\#\{i \mid a_i = 1\} \text{ is odd}] = 1/2$.*

⁹It also works if the characteristic field \mathbb{F} is positive but suitably large.

3 Statement of the main theorem and consequences

The main theorem of this section is the following lower bound.

Theorem 12. *Let $d \geq 1$ be a growing parameter and fix any $\Delta \leq \log d$. Any syntactic multilinear $(\Sigma\Pi)^\Delta\Sigma$ formula for IMM_d must have a size of $2^{\Omega(\Delta d^{1/\Delta})}$.*

Putting together Theorem 12 with Lemmas 4 and 5, we have the following (immediate) corollaries.

Corollary 13. *Let $d \geq 1$ be a growing parameter and fix any $\Delta \leq \log d / \log \log d$. Any multilinear circuit of product-depth at most Δ for IMM_d must have a size of $2^{\Omega(d^{1/\Delta})}$. In particular, any polynomial-sized multilinear circuit for IMM_d must have a product-depth of $\Omega(\log d / \log \log d)$.*

Corollary 14. *Let $d \geq 1$ be a growing parameter and fix any $\Delta \leq \log d$. Any multilinear formula of product-depth at most Δ for IMM_d must have a size of $2^{\Omega(\Delta d^{1/\Delta})}$. In particular, any polynomial-sized multilinear formula for IMM_d must have a product-depth of $\Omega(\log d)$.*

Since the product-depth of a formula is at most its depth, Lemma 7 and Corollary 14 further imply the following.

Corollary 15 (Tightness of Brent's depth-reduction for multilinear formulas). *For each $d \geq 1$, there is an explicit polynomial F_d defined on $O(d)$ variables such that F_d has a multilinear formula of size $d^{O(1)}$, but any multilinear formula of depth $o(\log d)$ for F_d must have a size of $d^{\omega(1)}$.*

Putting together Theorem 12 with Lemma 7, we see that circuits of size s and product-depth Δ cannot always be converted into formulas of product-depth Δ and size $s^{o(\Delta)}$ in the multilinear setting.

Corollary 16 (Separation of formulas and circuits at small depth). *The following holds for some fixed constant $\varepsilon > 0$. Let s and Δ be growing parameters with $\Delta \leq \varepsilon \log s / \log \log s$. There is an explicit polynomial $F_{s,\Delta}$ that is computed by a $(\Sigma\Pi)^\Delta$ syntactic multilinear circuit of size at most s but any multilinear formula of product-depth at most Δ that computes $F_{s,\Delta}$ must be of size $s^{\Omega(\Delta)}$.*

Proof. We choose $F_{s,\Delta}$ to be IMM_d for suitable d . Observe that Lemma 7 implies the following: as long as $d^{1/\Delta} \geq \log d$, IMM_d is computed by a $(\Sigma\Pi)^\Delta$ syntactic multilinear circuit of size at most $s(d, \Delta) = 2^{O(d^{1/\Delta})}$.

For some suitable constant $\varepsilon > 0$, we choose $d = (\varepsilon \log s)^\Delta$. Note that if $\Delta \leq \varepsilon \log s / \log \log s$, then

$$d^{1/\Delta} = \varepsilon \log s \geq \Delta \log \log s \geq \log d.$$

Also note that if ε is a sufficiently small constant, then $s(d, \Delta) = s$ and hence $F_{s,\Delta} = \text{IMM}_d$ can be computed by a $(\Sigma\Pi)^\Delta$ syntactic multilinear circuit of size at most s .

On the other hand, by Corollary 14 above, it follows that any multilinear formula of product-depth at most Δ computing $F_{s,\Delta}$ must have size $s^{\Omega(\Delta)}$. This proves the corollary. \square

Finally, we also obtain a separation between multilinear and general formulas.

Corollary 17 (Separation of multilinear formulas and general formulas over zero characteristic). *Let \mathbb{F} be a field of characteristic zero. Let $s \in \mathbb{N}$ be any growing parameter and $\Delta \in \mathbb{N}$ be such that $\Delta = o(\log s)$. There is an explicit multilinear polynomial $F_{s,\Delta}$ such that $F_{s,\Delta}$ has a $(\Sigma\Pi)^\Delta\Sigma$ formula of size s , but any multilinear formula for $F_{s,\Delta}$ of product-depth at most Δ must have a size of $s^{\omega(1)}$.*

Proof. We choose the polynomial $F_{s,\Delta}$ to be IMM_d for a suitable d and then simply apply Corollary 14 and Lemma 8 to obtain the result. Details follow.

Say $\Delta = \frac{\log s}{f(s)}$ for some $f(s) = \omega(1)$. By Lemma 8, for any d , IMM_d has a $(\Sigma\Pi)^\Delta\Sigma$ (non-multilinear) formula of size $s(d, \Delta) = 2^{O(\Delta d^{1/2\Delta})}$; we choose d so that $s(d, \Delta) = s$. It can be checked that $d = \Theta(f(s))^{2\Delta}$.

Having chosen d as above, we define $F_{s,\Delta} = \text{IMM}_d$. Clearly, $F_{s,\Delta}$ has a (non-multilinear) formula of product-depth Δ and size at most s . On the other hand, by Theorem 12, any multilinear product-depth Δ formula for IMM_d must have size at least

$$2^{\Omega(\Delta d^{1/\Delta})} = s^{\Omega(d^{1/2\Delta})} = s^{\Omega(f(s))} = s^{\omega(1)},$$

which proves the claim.

A similar proof shows that for d as chosen above, IMM_d in fact has no multilinear formulas of size $s^{O(1)}$ and product-depth up to $(2 - \varepsilon)\Delta$ for any absolute constant ε . \square

4 Proof of Main Theorem (Theorem 12)

Our proof follows a two-step argument as in [Raz06, RY09] (see the exposition in [SY10, Section 3.6]).

Step1 – The product lemma

The first step is a “product-lemma” for multilinear formulas.

We define a polynomial $f \in \mathbb{F}[X]$ to be a *t-product polynomial* if we can express f as $f = f_1 \cdots f_t$, and can find a partition of X into *non-empty* sets X_1^f, \dots, X_t^f such that f_i is a multilinear polynomial from $\mathbb{F}[X_i^f]$.¹⁰ We say that X_i^f is the set *ascribed* to f_i in the t -product polynomial f . We use $\text{Vars}(f_i)$ (with a slight abuse of notation¹¹) to denote X_i^f . We drop f from the superscript if f is clear from the context.

We define $f \in \mathbb{F}[X]$ to be *r-simple* if for some $r' \leq r$, f is an $(r' + 1)$ -product polynomial of the form $f = L_1 \cdots L_{r'} \cdot G$, where $L_1, \dots, L_{r'}$ are polynomials of degree at most 1, and the sets $X_1^f, \dots, X_{r'}^f$ ascribed to these linear polynomials satisfy $\left| \bigcup_{i \leq r'} X_i^f \right| \geq 400r$. We prove the following.

