A Near-Optimal Depth-Hierarchy Theorem for Small-Depth Multilinear Circuits

Suryajith Chillara[*]   Christian Engels[†]   Nutan Limaye[‡]   Srikanth Srinivasan[§]

April 7, 2018

Abstract

We study the size blow-up that is necessary to convert an algebraic circuit of product-depth $\Delta + 1$ to one of product-depth Δ in the multilinear setting.

We show that for every positive $\Delta = \Delta(n) = o(\log n / \log \log n)$, there is an explicit multilinear polynomial $P^{(\Delta)}$ on n variables that can be computed by a multilinear formula of product-depth $\Delta + 1$ and size $O(n)$, but not by any multilinear circuit of product-depth Δ and size less than $\exp(n^{\Omega(1/\Delta)})$. This result is tight up to the constant implicit in the double exponent for all $\Delta = o(\log n / \log \log n)$.

This strengthens a result of Raz and Yehudayoff (Computational Complexity 2009) who prove a quasipolynomial separation for constant-depth multilinear circuits, and a result of Kayal, Nair and Saha (STACS 2016) who give an exponential separation in the case $\Delta = 1$.

Our separating examples may be viewed as algebraic analogues of variants of the Graph Reachability problem studied by Chen, Oliveira, Servedio and Tan (STOC 2016), who used them to prove lower bounds for constant-depth *Boolean* circuits.

1 Introduction

This paper deals with a question in the area of *Algebraic Complexity,* which studies the Computational Complexity of any algorithmic task that can be cast as the problem of computing a fixed multivariate polynomial (or polynomials) $f \in \mathbb{F}[x_1, \ldots, x_N]$ on a given $a \in \mathbb{F}^N$. Many fundamental problems such as the Determinant, Permanent, the Fast Fourier Transform and Matrix Multiplication can be captured in this general paradigm. The natural computational model for solving such problems are Algebraic Circuits (and their close relations Algebraic Formulas and Algebraic Branching Programs) which use the algebraic operations of the polynomial ring $\mathbb{F}[x_1, \ldots, x_N]$ to compute the given polynomial.

Our main focus in this paper is on *Small-depth* Algebraic circuits, which are easily defined as follows.[1] Recall that any multivariate polynomial $f \in \mathbb{F}[x_1, \ldots, x_N]$ can be written as a linear combination of terms which are products of variables (i.e. monomials); we call such an expression a $\Sigma\Pi$ circuit for f. In general, such an expression for f could be prohibitively large. More compact representations for f may be obtained if we consider representations as linear combinations of products of linear functions, which we call $\Sigma\Pi\Sigma$ circuits, and subsequent generalizations such as $\Sigma\Pi\Sigma\Pi$ circuits and so on. We consider representations of the form

---

[*]Department of CSE, IIT Bombay. `suryajith@cse.iitb.ac.in`
[†]Department of CSE, IIT Bombay. `christian@cse.iitb.ac.in`
[‡]Department of CSE, IIT Bombay. `nutan@cse.iitb.ac.in`
[§]Department of Mathematics, IIT Bombay. `srikanth@math.iitb.ac.in`
[1]We actually define algebraic *formulas* here, but the distinction is not too important for our results in this paper.

$\Sigma\Pi\cdots O_d$ for some $d = d(N)$ where d is a slow growing function of the number of variables N. We consider such a representation of a polynomial f as a computation of f.

The efficiency of such a computation is captured by the following two complexity measures: the *size* of the corresponding expression, which captures the number of operations used in computing the polynomial; and the *product-depth* of the circuit, which is the number of Πs in the expression for the circuit class (e.g. 1 for $\Sigma\Pi\Sigma$ circuits, 2 for $\Sigma\Pi\Sigma\Pi$ circuits etc.) and which measures in some sense the inductive complexity of the corresponding computation.[2]

This paper is motivated by questions in a general body of results in the area that go by the name of *Depth-reduction.* Informally, the question is: what is the worst case blow-up in size required to convert a circuit of product-depth Δ to one of product-depth $\Delta' < \Delta$? This is an important question, since circuits of smaller depth are frequently easier to analyze and understand, and such results allow us to transfer this understanding to more complex (i.e. higher depth) classes of circuits. Consequently, there have been many results addressing this general question for various models of Algebraic (and also Boolean) computation including Algebraic and Boolean formulas [Bre74, Spi73], Boolean circuits [HPV77], and Algebraic circuits [VSBR83, AV08, Koi12, Tav15, GKKS16]. Recently, Tavenas [Tav15] and Gupta, Kamath, Kayal and Saptharishi [GKKS16] (building on [VSBR83, AV08, Koi12]) showed that a strong enough exponential lower bound for $\Sigma\Pi\Sigma$ circuits would imply a superpolynomial lower bound for general algebraic circuits. Interesting impossibility results are also known in this direction: for instance, it is known that the result of Tavenas [Tav15], which converts a general circuit to a $\Sigma\Pi\Sigma\Pi$ circuit of subexponential size, cannot be improved in the restricted model of *homogeneous*[3] $\Sigma\Pi\Sigma\Pi$ circuits [GKKS14, FLMS14, KLSS14, KS14].