Lemma 18. *Assume $1 \leq \Delta \leq \log d$. Assume that $f \in \mathbb{F}[X]$ can be computed by a syntactic multilinear $(\Sigma\Pi)^\Delta\Sigma$ formula F of size at most s . Then, f is the sum of at most s many t -product polynomials and at most s many t -simple polynomials for $t = \Omega(\Delta d^{1/\Delta})$.*

¹⁰Note that we do not need f_i to depend non-trivially on all (or any) of the variables in X_i^f . For example, $f_i \in \mathbb{F}[X_i^f]$ could be the constant polynomial 1.

¹¹ $\text{Vars}(\cdot)$ is used to describe variables ascribed to gates in a circuit as well as to denote variables ascribed to polynomials.

While our proof of the product lemma is motivated by earlier work [SY10, HY11, RY09], we give slightly better parameters, which turn out to be crucial for proving tight lower bounds. In particular, [RY09, Claim 5.5] yields the above with $t = \Omega(d^{1/\Delta})$.

We note that our parameters are optimal as any improvement in this regard would imply a better lower bound for IMM_d which is not possible because of a matching upper bound.

Proof of Lemma 18. Let F be a $(\Sigma\Pi)^\Delta\Sigma$ syntactic multilinear formula of size at most s computing f . We use layer i to denote the set of gates at distance i from the leaves. So in our formula, layer 1 contains only sum gates, layer 2 contains only product gates, and so on. Let $r = \Delta d^{1/\Delta}/400$.

We will prove by induction on the size s of the formula F that f is the sum of at most s polynomials, each of which is either a t -product polynomial or a t -simple polynomial for $t = \Delta d^{1/\Delta}/1000$.

The base case of the induction, corresponding to $s = 0$, is trivial.

Case 1: Suppose there is a gate Φ in layer 2 such that Φ computes a polynomial g and has fan-in at least t . Then we use Lemma 6 and decompose f as $Ag + B$. Here Ag is a t -product polynomial. Since B is computed by a formula of size at most $s - 1$, we are done by induction.

Case 2: Suppose the above case does not hold, i.e. all the gates at layer 2 have a fan-in of at most t . Now, if there is a gate Φ in layer 2 such that $|\text{Vars}(\Phi)| \geq 400r$ then we will decompose F using Lemma 6 and obtain $f = Ag + H$, where Ag is t -simple since $|\text{Vars}(\Phi)| \geq 400r \geq 400t$. Again, since H has a formula of size at most $s - 1$, and we are done by induction.

Case 3: Now assume that neither of the above cases is applicable. Since neither Case 1 nor Case 2 above is applicable to F , each gate Φ in layer 2 satisfies $|\text{Vars}(\Phi)| < p := 400r$. This immediately implies that $\Delta \geq 2$, since in the case of a $\Sigma\Pi\Sigma$ formula, we have $|\text{Vars}(\Phi)| = |X|$ by Proposition 3 item 2 but $p = 400r \leq d < |X|$.

If $\Delta \geq 2$, we use the following lemma.

Lemma 19. *Let $n, p \in \mathbb{N}$. Assume $2 \leq \Delta \leq 2 \log(n/p)$. Let f be computed by a syntactically multilinear $(\Sigma\Pi)^\Delta\Sigma$ formula F of size at most s over a set of n variables. Let $\Phi_1, \Phi_2, \dots, \Phi_{s'}$, where $s' \leq s$, be the product gates at layer 2 such that for all i , $|\text{Vars}(\Phi_i)| \leq p$. Then f can be expressed as a sum of at most s many T -product polynomials where $T = (\Delta(n/p)^{1/(\Delta-1)})/100$.*

The above lemma is applicable in our situation since we have $2 \leq \Delta \leq \log d$ and $n = |X| = 4d$, which implies that

$$(n/p) = (n/400r) = n/(\Delta d^{1/\Delta}) \geq n/(2\sqrt{d}) \geq \sqrt{d} \geq 2^{\Delta/2}.$$

Lemma 19 now yields a decomposition of f as a sum of at most s many T -product polynomials where

$$T = \Delta \cdot \frac{(n/400r)^{\frac{1}{\Delta-1}}}{100} \geq \frac{\Delta}{100} \cdot \left(\frac{d}{\Delta d^{1/\Delta}} \right)^{\frac{1}{\Delta-1}} = \frac{\Delta d^{1/\Delta}}{100 \Delta^{1/(\Delta-1)}} \geq \frac{\Delta d^{1/\Delta}}{200}.$$

Since $T \geq t$, these T -product polynomials are also t -product polynomials. This finishes the proof of the claim modulo the proof of Lemma 19, which we present below. \square

Proof of Lemma 19. We shall prove by induction on the depth Δ that we can take $T = t(n, \Delta) = (\Delta - 1) \left((n/p)^{1/(\Delta-1)} - 1 \right)$ for any n and $\Delta \geq 2$ (for this statement, we don't assume that $\Delta \leq 2 \log(n/p)$). However, since $\Delta \leq 2 \log(n/p)$, this statement implies that $T \geq \Delta(n/p)^{1/(\Delta-1)}/100$.

Let X denote the set of all n underlying variables.

The base case is when $\Delta = 2$. Here, we have a $\Sigma\Pi\Sigma\Pi\Sigma$ formula such that for all Φ at layer 2, $|\text{Vars}(\Phi)| \leq p$. Let Ψ be the output (sum) gate of the formula and Ψ_1, \dots, Ψ_r be the product gates feeding into it; further let f_i be the polynomial computed by Ψ_i . We claim that each f_i is an (n/p) -product polynomial. If this is true, we are done since $f = f_1 + \dots + f_r$ and r is at most s .

To show that f_i is an (n/p) -product polynomial, it suffices to show that each Ψ_i has fan-in at least (n/p) . This follows since each Φ at layer 2 satisfies $|\text{Vars}(\Phi)| \leq p$ and for each sum gate Φ' at layer 3, we have $\text{Vars}(\Phi') = \text{Vars}(\Phi)$ for any gate Φ at layer 2 feeding into Φ' (Proposition 3 item 2). By Proposition 3 item 3, the fan-in of each Ψ_i at layer 4 must thus be at least (n/p) . This concludes the base case.

Now consider $\Delta \geq 3$. Say we have a polynomial f that is computed by a $(\Sigma\Pi)^\Delta\Sigma$ formula F of size at most s and top fan-in (say) r . Let Ψ be the output gate of F and Ψ_1, \dots, Ψ_r the product gates feeding into it; let f_i be the polynomial computed by Ψ_i . It suffices to show that each f_i is the sum of at most s_i many $t(n, \Delta)$ -product polynomials, where s_i is the size of the subformula rooted at Ψ_i . We show this now.

Fix any $i \in [r]$. Let the children of Ψ_i be $\Psi_{i,1}, \dots, \Psi_{i,k}$ and assume that they compute the polynomials $f_{i,1}, \dots, f_{i,k}$. Since $X = \text{Vars}(\Psi) = \bigcup_{j=1}^k \text{Vars}(\Psi_{i,j})$ (Proposition 3 item 3), there must be some gate $\Psi_{i,j}$ feeding into Ψ_i such that $|\text{Vars}(\Psi_{i,j})| \geq n/k$; w.l.o.g., assume that $j = 1$. Applying the induction hypothesis for depth $(\Delta - 1)$ formulas to the polynomial $f_{i,1} \in \mathbb{F}[\text{Vars}(\Psi_{i,1})]$ computed by the subformula rooted at $\Psi_{i,1}$, we obtain

$$f_{i,1} = \sum_{\ell=1}^{s_i} h_{i,1,\ell}$$

where each $h_{i,1,\ell}$ is a $t(n/k, \Delta - 1)$ -product polynomial. Hence, we see that

$$f_i = f_{i,1} \cdots f_{i,k} = \sum_{\ell=1}^{s_i} h_{i,1,\ell} f_{i,2} \cdots f_{i,k}.$$

Each term in the above decomposition of f_i is a t' -product polynomial for $t' = t(n/k, \Delta - 1) + (k - 1)$ where k is the fan-in of Ψ_i . Some calculus (see Claim 20 below) now shows that $t' \geq t(n, \Delta)$. This completes the induction.

Claim 20. Define $g : [1, \infty) \rightarrow \mathbb{R}$ by $g(x) = (\Delta - 2)((n/xp)^{1/(\Delta-2)} - 1) + (x - 1)$. Then g has a global minimum at $x_0 = (n/p)^{1/(\Delta-1)}$. Further, $g(x_0) = t(n, \Delta)$.

Proof. Consider the first and second order derivatives of $g(x)$: $g'(x)$ and $g''(x)$ respectively.

$$g'(x) = 1 - \left(\frac{n}{p}\right)^{1/(\Delta-2)} \cdot \frac{1}{x^{(\Delta-1)/(\Delta-2)}},$$

$$g''(x) = \frac{\Delta - 1}{\Delta - 2} \cdot \left(\frac{n}{p}\right)^{1/(\Delta-2)} \cdot \frac{1}{x^{(2\Delta-3)/(\Delta-2)}}.$$

Let $x_0 = (n/p)^{1/(\Delta-1)}$. It is easy to see that x_0 is a zero of the function $g'(x)$ and $g''(x_0) > 0$, thus making it a local minima. We can further observe that the second derivative of g over the domain $[1, \infty)$ (and when $\Delta \geq 2$) is strictly positive everywhere. Hence, the function g is a strict convex function and x_0 is indeed the global minimum of g . Further,

$$\begin{aligned} g(x_0) &= (\Delta - 2) \left(\frac{n}{p} \cdot \left(\frac{p}{n} \right)^{1/(\Delta-1)} \right)^{1/(\Delta-2)} + \left(\frac{n}{p} \right)^{1/(\Delta-1)} - 1 \\ &= (\Delta - 1) \left(\frac{n}{p} \right)^{1/(\Delta-1)} - 1 \\ &= t(n, \Delta). \end{aligned}$$

□

This finishes the proof of Lemma 19. □

Step 2 – Rank measure and the hard polynomial

Outline. The second step is to show that any decomposition of IMM_d of the form described in Lemma 18 must have many terms. Our proof of this step is inspired by the proof of the multilinear formula lower bound of Raz [Raz06] for the determinant and also the slightly weaker lower bound of Nisan and Wigderson [NW97] for IMM_d in the *set-multilinear* case. Following [Raz06], we define a suitable *random restriction* ρ , of the IMM_d polynomial by assigning variables from the underlying variable set X to $Y \cup Z \cup \{0, 1\}$, where Y and Z are disjoint sets of new variables. The restriction sets distinct variables in X to distinct variables in $Y \cup Z$ or constants, and hence preserves multilinearity.

Having performed the restriction, we consider the *partial derivative matrix* of the restricted polynomial, which is defined as follows.

Definition 21 (Partial derivative matrix). *Let $g \in \mathbb{F}[Y \cup Z]$ be a multilinear polynomial. Define the $2^{|Y|} \times 2^{|Z|}$ matrix $M_{(Y,Z)}(g)$ as follows: the rows and columns of $M_{(Y,Z)}(g)$ are labelled by distinct multilinear monomials in Y and Z respectively and the (m_1, m_2) th entry of $M_{(Y,Z)}(g)$ is the coefficient of the monomial $m_1 \cdot m_2$ in g .*

Our restriction is defined to have the following two properties.

1. Let g be the restriction of the IMM_d polynomial under ρ . The rank of $M_{(Y,Z)}(g)$ is equal to its maximum possible value (i.e. $\min\{2^{|Y|}, 2^{|Z|}\}$) with probability 1.
2. On the other hand, let f be either a t -product polynomial or a t -simple polynomial, and let f' denote its restriction under ρ . Then, the rank of $M_{(Y,Z)}(f')$ is small with high probability.

Now, if IMM_d has a $(\Sigma\Pi)^\Delta\Sigma$ formula F of small size, then it is a sum of a small number of t -product and t -simple polynomials by Lemma 18 and hence by a union bound, we will be able to find a restriction under which the partial derivative matrices of each of these polynomials have a small rank. By the subadditivity of rank, this will imply that for some restriction g of IMM_d , $M_{(Y,Z)}(g)$ will have low rank, contradicting the first property of our restriction.

Our Restriction. To make the above precise, we first define our restrictions. Let $\tilde{Y} = \{y_1, \dots, y_d\}$ and $\tilde{Z} = \{z_1, \dots, z_d\}$ be two disjoint sets of variables. A restriction ρ is a function mapping variables X to elements of $\tilde{Y} \cup \tilde{Z} \cup \{0, 1\}$.

Recall that for all $i \in [d]$, $M^{(i)}$ is the 2×2 matrix whose (u, v) th entry is $x_{u,v}^{(i)}$. Let I and E denote the standard 2×2 identity matrix and the 2×2 flip permutation matrix respectively. For $a \in \{1, 2\}$, we use \bar{a} to denote the other element of the set.

Algorithm 1 Sampling algorithm \mathcal{S}

```

1: Choose  $\pi$  uniformly at random from  $\{1, 2\}^d$ . Define  $\pi(0) = 1$ .
2: Choose  $a$  uniformly at random from  $\{0, 1\}^d$ . Let  $A = \{i \mid a_i = 1\}$ .
3: for  $i \in [d]$  do
4:   Let  $b_i = 0$  if  $\pi(i-1) = \pi(i)$  and  $1$  if  $\pi(i-1) \neq \pi(i)$ .
5: end for
6: for  $i = 1$  to  $d$  do
7:   if  $i \notin A$  then
8:     Choose  $\rho|_{X^{(i)}}$  such that  $M^{(i)} = I$  if  $b_i = 0$  and  $M^{(i)} = E$  if  $b_i = 1$ . (In particular, all
     variables in  $X^{(i)}$  are set to constants from  $\{0, 1\}$ .)
9:   else if  $i \in A$  and  $i$  is the  $j$ th smallest element of  $A$  for odd  $j$  then
10:    Fix
        
$$\rho(x_{u,v}^{(i)}) = \begin{cases} y_{\lceil j/2 \rceil} & \text{if } u = \pi(i-1) \text{ and } v = \pi(i), \\ 1 & \text{if } u = \pi(i-1) \text{ and } v = \overline{\pi(i)}, \\ 0 & \text{otherwise.} \end{cases}$$

11:   else
12:     Now,  $i \in A$  and  $i$  is the  $j$ th smallest element of  $A$  for even  $j$ . We fix
        
$$\rho(x_{u,v}^{(i)}) = \begin{cases} z_{\lceil j/2 \rceil} & \text{if } u = \pi(i-1) \text{ and } v = \pi(i), \\ 1 & \text{if } u = \overline{\pi(i-1)} \text{ and } v = \pi(i), \\ 0 & \text{otherwise.} \end{cases}$$

13:   end if
14: end for

```

We give a procedure \mathcal{S} for sampling a random restriction $\rho : X \rightarrow \tilde{Y} \cup \tilde{Z} \cup \{0, 1\}$ in Algorithm 1. Based on the output ρ of \mathcal{S} , we define the (random) sets $Y = \tilde{Y} \cap \text{Img}(\rho)$ and $Z = \tilde{Z} \cap \text{Img}(\rho)$. Let $m = m(\rho) = \min\{|Y|, |Z|\}$.