Here, we study a more fine-grained version of the question of depth-reduction. We ask: what is the size blow-up in converting a circuit of product-depth $\Delta + 1$ to one of product-depth Δ? A natural strategy to carry out such a depth-reduction for a given $(\Sigma\Pi)^{\Delta+1}\Sigma$[4] expression F is to take some product terms in the expression and interchange them with the inner sum terms via the distributive law. This creates a blow-up in the size of the expression that is exponential in the number of sum terms. It is not hard to show that by choosing the sum terms carefully, one can limit this blow-up to $\exp(s^{1/\Delta+o(1)})$ for constant Δ,[5] where s is the size of the expression of depth $\Delta + 1$.

Is this exponential[6] blow-up unavoidable for any Δ? The evidence we do have seems to suggest the answer is yes. For example, in the Boolean setting (where the Σ and Π are replaced by their Boolean counterparts $\bigvee$ and $\bigwedge$), such an exponential *Depth-hierarchy theorem* is a classical result of Håstad [Hås87], with recent improvements by Rossman, Servedio, Tan and Håstad [RST15, Hås16, HRST17]. Even in the algebraic setting, there have been partial results in this direction [NW97, RY09, KNS16, KS16]. For *homogeneous* circuits such a result was proved for $\Delta = 1$ in the work of Nisan and Wigderson [NW97].

We study this question in the *multilinear* setting, where the circuits are restricted to computing *multilinear* polynomials at each stage of computation (multilinear polynomials are polynomials where the degree of each variable is at most 1). This is a fairly natural model of

---

[2]It is also standard in the literature to consider the *depth* of the circuit, which is the number of Σ and Π terms in the defining expression for the circuit class, but the product-depth is frequently nicer (invariant under simple operations such as linear transformations etc.) and is essentially $\lfloor\text{depth}/2\rfloor$. So we mostly use product-depth. We also state our results in terms of depth.

[3]I.e. a circuit where every intermediate expression computes a homogeneous polynomial.

[4]I.e. a $\Sigma\Pi\cdots\Sigma$ expression with exactly $(\Delta + 1)$ many Πs.

[5]In fact, the upper bound is of the form $\exp(s^{1/\Delta}\log s)$, and is the $\exp(s^{1/\Delta+o(1)})$ as long as $\Delta = o(\log s/\log\log s)$.

[6]Strictly speaking, we get an exponential blow-up only for *constant* Δ. The careful reader should read "exponential" as "exponential in $s^{1/\Delta}$" for general Δ.

computation for multilinear polynomials, and has been extensively studied in the literature [Raz06, Raz04, RY08, RY09, RSY08, DMPY12, KNS16].

Raz and Yehudayoff [RY09] considered the problem of separating product-depth $\Delta + 1$ circuits from product-depth Δ circuits in the multilinear setting and proved a *superpolynomial* separation implying a superpolynomial depth-hierarchy theorem. More precisely, they showed that there are circuits of size s and product-depth $\Delta + 1$ such that any product-depth Δ circuit computing the same polynomial must have size at least $s^{(\log s)^{\Omega(1/\Delta)}}$. While this result shows that some blow-up is unavoidable in the multilinear setting, this is still only a quasipolynomial separation and does not completely resolve our original question. More recently, Kayal, Nair and Saha [KNS16] resolved the question completely in the case when $\Delta = 1$ by giving an optimal exponential separation between product-depth 2 and product-depth 1 multilinear circuits.

We extend both these results to prove a strong depth-hierarchy theorem for all small depths. The following is implied by Corollary 14.

Theorem 1. *For each $\Delta = \Delta(n) = o(\log n / \log \log n)$, there is an explicit multilinear polynomial $P^{(\Delta)}$ on n variables that can be computed by a multilinear formula of product-depth $(\Delta + 1)$ and linear size, but not by any multilinear circuit of product-depth Δ and size less than $\exp(n^{\Omega(1/\Delta)})$.*

We also prove an analogous result for depth instead of product-depth (Corollary 15). Note that, from the above discussion, the above results are tight up to the constant implicit in the $\Omega(1/\Delta)$.

1.1 Related Work

We survey here some of the work on depth-hierarchy theorems in the Algebraic and Boolean settings.

As mentioned above, this is a well-known question in the Boolean setting, with a near-optimal separation between different depths due to Håstad [Hås87] (building on [Ajt83, FSS84, Yao85]); the separating examples in Håstad's result are the *Sipser functions*, which are computed by linear-sized Boolean circuits of depth $\Delta + 1$ but cannot be computed by subexponential-sized depth Δ circuits. A recent variant of this lower bound was proved by Chen et al. [COST16] for *Skew-Sipser functions*, which in turn was used to prove near-optimal lower bounds for constant-depth Boolean circuits solving variants of the Boolean Graph Reachability problem. Their construction motivates our hard polynomials, as we describe below.

In the algebraic setting, we have separations between fixed constant-depth circuits under the restriction of homogeneity. Nisan and Wigderson [NW97] show that converting a homogeneous $\Sigma\Pi\Sigma\Pi$ circuit to a homogeneous $\Sigma\Pi\Sigma$ circuit requires an exponential blow-up. A quasipolynomial separation between homogeneous $\Sigma\Pi\Sigma\Pi\Sigma$ and $\Sigma\Pi\Sigma\Pi$ circuits was shown by Kumar and Saptharishi [KS16]. However, as far as we know, nothing is known for larger depths.