We observe the following simple properties of ρ .

Observation 22. *The restriction ρ satisfies the following.*

1. $|Y| = \lceil |A|/2 \rceil$ and $|Z| = \lfloor |A|/2 \rfloor$. Hence, $|Z| \leq |Y| \leq |Z| + 1$ and $m = |Z|$.
2. Distinct variables in X cannot be mapped to the same variable in $Y \cup Z$.
3. Only the variables of the form $x_{\pi(i-1), \pi(i)}^{(i)}$ can be set to variables in $Y \cup Z$ by ρ , where π is a path chosen (at random) from $\{1, 2\}^d$. The rest are set to constants. Further, the variable $x_{\pi(i-1), \pi(i)}^{(i)}$ is set to a variable if and only if $a_i = 1$.

Note that b is distributed uniformly over $\{0, 1\}^d$. Given a polynomial $f \in \mathbb{F}[X]$, the restriction ρ yields a natural polynomial $f|_\rho \in \mathbb{F}[Y \cup Z]$ by substitution. Note, moreover, that if f is

multilinear then so is $f|_\rho$ since distinct variables in X cannot be mapped to the same variable in $Y \cup Z$ (Observation 22).

The following is the main technical lemma in the proof of the lower bound.

Lemma 23. *Let us assume that ρ is a random restriction sampled by Algorithm \mathcal{S} described above. Then we have the following:*

1. $\text{rank}(M_{(Y,Z)}(\text{IMM}_d|_\rho)) = 2^m$ with probability 1.
2. If $f \in \mathbb{F}[X]$ is any t -product polynomial, then for some absolute constant $\varepsilon > 0$,

$$\Pr_\rho[\text{rank}(M_{(Y,Z)}(f|_\rho)) \geq 2^{m-\varepsilon t}] \leq \frac{1}{2^{\Omega(t)}}.$$

3. If $f \in \mathbb{F}[X]$ is any r -simple polynomial, then for some absolute constant $\delta > 0$,

$$\Pr_\rho[\text{rank}(M_{(Y,Z)}(f|_\rho)) \geq 2^{m-\delta r}] \leq \frac{1}{2^{\Omega(r)}}.$$

Assuming 23 for now, we can finish the proof of Theorem 12 as follows.

Proof of Theorem 12 assuming Lemma 23. Assume that IMM_d is computed by a syntactic multilinear $(\Sigma\Pi)^\Delta\Sigma$ formula F of size at most s . By Lemma 18, we get that f can be expressed as a sum of at most $2s$ many polynomials, say f_1, f_2, \dots, f_s and g_1, g_2, \dots, g_s , where each summand f_i is a t -product polynomial and each summand g_j is a t -simple polynomial for $t = \Omega(\Delta d^{1/\Delta})$.

For each $i \in [s]$, Lemma 23 implies that for a random restriction ρ sampled by the algorithm \mathcal{S}

$$\Pr_\rho[\text{rank}(M_{(Y,Z)}(f_i|_\rho)) \geq 2^{m-\varepsilon t}] \leq \frac{1}{2^{\Omega(t)}} \quad \text{and} \quad \Pr_\rho[\text{rank}(M_{(Y,Z)}(g_i|_\rho)) \geq 2^{m-\delta t}] \leq \frac{1}{2^{\Omega(t)}},$$

where ε and δ are absolute constants.

Thus, unless $s \geq 2^{\Omega(t)}$, we see by a union bound that there exists a ρ that is sampled by \mathcal{S} with positive probability such that for each $i \in [s]$, $\text{rank}(M_{(Y,Z)}(f_i|_\rho)) \leq 2^{m-\varepsilon t}$ and $\text{rank}(M_{(Y,Z)}(g_i|_\rho)) \leq 2^{m-\delta t}$. For such a ρ , we have

$$\text{rank}(M_{(Y,Z)}(F|_\rho)) \leq 2^m \cdot \left(\frac{s}{2^{\varepsilon t}} + \frac{s}{2^{\delta t}} \right) < 2^m$$

unless $s \geq 2^{\Omega(t)}$.

From Lemma 23, we also know that for *any* restriction ρ sampled by \mathcal{S} with positive probability, we have $\text{rank}(M_{(Y,Z)}(\text{IMM}_d|_\rho)) = 2^m$. In particular, since F computes IMM_d , we must have $s \geq 2^{\Omega(t)} = 2^{\Omega(\Delta d^{1/\Delta})}$. \square

4.1 Proof of Lemma 23

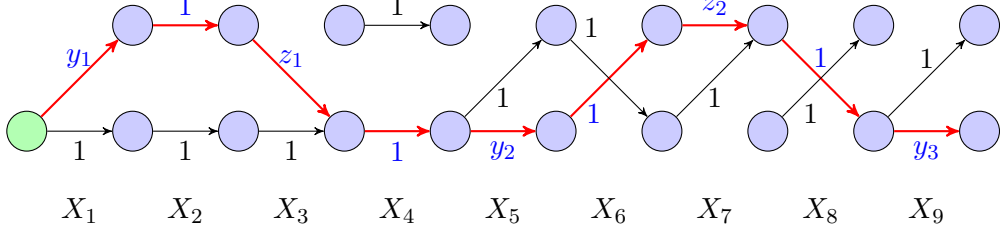


Figure 2: Effect of ρ on IMM_9 when the sampling algorithm \mathcal{S} yields $\pi = (2, 2, 1, 1, 1, 2, 2, 1, 1)$ and $a = (1, 0, 1, 0, 1, 0, 1, 0, 1)$. Thus, $\text{IMM}_9|_\rho$ in this case yields us $(1 + y_1 z_1)(1 + y_2 z_2)(1 + y_3)$.

Part 1: IMM_d has high rank

Let $\pi \in \{1, 2\}^d$ and $a \in \{0, 1\}^d$ be arbitrary. Note that in our sampling algorithm, ρ, A, b are completely determined given π and a .