When the algebraic circuits are instead restricted to be multilinear, more is known. Raz and Yehudayoff [RY09] were the first to study this question, and showed a quasipolynomial depth-hierarchy theorem for all constant product-depths. In the case of product-depth 1 vs. product-depth 2, this was strengthened to an exponential separation by Kayal, Nair and Saha [KNS16].

One can ask if the methods of [RY09, KNS16] can be used to prove our result. Kayal et al.[KNS16] prove their result by defining a suitable complexity measure for any polynomial $f \in \mathbb{F}[x_1, \ldots, x_N]$, and show that this measure is small for any subexponential-sized $\Sigma\Pi\Sigma$ circuit but large for some linear-sized $\Sigma\Pi\Sigma\Pi$ circuit. In particular, this means that this measure needs to be changed to prove lower bounds for larger depth circuits. However, it is not clear how to modify this measure to take into account the product-depth of the circuit class.

This is not a problem with the technique of Raz and Yehudayoff [RY09], which can indeed be used to prove exponential lower bounds on the sizes of small-depth circuits for computing certain polynomials. However, the polynomials used to witness the superpolynomial separation cannot give an exponential depth-hierarchy, for the reasons we now explain.

The depth-hierarchy theorem of [RY09] is obtained via an exponential lower bound $s(n, \Delta) \approx \exp(n^{1/\Delta})$ against product-depth Δ circuits computing polynomials from a certain "hard" class $\mathcal{P}$ of polynomials on n variables. Importantly, these lower bounds are tight in the sense that there also exist product-depth Δ circuits of size roughly $s(n, \Delta)$ computing polynomials from $\mathcal{P}$. Since $s(n, \Delta - 1)$ is superpolynomially larger than $s(n, \Delta)$, we obtain a superpolynomial separation between circuits of product-depth Δ and product-depth $\Delta - 1$. However, since we cannot improve on either the upper bound of $s(n, \Delta)$ or the lower bound of $s(n, \Delta - 1)$ for this class of polynomials, we cannot hope to improve this separation.

There is a striking parallel between this line of work and the setting of Boolean circuits, where also a similar quasipolynomial depth-hierarchy theorem can be obtained by appealing to easier lower bounds for explicit functions such as the "Parity" function [Hås87]. It is known that the depth-Δ Boolean complexity of the Parity function is $\exp(\Theta(n^{1/\Delta - 1}))$ and this yields a quasipolynomial depth-hierarchy theorem for constant-depth Boolean circuits as in the work of Raz and Yehudayoff described above. However, Håstad [Hås87] was able to improve this to an exponential depth-hierarchy theorem by changing the candidate hard function to the Sipser functions and then proving a lower bound for these functions by a related, but more involved, technique [Hås87, COST16].

1.2 Proof Outline

As mentioned in the Related Work section above, the polynomials considered by Raz and Yehudyaoff [RY09] cannot be used to prove better than a quasipolynomial depth-hierarchy theorem. This parallels a quasipolynomial depth-hierarchy theorem in the Boolean setting. However, in the Boolean setting, there is a different family of explicit functions that can be used to prove an exponential depth-hierarchy theorem.

Our aim is to do something similar in the setting of multilinear small-depth circuits. While the methods for proving Boolean circuit lower bounds do not seem to apply in the algebraic setting, we can take inspiration in the matter of choosing the candidate hard polynomial. For this, we look to the recent result of Chen et al. [COST16], who observe that the Sipser functions (and also their "skew" variants) can be interpreted as special cases of the Boolean Graph Reachability problem. This is quite appealing for us, since Graph Reachability has a natural polynomial analogue, the Iterated Matrix Multiplication polynomial, which has been a source of many lower bounds in algebraic circuit complexity [NW97, FLMS14, KS14, BC15, KNS16, KST16, CLS18]. We therefore choose our lower bound candidate to be a restriction of the Iterated Matrix Multiplication polynomial, which we now describe.

1.2.1 The Hard Polynomials

All the polynomials we consider will be naturally defined in terms of Directed Acyclic graphs (DAGs) with a unique source and sink. Given such a graph G with source s and sink t, we define a corresponding polynomial P_G as follows. Label each edge e of G with a distinct variable x_e. The polynomial P_G is defined to be the sum, over all paths π from s to t, of the monomial which is the product of edge labels along that path. (See Figure 1 for a simple example.) This polynomial P_G is the algebraic analogue of the Boolean Computational problem of checking s-t reachability on subgraphs of G. (Informally, we think of each x_e as a Boolean variable that
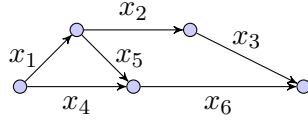
Figure 1: A simple example of a graph G (all edges go from left to right). In this case, P_G is $x_1x_2x_3 + x_1x_5x_6 + x_4x_6$.

determines if e remains in the subgraph or not. Then the polynomial P_G on the Boolean input corresponding to a subgraph H of G counts the number of s-t paths in H.)

When G is a layered graph with d layers and all possible edges between consecutive layers, P_G is known as the Iterated Matrix Multiplication polynomial (for the connection to matrix product, see, e.g., [FLMS14]). This polynomial has been studied in many contexts in Algebraic circuit complexity. Unfortunately, it is not useful in our setting, since it does not have a subexponential-size constant-depth multilinear circuit, as was shown recently by some of the authors [CLS18]. In particular, this means that it cannot be used to obtain the claimed separation between depths $\Delta + 1$ and Δ.

However, changing G can drastically reduce the complexity of the polynomial P_G. To obtain a G such that P_G has an efficient depth $\Delta + 1$ circuit, we use *series-parallel* graphs, as in the result of Chen et al. [COST16].

Recall that given DAGs $G_1, \ldots, G_k$ with sources $s_1, \ldots, s_k$ and sinks $t_1, \ldots, t_k$, we can construct larger DAGs by composing these graphs in parallel (by identifying all the sources and all the sinks) or in series (by identifying t_1 with s_2, t_2 with s_3 and so on until t_{k-1} with s_k) to get larger DAGs G_{par} or G_{ser} respectively. Note that the corresponding polynomials P_{par} and P_{ser} are the sums and products of the polynomials $P_1, \ldots, P_k$, respectively. In particular, if $P_1, \ldots, P_k$ have efficient circuits of product-depth at most Δ, then P_{par} (resp. P_{ser}) has an efficient circuit of product-depth at most Δ (resp. $\Delta + 1$).

In this way, we can inductively construct polynomials which, by their very definition, have efficient circuits of small depth. In particular, to keep the product-depth of the circuit bounded by $\Delta + 1$, it is sufficient to ensure that the number of series compositions used in the construction of the graph is at most $\Delta + 1$.

1.2.2 The $\Sigma\Pi\Sigma$ lower bound

We motivate our proof with the solution to the simpler problem of separating product-depth 2 and product-depth 1 circuits. In fact, we will separate the power of $\Pi\Sigma\Pi$ and $\Sigma\Pi\Sigma$ circuits. While an exponential separation was already known in this case thanks to the work of Kayal et al. [KNS16], we will outline a different proof that extends to larger depths.

Given the above discussion, a natural polynomial to witness the separation between product-depth 1 and product-depth 2 circuits is a polynomial corresponding to a series parallel graph obtained with exactly 2 series compositions. Unfortunately, our proof technique is not able to prove a lower bound for such a graph. However, we are able to prove such a separation with the slightly more complicated graph $G^{(1)}$ in Figure 3, which is made of a composition of m copies of the basic graph $H^{(1)}$ (see Figure 2). Though the graph $G^{(1)}$ is constructed using three series compositions rather than two, the corresponding polynomial $P^{(1)}(X)$ nevertheless has a product-depth 2 circuit of small size (this is because the polynomial corresponding to $H^{(1)}$ depends only on a constant-number of variables and hence has a "brute-force" $\Sigma\Pi$ circuit).

We now describe how to prove that any $\Sigma\Pi\Sigma$ circuit for $P^{(1)}$ must have large size (the lower bound we obtain is $2^{\Omega(m)}$). Let F be a $\Sigma\Pi\Sigma$ circuit of size s computing $P^{(1)}$ and assume s is
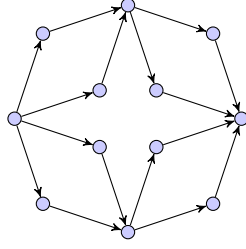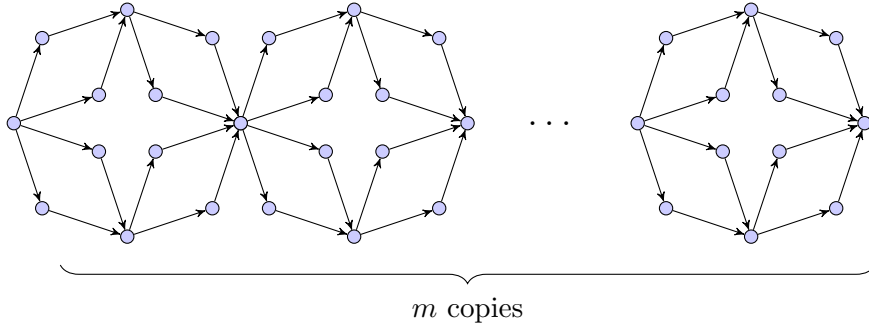
Figure 2: $H^{(1)}$.

m copies

Figure 3: $G^{(1)}$

small. We can write F as $T_1 + \cdots + T_s$, where each T_i is a product of linear polynomials.

The lower bound is via a rank argument first used by Raz [Raz06]. The idea is to associate a rank-based complexity measure with any polynomial f, and show that while the polynomial $P^{(1)}$ has rank as large as possible, the rank of F must be small.

This rank measure is defined as follows. 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 rank of the space of coefficients (as vectors over the base field $\mathbb{F}$) is considered a measure of the complexity of f.