Recall that we defined the matrix M to be equal to the product of matrices $M^{(1)}, \dots, M^{(d)}$, where the matrices $M^{(1)}, \dots, M^{(d)}$ are such that for each $i \in [d]$, the entries of $M^{(i)}$ are the variables of $X^{(i)}$ arranged in the obvious way. We defined $\text{IMM}_d(X) = M(1, 1) + M(1, 2)$. Let us use $\text{IMM}_d(X)_1$ to denote $M(1, 1)$ and $\text{IMM}_d(X)_2$ to denote $M(1, 2)$.

By observing the effect of ρ on IMM_d , we can claim the following.

Claim 24. *We have the following.*

- If $|A| = 2m$, then either $\text{IMM}_d(X)_{\pi(d)}|_\rho = \prod_{i=1}^m (1 + y_i z_i)$ and $\text{IMM}_d(X)_{\overline{\pi(d)}}|_\rho = 0$.
- If $|A| = 2m + 1$, then either $\text{IMM}_d(X)_{\pi(d)}|_\rho = \prod_{i=1}^m (1 + y_i z_i) y_{m+1}$ and $\text{IMM}_d(X)_{\overline{\pi(d)}}|_\rho = \prod_{i=1}^m (1 + y_i z_i)$.

Proof. We prove the claim by induction on d . When $d = 1$, $|A|$ is either 0 or 1, and the claim is easily verified. This is the base case of the induction.

Now consider the inductive case when $d > 1$. Let M' be the product of the first $d-1$ matrices after the restriction and let $f_1 = M'(1, 1)$ and $f_2 = M'(1, 2)$. Also, let $A' = \{i \in [d-1] \mid a_i = 1\}$. The induction hypothesis is applicable to f_1 and f_2 .

Consider $a_d \in \{0, 1\}$. If $a_d = 0$, then it follows from the sampling algorithm \mathcal{S} that $(\text{IMM}_d(X)_1|_\rho, \text{IMM}_d(X)_2|_\rho)$ is either (f_1, f_2) or (f_2, f_1) depending on whether $\pi(d)$ is equal to $\pi(d-1)$ or not. In either case, the claim follows from the inductive hypothesis.

Now consider the case when $a_d = 1$. Assume that $|A'| = 2m$. In this case, by the sampling algorithm, we get that $\text{IMM}_d(X)_{\pi(d)}|_\rho = f_{\pi(d-1)} \cdot y_{m+1}$ and $\text{IMM}_d(X)_{\overline{\pi(d)}}|_\rho = f_{\pi(d-1)}$. Using the inductive hypothesis, we are again done.

Finally, when $a_d = 1$ and $|A'| = 2m + 1$, we have $\text{IMM}_d(x)_{\pi(d)}|_\rho = f_{\pi(d-1)} \cdot z_{m+1} + f_{\overline{\pi(d-1)}}$ and $\text{IMM}_d(x)_{\overline{\pi(d)}}|_\rho = 0$ by definition. We can now complete the proof using the induction hypothesis. \square

Figure 2 illustrates how $\text{IMM}_d(X)$ behaves under the effect of the restriction chosen by the sampling algorithm \mathcal{S} .

Using the claim, we get the following.

$$\text{IMM}_d(X)|_\rho = \begin{cases} \prod_{i=1}^m (1 + y_i z_i) & \text{if } |A| = 2m, \\ (\prod_{i=1}^m (1 + y_i z_i)) \cdot (1 + y_{m+1}) & \text{if } |A| = 2m + 1, \end{cases}$$

where $m = \lfloor |Z| \rfloor$.

For any $S \subseteq [m]$, let Z_S (resp., Y_S) denote the monomial $\prod_{i \in S} z_i$ (resp., $\prod_{i \in S} y_i$). Now consider the partial derivative matrix $M_{(Y,Z)}(\text{IMM}_d|_\rho)$ which we will denote by \mathcal{M} . For the sake of simplicity let us assume that $|A| = 2m$. By the definition of \mathcal{M} , the rows and columns of \mathcal{M} are labelled by subsets of $[m]$ and the entry $\mathcal{M}(S, T)$ is the coefficient of $Y_S \cdot Z_T$ in $\text{IMM}_d|_\rho$. It is easy to see from the expression for $\text{IMM}_d|_\rho$ above that $\mathcal{M}(S, T) = 1$ if $S = T$ and 0 otherwise. That is, \mathcal{M} is the Identity matrix of size $2^m \times 2^m$ and hence it has rank 2^m . The proof in the case that $|A| = 2m + 1$ is similar. ¹² \square

Part 2: t -product polynomials have low rank

We now prove that for a t -product polynomial f , $\text{rank}(M_{(Y,Z)}(f|_\rho))$ is small with high probability.

Let f be a t -product polynomial, i.e. $f = f_1 f_2 \dots f_t$. Let $\chi : X \rightarrow [t]$ be a coloring such that $\chi^{-1}(i) = X_i$, where X_i is the variable set ascribed to f_i . That is, all the variables ascribed to f_i are assigned color i by χ . Recall from the definition of a t -product polynomial that each X_i is nonempty.

To prove part 2 of Lemma 23, we will first show that, with high probability over the choice of π , a constant fraction of the t colors appear along the path defined by π , i.e. along $(\pi(0), \pi(1)), (\pi(1), \pi(2)), \dots, (\pi(d-1), \pi(d))$. Given such a *multi-colored path*, we will then show that with high probability, over the choice of a , many of these colors have an *imbalance*. A color is said to have an imbalance under ρ if more variables from X of that color are mapped to the Y variables than the Z variables or vice versa. We will then appeal to arguments that are similar to those in [Raz06, RY09, DMPY12] to conclude that imbalance results in a low rank.

Variable coloring, t -product polynomials and imbalance. We start with some notation. Given a string $\pi \in \{1, 2\}^d$, let the path defined by π be the following sequence of pairs $(\pi(0), \pi(1)), (\pi(1), \pi(2)), \dots, (\pi(d-1), \pi(d))$ (we call it a path since these pairs correspond naturally to the edges of a path in the graph G_d defined in Section 2.1). We say that a color $\gamma \in [t]$ appears in layer $\ell \in [d]$ if there exist $u, v \in \{1, 2\}$ such that $\gamma = \chi(x_{u,v}^{(\ell)})$.

Let $C^0 = \emptyset$ and let $C^i = C^{i-1} \cup \{\chi(x_{u,v}^{(i)}) \mid u, v \in \{1, 2\}\}$ for $i \in [d]$, i.e., C^i is the set of colors appearing in layers $\{1, 2, \dots, i\}$. Therefore, $|C^d| = t$.

Let $C_\pi^0 = \emptyset$ and $C_\pi^i = C_\pi^{i-1} \cup \{\chi(x_{\pi(i-1), \pi(i)}^{(i)})\}$, i.e. C_π^i contains all the distinct colors appearing along the path defined by π up to layer i . We will show that $|C_\pi^d|$ is large with high probability. Formally,

Claim 25. *If $|C^d| = t$, then $\Pr_\pi[|C_\pi^d| \leq t/100] \leq 1/2^{\Omega(t)}$.*

We assume the claim for now and finish the proof of Part 2 of Lemma 23.

¹²In this case, the matrix \mathcal{M} has a $2^m \times 2^m$ sized Identity matrix as a submatrix.