It is easy to come up with a partition of the underlying variable set X into Y, Z so that the complexity of our polynomial $P^{(1)}$ is as large as it can be. Unfortunately, it is also easy to find $\Sigma\Pi\Sigma$ formulas that have maximum rank 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 $P^{(1)}$: we show that its complexity is *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* $\rho^{(1)} : X \to Y \cup Z \cup \{0, 1\}$ such that for any such restriction $\rho^{(1)}$, the resulting substitution of $P^{(1)}$, which we will call a *restriction of $P^{(1)}$*, continues to have full rank, but the rank of the restriction of F is small under many of them. We define these restrictions now.

The definition of the space of restrictions is motivated by the following easily verified observation (also used in many previous results [Raz04, RY08, RY09, DMPY12]): any polynomial of the form

$$\prod_{i \in [t]} (y_i + z_i) \tag{1}$$

is full-rank w.r.t. the natural partition $Y = \{y_1, \ldots, y_t\}$ and $Z = \{z_1, \ldots, z_t\}$. Given this, a possible option is to have the restriction $\rho^{(1)}$ set variables in each copy of $H^{(1)}$ in ways that ensure that after substitution, the polynomial is of the form in (1). We choose one among the three different (up to isomorphism) restrictions shown in Figure 4. It can be checked that each
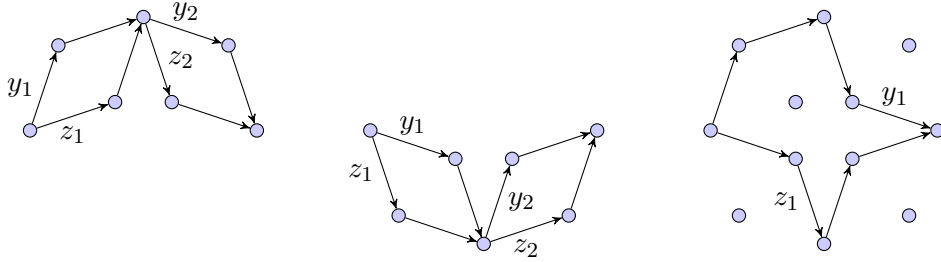
6

Figure 4: Restrictions applied to each copy of $H^{(1)}$. Edges that are not labelled have their variables set to 1. Edges that are not present have their variables set to 0.

such restriction results in a polynomial of the form in (1) and hence is full rank. Since $P^{(1)}$ is a product of many disjoint copies of this basic polynomial, it remains full rank also.

Now, to show that F has small rank under some such restriction, we apply a *uniformly random* copy of such a restriction $\rho^{(1)}$ to F and show that with high probability each of its constituent terms $T_1, \ldots, T_s$ is very small in rank. Since rank is subadditive and s is assumed to be small, this implies that F cannot be full rank after the restriction and hence cannot have been computing the polynomial $P^{(1)}$.

Fix a term T_i which is a product of linear functions as mentioned above. We argue that it restricts to a small-rank term with high probability as follows.

- A standard argument [Raz06] shows that, by multilinearity,[7] the rank of T_i is small if many of its constituent linear functions have smaller than "full-rank" after applying $\rho^{(1)}$. Here, a linear polynomial $L(Y', Z')$ ($Y' \subseteq Y$, $Z' \subseteq Z$) is said to be full-rank if its complexity (as defined above) is $2^{(|Y'|+|Z'|)/2}$.

- By a simple argument we can argue that, upon applying a random restriction $\rho^{(1)}$, any constituent linear function L of T_i becomes rank-deficient (i.e. short of full-rank) with good (constant) probability. This is by a case analysis on the set of variables X' that appear in L and proceeds roughly as follows.

  - Say X' is *structured* in that it is the union of the variable sets in some copies of $H^{(1)}$. In this case, we observe that with good probability, the total number of variables in L after restriction is larger than 2 and hence, the target full-rank is a number larger than 2. On the other hand, any linear function cannot have rank more than 2.

  - On the other hand if X' is not structured in the above sense, then it can be shown that with good probability, the number of restricted variables in L is odd and then it cannot be full-rank, since the target full-rank is $2^{t+1/2}$ for some integer t, while the rank of L can be at most 2^t.

- Given the above, we can easily argue using a concentration bound[8] that with high probability, T_i has *many* rank-deficient linear functions and consequently has small rank. This yields the proof of the $\Sigma\Pi\Sigma$ lower bound.

1.2.3 Separating Δ from $\Delta + 1$

To prove the lower bound for product-depth Δ circuits, we proceed by induction on Δ. Each step in the inductive proof is analogous to a step in the $\Sigma\Pi\Sigma$ lower bound so we will be brief.

---

[7]This essentially means that the linear functions that participate in T_i do not share any variables.

[8]By multilinearity, the events that the various linear functions are small rank become (roughly) independent.

The graph $H^{(\Delta)}$ is constructed from two copies of $G^{(\Delta-1)}$ by parallel composition and the graph $G^{(\Delta)}$ is constructed from many copies of $H^{(\Delta)}$ by series composition as shown in Figure 5. By construction $P^{(\Delta)}$ will have a circuit of product-depth $\Delta + 1$.
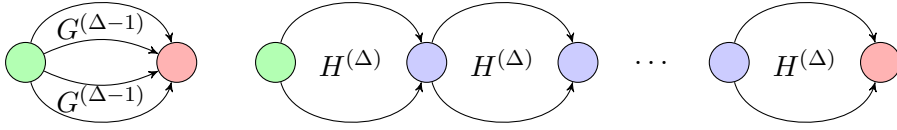

Figure 5: $H^{(\Delta)}$ (left) and $G^{(\Delta)}$

The lower bound is again proved via a random restriction argument. We define the random restriction $\rho^{(\Delta)}$ by restricting each copy of $H^{(\Delta)}$ independently by choosing one of these options at random.

- set it to a polynomial of the form $(y + z)$ by choosing a random path in each copy of $G^{(\Delta-1)}$ and setting a single variable in each to y or z (other variables in the path are set to 1 and off-path variables are set to 0),

- inductively applying the restriction $\rho^{(\Delta-1)}$ to one of the copies of $G^{(\Delta-1)}$ (setting the other one to 0).

As before, the polynomial $P^{(\Delta)}$ remains full-rank after the restriction with probability 1, since it always transforms into a polynomial of the form (1) after the restrictions.

Let F be a small product-depth Δ circuit. To argue that it is low-rank w.h.p. after applying $\rho^{(\Delta)}$, we prove a variant of a decomposition lemma from the literature that allows us to write $F = T_1 + \ldots + T_s$ where each T_i now is a product of circuits each of which has small size and product depth at most $\Delta - 1$.

As before, we argue that each T_i is low-rank with high probability. Say $T_i = \prod_{j \in [t]} Q_{i,j}$. To show this, it suffices to show that each $Q_{i,j}$ is somewhat short of full-rank with good probability. This does indeed turn out to be true in one of two ways.

- If the variable set X' of $Q_{i,j}$ is structured (as defined above), then we can use induction and show that it is low-rank if we happen to apply the restriction $\rho^{(\Delta-1)}$ to the variables of $Q_{i,j}$. This happens with constant probability.

- On the other hand, if $Q_{i,j}$ is unstructured, then with good probability, $Q_{i,j}$ is left with an odd number of variables, which implies that it is rank-deficient exactly as before.

At an intuitive level, the above cases correspond to two different ways in which a small circuit can try to compute the hard polynomial $P^{(\Delta)}$. In the first case, it uses the fact that the polynomial $P^{(\Delta)}$ is a product of many polynomials and tries to compute each of them with a smaller circuit of depth $\Delta - 1$; in this case, we use the inductive hypothesis to show that this cannot happen. In the second case, the circuit tries to do some non-trivial (i.e. unexpected) computation at the top level and in this case, we argue directly that the circuit fails.

This finishes the sketch of an idealized version of the argument. While the above strategy can be carried out as stated, it would yield a sub-optimal but still exponential bound of the form $\exp(n^{\alpha_\Delta})$ for some $\alpha_\Delta > 0$ that depends on Δ. To obtain the near-optimal lower bound of $\exp(n^{\Omega(1/\Delta)})$, we need to make the following changes to the above idealized proof strategy.

- We expand the set of "structured" polynomials to allow for polynomials $Q_{i,j}$ (and inductively formulas computed at smaller depths) that compute polynomials over a non-trivial

fraction, say ε, of the variables in many different copies, say M, of $H^{(\Delta-1)}$. The exact relationship between ε and M is somewhat delicate (we choose $\varepsilon \geq 1/M^{0.25}$) and must be chosen carefully for the proof to yield an optimal lower bound.

- Given this, we also need to handle (for the general inductive statement) the case that F does not depend on all the variables in $G^{(\Delta)}$ but rather at least δ-fraction of the variables in K many copies of $H^{(\Delta)}$ for some suitable δ and K. Given a single term $T_i = \prod_j Q_{i,j}$ in F as above, one of the following two cases occurs.

  - One of the $Q_{i,j}$s depends on at least a δ'-fraction of the variables in K' many copies of $H^{(\Delta)}$, where δ' and K' are not much smaller than δ and K respectively. This corresponds to the structured case above and in this case, we can actually use the induction hypothesis to show that we are done with high probability.

    To argue this, we have to use the fact that while δ has decreased to δ', the K' many copies of $H^{(\Delta)}$ contain many more copies of $H^{(\Delta-1)}$. Hence the number of copies of $H^{(\Delta-1)}$, say K'', and δ' still have the "correct" relationship so that the inductive statement is applicable.

  - Otherwise, in many copies of $H^{(\Delta)}$, the δ-fraction that are variables of F are further partitioned into parts of relative size $< \delta'$ by the variable sets of the $Q_{i,j}$. In this case, we have to argue that many of the $Q_{i,j}$ are rank-deficient with high probability. This is the most technical part of the proof and is done by a careful re-imagining of the sampling process that defines the random restriction.

The more general inductive statement (and some additional technicalities that force us to carry around auxiliary sets of Y and Z variables) results in a technically complicated theorem statement (see Theorem 16), which yields the near-optimal depth-hierarchy theorem.

Organization. We start with some Preliminaries in Section 2. We define the hard polynomials that we use in Section 3. The general inductive statement (Theorem 16) is given in Section 4, which also contains the proof of the main depth-hierarchy theorem (Corollary 14) assuming Theorem 16. Finally, we prove Theorem 16 in Section 5.

2 Preliminaries

2.1 Polynomials and restrictions

Throughout, let $\mathbb{F}$ be an arbitrary field. 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.