We will say that π is good if $|C_\pi^d| > t/100$. Let $L = t/100$. The above claim shows that a random π is good with high probability. In what follows, we condition on picking a good π . Recall (Observation 22) that every variable x that does not label an edge along π is set to a constant by the restriction ρ . Moreover, the variable $x_{\pi(i-1),\pi(i)}^{(i)}$ is set to a variable in $\tilde{Y} \cup \tilde{Z}$ if and only if $a_i = 1$.

Let $\gamma \in [t]$ be any color. Let π_γ be the edges along the path π that are labelled by a variable of color γ , i.e. $\pi_\gamma = \{(\pi(i-1), \pi(i)) \mid \chi(x_{\pi(i-1),\pi(i)}^{(i)}) = \gamma\}$ (this set could be empty). Let $\rho(\pi_\gamma) = \{\rho(x_{\pi(i-1),\pi(i)}^{(i)}) \mid (\pi(i-1), \pi(i)) \in \pi_\gamma\} \cap (Y \cup Z)$. A color $\gamma \in [t]$ is said to have an imbalance w.r.t. ρ if $|\rho(\pi_\gamma) \cap Y| - |\rho(\pi_\gamma) \cap Z| \geq 1$.

It is easy to see that if $|\rho(\pi_\gamma)|$ is odd (or equivalently an odd number of variables in π_γ are set to variables by ρ), then γ must have an imbalance w.r.t. ρ . Note that the former event is equivalent to the event that $|\{i \in P_\gamma \mid a_i = 1\}|$ is odd, where $P_\gamma = \{i \mid (\pi(i-1), \pi(i)) \in \pi_\gamma\}$. Hence for any $\gamma \in C_\pi^d$ (i.e. γ such that $\pi_\gamma \neq \emptyset$),

$$\Pr_a[\gamma \text{ has an imbalance with respect to } \rho] = \Pr[\{i \in P_\gamma \mid a_i = 1\} \text{ is odd}] = 1/2,$$

where the last equality follows from Proposition 11. Further, since $|C_\pi^d| \geq L$ and the events corresponding to distinct $\gamma \in C_\pi^d$ are mutually independent, the Chernoff bound (Theorem 10) implies that

$$\Pr_a[\text{at most } L/4 \text{ colors have an imbalance with respect to } \rho] \leq 1/2^{\Omega(L)}.$$

The above probability is conditioned on π being good. Using Claim 25 and a union bound, we get that

$$\Pr_\rho[\text{at most } L/4 \text{ colors have an imbalance with respect to } \rho] \leq \frac{1}{2^{\Omega(L)}} + \frac{1}{2^{\Omega(t)}} \leq \frac{1}{2^{\Omega(t)}}. \quad (4)$$

We now show that this implies that $\text{rank}(M_{(Y,Z)}(f|\rho))$ is small.

Imbalance implies low rank. Let us recall that $f = f_1 f_2 \dots f_t$ is a t -product polynomial with X_i being the variable set ascribed to f_i ($i \in [t]$). The following lemma (see, e.g., [RY09]) will be useful in bounding $\text{rank}(M_{(Y,Z)}(f|\rho))$.

Lemma 26 ([RY09], Proposition 2.5). *Let $g = g_1 g_2 \dots g_t$ be a t -product polynomial over the set of variables $Y \cup Z$ where $\text{Vars}(g_i) = Y_i \cup Z_i$. Then $\text{rank}(M_{(Y,Z)}(g)) = \prod_{i \in [t]} \text{rank}(M_{(Y_i, Z_i)}(g_i))$.*

From Lemma 26, we get that $\text{rank}(M_{(Y,Z)}(f|\rho)) = \prod_{i=1}^t \text{rank}(M_{(Y_i, Z_i)}(f_i|\rho))$ where $Y_i = Y \cap \{\rho(x) \mid x \in X_i\}$ and $Z_i = Z \cap \{\rho(x) \mid x \in X_i\}$. For all $i \in [t]$, from the definition it is clear that the rank of the matrix $M_{(Y_i, Z_i)}(f_i|\rho)$ is upper bounded by $2^{\min\{|Y_i|, |Z_i|\}} \leq 2^{(|Y_i|+|Z_i|)/2}$.

Further, note that if color i has imbalance w.r.t. ρ , then $||Y_i| - |Z_i|| \geq 1$ and hence we have

$$\text{rank}(M_{(Y_i, Z_i)}(f_i|\rho)) \leq 2^{\min\{|Y_i|, |Z_i|\}} \leq 2^{(|Y_i|+|Z_i|-1)/2}.$$

Thus, $\text{rank}(M_{(Y,Z)}(f|\rho)) \leq \prod_{i=1}^t 2^{(|Y_i|+|Z_i|-1)/2} = 2^{((|Y|+|Z|)/2) - (\ell/2)} \leq 2^{m - (\ell-1)/2}$ where ℓ is the number of colors that have imbalance w.r.t. ρ . From the above discussion and (4), we can infer that

$$\Pr_{\pi}[\text{rank}(M_{Y,Z}(f|\rho)) \geq 2^{m-(t/1000)}] \leq \Pr_{\pi}[\ell \leq t/400] \leq \frac{1}{2^{\Omega(t)}}.$$

Assuming Claim 25 we are now done with Part 2 of Lemma 23. We now present the proof of Claim 25.

Proof of Claim 25. Define O^{2i+1} to be all the colors appearing in odd numbered layers up to $2i+1$, i.e. $O^{2i+1} = O^{2i-1} \cup \{\chi(x_{u,v}^{(2i+1)}) \mid u, v \in \{1, 2\}\}$, with the assumption that $O^{-1} = \emptyset$. Similarly, we define $E^{2i} = E^{2i-2} \cup \{\chi(x_{u,v}^{(2i)}) \mid u, v \in \{1, 2\}\}$ with $E^0 = \emptyset$.

We also define O_{π}^{2i+1} to be all the colors appearing in odd numbered layers along π up to the layer $2i+1$, i.e. $O_{\pi}^{2i+1} = O_{\pi}^{2i-1} \cup \{\chi(x_{\pi(2i), \pi(2i+1)}^{(2i+1)})\}$. Similarly, we define $E_{\pi}^{2i} = E_{\pi}^{2i-2} \cup \{\chi(x_{\pi(2i-1), \pi(2i)}^{(2i)})\}$.

We know that $|C^d| = t$. Therefore, either $|O^d| \geq t/2$ or $|E^d| \geq t/2$. Let us assume without loss of generality that $|O^d| \geq t/2$.

Let $j_1 < j_2 < \dots < j_{\tau}$ be those odd indices such that for each $1 \leq i \leq \tau$, $|O^{j_i-2}| < |O^{j_i}|$, i.e. each O^{j_i} has at least one color that does not appear in odd layers $j < j_i$. Let $\gamma_1, \gamma_2, \dots, \gamma_{\tau}$ be new colors which appear in each of these sets. (If multiple new colors appear in a set then choose any one.) Since we can accumulate at most 4 new colors in any layer and the total number of colors appearing in all the odd layers is at least $t/2$, τ is at least $t/8$.