Let X, Y, Z be disjoint sets of variables. An (X, Y, Z)-*restriction* is a function $\rho : X \to Y \cup Z \cup \{0, 1\}$. The sets X, Y and Z will sometimes be omitted when clear from context. We say that the restriction ρ is *multilinear* if no two variables in X are mapped to the same variable in $Y \cup Z$ by ρ. A *random* (X, Y, Z)-restriction is simply a random function $\rho : X \to Y \cup Z \cup \{0, 1\}$ (chosen according to some distribution).

Let X, Y, Z be as above and say we have for each $i \in [k]$, an (X_i, Y_i, Z_i)-restriction ρ_i where $X_i \subseteq X, Y_i \subseteq Y$ and $Z_i \subseteq Z$. If $X_1, \ldots, X_k$ form a (pairwise disjoint) partition of X, we define their *composition* — denoted $\rho_1 \circ \rho_2 \cdots \circ \rho_k$ — to be the (X, Y, Z)-restriction ρ such that $\rho(x)$ agrees with $\rho_i(x)$ for each $x \in X_i$. Further, if restrictions ρ_i are multilinear restrictions and the sets Y_i ($i \in [k]$) and Z_j ($j \in [k]$) are pairwise disjoint, then $\rho_1 \circ \cdots \circ \rho_k$ is also multilinear.

Let ρ be an (X, Y, Z)-restriction. Given a polynomial $f \in \mathbb{F}[X]$, the restriction ρ yields a natural polynomial in $\mathbb{F}[Y \cup Z]$ obtained from f by substitution; we denote this polynomial $f|_\rho$. Note, moreover, that if f is multilinear and ρ is multilinear, then so is $f|_\rho$.

2.2 Partial derivative matrices and relative rank

Let Y and Z be two disjoint sets of variables and let $g \in \mathbb{F}[Y \cup Z]$ be a multilinear polynomial. Define the $2^{|Y|} \times 2^{|Z|}$ matrix $M_{(Y,Z)}(g)$ whose rows and columns 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. We will use the rank of this matrix as a measure of the complexity of g.

We define the *relative-rank* of g w.r.t. (Y, Z) (denoted by $\mathrm{relrk}_{(Y,Z)}(g)$) by

$$\mathrm{relrk}_{(Y,Z)}(g) = \frac{\mathrm{rank}(M_{(Y,Z)}(g))}{2^{(|Y|+|Z|)/2}}.$$

We note the following properties of relative rank [RY09].

Proposition 2. *Let $g, g_1, g_2 \in \mathbb{F}[Y \cup Z]$ be multilinear polynomials.*

1. $\mathrm{relrk}_{(Y,Z)}(g) \leq 1$. *Further if $|Y| + |Z|$ is odd, $\mathrm{relrk}_{(Y,Z)}(g) \leq 1/\sqrt{2}$.*

2. $\mathrm{relrk}_{(Y,Z)}(g_1 + g_2) \leq \mathrm{relrk}_{(Y,Z)}(g_1) + \mathrm{relrk}_{(Y,Z)}(g_2)$.

3. *If Y is partitioned into Y_1, Y_2 and Z into Z_1, Z_2 with $g_i \in \mathbb{F}[Y_i \cup Z_i]$ ($i \in [2]$), then $\mathrm{rank}(M_{(Y,Z)}(g)) = \mathrm{rank}(M_{(Y_1,Z_1)})(g_1) \cdot \mathrm{rank}(M_{(Y_2,Z_2)})(g_2)$. In particular, $\mathrm{relrk}_{(Y,Z)}(g_1 \cdot g_2) = \mathrm{relrk}_{(Y_1,Z_1)}(g_1) \cdot \mathrm{relrk}_{(Y_2,Z_2)}(g_2)$.*

2.3 Multilinear models of computation

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 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 the depth of the formula or circuit will refer to the maximum number of gates on a 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 the input gate to output gate. Note that if a formula or circuit has depth Δ, we can assume without loss of generality that its product depth is between $\lfloor \Delta/2 \rfloor$ and $\lceil \Delta/2 \rceil$ (by collapsing sum and product gates if necessary).