For $i \in \{2, \dots, \tau\}$, let W_i be the indicator random variable which takes value 1 if the color γ_i appears in the set $O_{\pi}^{j_i}$ and 0 otherwise. We claim that $\mathbf{E}[W_i] \geq 1/4$. This is because the probability that the color γ_i appears in $O_{\pi}^{j_i}$ is at least the probability that some fixed variable $x_{u,v}^{(j_i)} \in X^{(j_i)}$ with color γ_i appears along the path π . This latter event happens exactly when $\pi(j_i-1) = u$ and $\pi(j_i) = v$, which happens with probability $1/4$ (as π is chosen uniformly from $\{1, 2\}^d$ and $j_i > 1$ for each $i \in \{2, \dots, \tau\}$).

Note also that the random variables W_2, \dots, W_{τ} are independently distributed since they depend on distinct co-ordinates of π (W_i depends only on $\pi(j_i-1)$ and $\pi(j_i)$).

Now $\mathbf{E}[\sum_{2 \leq i \leq \tau} W_i] = \sum_{2 \leq i \leq \tau} \mathbf{E}[W_i] \geq \sum_{2 \leq i \leq \tau} 1/4 \geq (t/32) - 1$, and thus we get

$$\Pr_{\pi}[|C_{\pi}^d| \leq t/100] \leq \Pr_{\pi}[|O_{\pi}^d| \leq t/100] \leq \Pr_{\pi}[\sum_i W_i \leq t/100] \leq 1/2^{\Omega(t)}$$

where the first inequality uses the fact that $|O_{\pi}^d| \leq |C_{\pi}^d|$, the second uses the fact that $\sum_i W_i \leq |O_{\pi}^d|$, and the last inequality follows by the Chernoff bound (Theorem 10). \square

Part 3: r -simple polynomials have low rank.

Here we prove that if $f \in \mathbb{F}[X]$ is any r -simple polynomial, then for some absolute constant $\delta > 0$,

$$\Pr[\text{rank}(M_{(Y,Z)}(f|\rho)) \geq 2^{m-\delta r}] \leq \frac{1}{2^{\Omega(r)}}.$$

As f is an r -simple polynomial, we know that $f = \left(\prod_{i=1}^{r'} L_i\right) \cdot G$, where $r' \leq r$, the L_i ($i \in [r']$) are linear polynomials. For each $i \in [r']$, let X_i be the set of variables ascribed to L_i and $X_{r'+1}$ be the set of variables ascribed to G . We also have $|\bigcup_{i=1}^{r'} X_i| \geq 400r$.

Fix a restriction ρ . We have $f|_\rho = \left(\prod_{i=1}^{r'} L_i|_\rho\right) \cdot G|_\rho$. Let $Y_i = \{\rho(x) \mid x \in X_i\} \cap Y$ and $Z_i = \{\rho(x) \mid x \in X_i\} \cap Z$ for each $i \in [r']$. Let $Y' = \cup_{i=1}^{r'} Y_i$ and $Z' = \cup_{i=1}^{r'} Z_i$. Also, let $Y'' = Y \setminus Y'$ and $Z'' = Z \setminus Z'$. Let U denote $\cup_{i=1}^{r'} X_i$ and let $U|_\rho = (\cup_{i=1}^{r'} Y_i) \cup (\cup_{i=1}^{r'} Z_i)$.

In the following claim we show that if U is a large set to begin with then with high probability (over the restriction ρ defined by the sampling algorithm), $U|_\rho$ is also large.

Claim 27. *If $|U| \geq 400r$, then $\Pr[|U|_\rho \leq 4r] \leq 1/2^{\Omega(r)}$.*

We first finish the proof of Part 3 of Lemma 23 assuming this claim.

We say that a restriction ρ is good if we get $|U|_\rho \geq 4r$. In what follows we will condition on the event that we have a good ρ .

For a restriction ρ , for each $i \in [r']$, we can write $L_i|_\rho(Y_i, Z_i)$ as $L'_i|_\rho(Y_i) + L''_i|_\rho(Z_i)$ as L_i s are linear polynomials. Therefore we get

$$\prod_{i=1}^{r'} L_i|_\rho(Y', Z') = \sum_{S \subseteq [r']} \prod_{i \in S} L'_i|_\rho(Y_i) \cdot \prod_{j \in [r'] \setminus S} L''_j|_\rho(Z_j).$$

Let L_S denote the polynomial $\prod_{i \in S} L'_i|_\rho(Y_i) \cdot \prod_{j \in [r'] \setminus S} L''_j|_\rho(Z_j)$. Note that for all $S \subseteq [r']$, $\text{rank}(M_{(Y', Z')}(L_S))$ is at most 1. Therefore, by the subadditivity of matrix rank, we get that $\text{rank}\left(M_{(Y', Z')}\left(\prod_{i=1}^{r'} L_i|_\rho(Y', Z')\right)\right) \leq 2^{r'} \leq 2^r$. We can now bound $\text{rank}(M_{(Y, Z)}(f|_\rho))$.

$$\begin{aligned} \frac{\text{rank}(M_{(Y, Z)}(f|_\rho))}{2^{(|Y|+|Z|)/2}} &= \frac{\text{rank}\left(M_{(Y, Z)}\left(\prod_{i=1}^{r'} L_i|_\rho \cdot G|_\rho\right)\right)}{2^{(|Y|+|Z|)/2}} \\ &= \frac{\text{rank}\left(M_{(Y', Z')}\left(\prod_{i=1}^{r'} L_i|_\rho\right)\right)}{2^{(|Y'|+|Z'|)/2}} \cdot \frac{\text{rank}(M_{(Y'', Z'')}(G|_\rho))}{2^{(|Y''|+|Z''|)/2}} \\ &\leq \frac{2^r}{2^{|U|_\rho/2}} \cdot 1 \leq \frac{2^r}{2^{2r}} = \frac{1}{2^r}. \end{aligned}$$

where the second equality follows from Lemma 26 and the first inequality uses the fact that $\text{rank}(M_{(Y'', Z'')}(G|_\rho)) \leq \min\{2^{|Y''|}, 2^{|Z''|}\} \leq 2^{(|Y''|+|Z''|)/2}$.

Therefore, we have $\text{rank}(M_{(Y, Z)}(f|_\rho)) \leq 2^{(|Y|+|Z|)/2} / 2^r \leq 2^{m+(1/2)-r}$ for any good ρ . As Claim 27 tells us that ρ is good with probability $1 - 1/2^{\Omega(r)}$, we are done.

Proof of Claim 27. We say that a layer $i \in [d]$ is *touched* by U if there is a variable $x_{u,v}^{(i)} \in U$. We call such an $x_{u,v}^{(i)}$ a *contact edge*. Any layer touched by U has at most 4 contact edges. As $|U| \geq 400r$, U touches at least $100r$ layers. Let us assume without loss of generality that at least half of them are odd numbered. Let these be $\ell_1 < \ell_2 < \dots < \ell_R$, where $R \geq 50r$. Let us fix a contact edge (u_i, v_i) per ℓ_i for each $i \in [R]$. Let us denote these edges by $x_{u_i, v_i}^{(\ell_i)}$ for $i \in [R]$.

Note that $\ell_i > 1$ for all $i \in \{2, \dots, R\}$. For each $i \in \{2, \dots, R\}$, we define the indicator random variable W_i so that it is 1 if $\rho(x_{u_i, v_i}^{(\ell_i)}) \in Y \cup Z$ and 0 otherwise. Note that $|U|_\rho \geq \sum_{i=2}^R W_i$.

We claim $\Pr_\rho[W_i = 1] = 1/8$, where ρ is sampled as per the sampling algorithm \mathcal{S} . This is because $\rho(x_{u_i, v_i}^{(\ell_i)}) \in Y \cup Z$ if and only if the edge labelled by $x_{u_i, v_i}^{(\ell_i)}$ lies along π and $a_{\ell_i} = 1$. The probability that the edge labelled by $x_{u,v}^{(\ell)}$ (for $\ell > 1$) lies along π is exactly $1/4$ and for any layer ℓ the probability that $a_\ell = 1$ is exactly $1/2$.

Therefore, we get $\mathbf{E}[\sum_{i=2}^R W_i] = (R - 1)/8 \geq 5r$. Note that the random variables W_i are mutually independent, since W_i depends only on $\pi(\ell_i - 1), \pi(\ell_i)$ and a_{ℓ_i} which are mutually independent across various $i \in [R]$.

Hence we get that $\Pr[|U|_{\rho} \leq 4r] \leq \Pr[\sum_{i=1}^R W_i \leq 4r] \leq \frac{1}{2^{\Omega(r)}}$, where the last inequality uses the Chernoff bound (Theorem 10). \square

Acknowledgements We thank the organizers of the NMI workshop on Arithmetic Circuit complexity at IMSc in 2017, where this collaboration began. We thank Mrinal Kumar, Rafael Oliveira and Benjamin Rossman for communicating to us that Corollary 16 follows from our main theorem. We thank the anonymous referees of STACS 2018 and SICOMP for their comments and suggestions.

References

- [AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3):14:1–14:36, 2010. URL: <http://doi.acm.org/10.1145/1706591.1706594>, doi:10.1145/1706591.1706594.
- [AKV17] Noga Alon, Mrinal Kumar, and Ben Lee Volk. Unbalancing sets and an almost quadratic lower bound for syntactically multilinear arithmetic circuits. 2017. URL: <https://mrinalkr.bitbucket.io/papers/synt-multilinear.pdf>.
- [AV08] Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *proceedings of Foundations of Computer Science (FOCS)*, pages 67–75, 2008. doi:10.1109/FOCS.2008.32.
- [Bre74] Richard P. Brent. The parallel evaluation of general arithmetic expressions. *Journal of the ACM*, 21(2):201–206, April 1974. doi:10.1145/321812.321815.
- [Che52] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952. URL: <http://www.jstor.org/stable/2236576>.
- [DMPY12] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In *proceedings of Symposium on Theory of Computing (STOC)*, pages 615–624, 2012. URL: <http://doi.acm.org/10.1145/2213977.2214034>, doi:10.1145/2213977.2214034.
- [DP09] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009. URL: <http://www.cambridge.org/gb/knowledge/isbn/item2327542/>.
- [FLMS14] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In *proceedings of Symposium on Theory of Computing (STOC)*, pages 128–135, 2014. URL: <http://doi.acm.org/10.1145/2591796.2591824>.
- [GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM Journal of Computing*, 45(3):1064–1079, 2016. URL: <https://doi.org/10.1137/140957123>, doi:10.1137/140957123.

- [Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. URL: <http://www.jstor.org/stable/2282952>.
- [HY11] Pavel Hrubeš and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Computational Complexity*, 20(3):559–578, 2011.
- [KLSS14] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An Exponential Lower Bound for Homogeneous Depth Four Arithmetic Circuits. In *proceedings of Foundations of Computer Science (FOCS)*, 2014. doi:10.1109/FOCS.2014.15.
- [KNS16] Neeraj Kayal, Vineet Nair, and Chandan Saha. Separation between read-once oblivious algebraic branching programs (ROABPs) and multilinear depth three circuits. In *proceedings of Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 46:1–46:15, 2016. URL: <https://doi.org/10.4230/LIPIcs.STACS.2016.46>, doi:10.4230/LIPIcs.STACS.2016.46.
- [KS14] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *proceedings of Foundations of Computer Science (FOCS)*, 2014. doi:10.1109/FOCS.2014.46.
- [KST16] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. On the size of homogeneous and of depth four formulas with low individual degree. In *proceedings of Symposium on Theory of Computing, STOC*, pages 626–632, 2016. URL: <http://doi.acm.org/10.1145/2897518.2897550>, doi:10.1145/2897518.2897550.
- [NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. doi:10.1007/BF01294256.
- [Raz04] Ran Raz. Multilinear-NC² ≠ multilinear-NC¹. In *proceedings of Foundations of Computer Science (FOCS)*, pages 344–351, 2004. URL: <https://doi.org/10.1109/FOCS.2004.42>, doi:10.1109/FOCS.2004.42.
- [Raz06] Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing*, 2(1):121–135, 2006. doi:10.4086/toc.2006.v002a006.
- [Raz13] Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *Journal of the ACM*, 60(6):40:1–40:15, 2013. URL: <http://doi.acm.org/10.1145/2535928>, doi:10.1145/2535928.
- [Ros15] Benjamin Rossman. The average sensitivity of bounded-depth formulas. In *proceedings of Foundations of Computer Science (FOCS)*, pages 424–430, 2015. URL: <https://doi.org/10.1109/FOCS.2015.33>, doi:10.1109/FOCS.2015.33.
- [RS17] Benjamin Rossman and Srikanth Srinivasan. Separation of AC⁰[⊕] formulas and circuits. In *proceedings of International Colloquium on Automata, Languages, and Programming, (ICALP)*, pages 50:1–50:13, 2017. URL: <https://doi.org/10.4230/LIPIcs.ICALP.2017.50>, doi:10.4230/LIPIcs.ICALP.2017.50.
- [RSY08] Ran Raz, Amir Shpilka, and Amir Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM Journal of Computing*, 38(4):1624–1647, 2008. doi:10.1137/070707932.

- [RY08] Ran Raz and Amir Yehudayoff. Balancing syntactically multilinear arithmetic circuits. *Computational Complexity*, 17(4):515–535, 2008. doi:10.1007/s00037-008-0254-0.
- [RY09] Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009. doi:10.1007/s00037-009-0270-8.
- [Sap15] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2015. URL: <https://github.com/dasarpmar/lowerbounds-survey/releases/>.
- [Spi73] Philip M Spira. Computation times of arithmetic and boolean functions in (d, r) circuits. *IEEE Transactions on Computers*, 100(6):552–555, 1973.
- [SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010. URL: <http://dx.doi.org/10.1561/04000000039>.
- [Tav15] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Information and Computation*, 240:2–11, 2015. doi:10.1016/j.ic.2014.09.004.
- [VSB83] Leslie G. Valiant, Sven Skyum, Stuart J. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM Journal of Computing*, 12(4):641–644, 1983. doi:10.1137/0212043.