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 polynomial. Moreover, a formula F is said to be *syntactic multilinear* if for each $\times$-gate Φ of F with children $\Psi_1, \ldots, \Psi_t$, we have $\mathrm{Supp}(\Psi_i) \cap \mathrm{Supp}(\Psi_j) = \emptyset$ for each $i \neq j$, where $\mathrm{Supp}(\Phi)$ denotes the set of variables that appear in the subformula 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 gate and along any path from an input gate to the output gate, the sum and product gates alternate, with product gates 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.
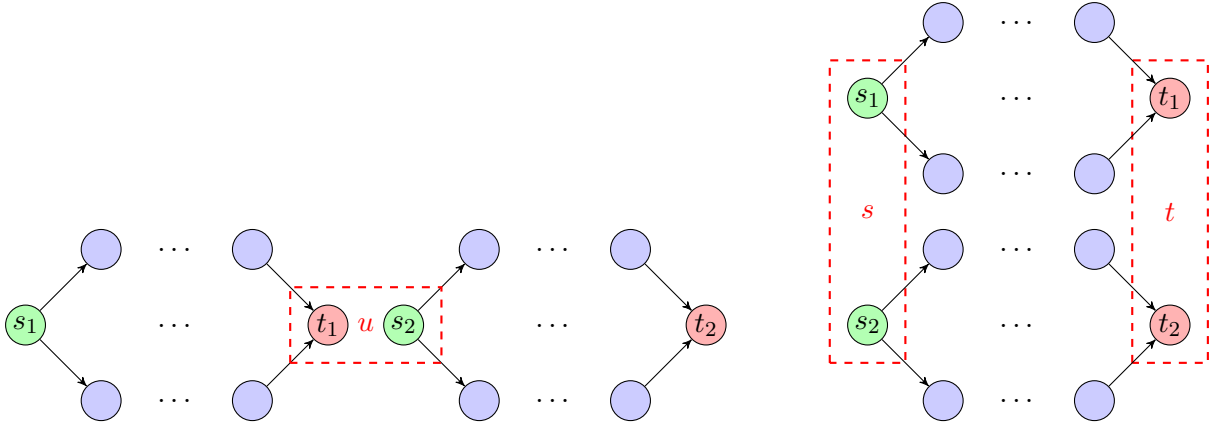
Figure 6: Composition of ABPs in series and parallel.

An *Algebraic Branching Program* (ABP), over the set of variables X and field $\mathbb{F}$ is a layered (i.e., the edges are only between two consecutive layers) directed acyclic graph G with two special vertices called the *source* and the *sink*. The label of an edge is a linear polynomial in $\mathbb{F}[X]$. The weight of a path is the product of the labels of its edges. The polynomial computed by G is the sum of the weights of all the paths from s to t in G.

A *Multilinear ABP* (mlABP) is an algebraic branching program such that for any path from the source to sink, the labels of edges on that path are linear polynomials over pairwise-disjoint sets of variables.

Composing ABPs in series and in parallel. Let $G_1, \ldots, G_k$ be ABPs (on disjoint sets of vertices) with sources $s_1, \ldots, s_k$ and sinks $t_1, \ldots, t_k$. We say that G is obtained by composing $G_1, \ldots, G_k$ *in parallel* if G is obtained by identifying all sources $s_1, \ldots, s_k$ to obtain a single source s and all the sinks $t_1, \ldots, t_k$ to obtain a single sink t. Note that the polynomial computed by G is the sum of the polynomials computed by $G_1, \ldots, G_k$.

We say that G is obtained by composing $G_1, \ldots, G_k$ *in series* if G is obtained by identifying t_i with s_{i+1} for each $i \in [k-1]$ to obtain an ABP with source s_1 and sink t_k. Note that the polynomial computed by G is the product of the polynomials computed by $G_1, \ldots, G_k$. Further if $G_1, \ldots, G_k$ are mlABPs over disjoint sets of variables, then G is also an mlABP.

2.4 Some structural lemmas

Given a syntactically multilinear formula F computing a polynomial $P \in \mathbb{F}[X]$, we define a *variable-set labelling* of F to be a labelling function that assigns to each gate Φ of F a set $\mathrm{Vars}(\Phi) \subseteq X$ with the following properties.

1. For any gate Φ in F, $\mathrm{Supp}(\Phi) \subseteq \mathrm{Vars}(\Phi)$.

2. If Φ is an sum gate, with children $\Psi_1, \Psi_2, \ldots, \Psi_k$, then $\forall i \in [k]$, $\mathrm{Vars}(\Psi_i) = \mathrm{Vars}(\Phi)$.

3. If Φ is a product gate, with children $\Psi_1, \Psi_2, \ldots, \Psi_k$, then $\mathrm{Vars}(\Phi) = \cup_{i=1}^k \mathrm{Vars}(\Psi_i)$ and the sets $\mathrm{Vars}(\Psi_i)$ ($i \in [k]$) are pairwise disjoint.

We call a syntactically multilinear formula F *variable-labelled* if it is equipped with a labelling function as above. For such an F, we define $\mathrm{Vars}(F)$ to be $\mathrm{Vars}(\Phi_0)$, where Φ_0 is the output gate of F.

The following lemma shows that we may always assume that a syntactically multilinear formula F is variable-labelled.

Proposition 3. *For each syntactically multilinear formula F computing $P \in \mathbb{F}[X]$, there is a variable-labelled syntactically multilinear formula F' of the same size as F computing P such that $\mathrm{Vars}(\Phi) \subseteq X$ for each gate Φ of F' and $\mathrm{Vars}(F) = X$.*

Proof. The formula F' is the same as the formula F along with variable-set labels $\mathrm{Vars}(\Phi)$ for each gate Φ of F'. These labels are defined by downward induction on the structure of F as follows.

For the output gate Φ_0 of F, we define $\mathrm{Vars}(\Phi_0) = X$ (this ensures that $\mathrm{Vars}(F) = X$). If Φ is a sum gate with children $\Psi_1, \ldots, \Psi_k$ and $\mathrm{Vars}(\Phi) = S \subseteq X$, then for each $1 \leq i \leq k$, we define $\mathrm{Vars}(\Psi_i) = S$. If Φ is a product gate with children $\Psi_1, \ldots, \Psi_k$ and $\mathrm{Vars}(\Phi) = S \subseteq X$, then we define $\mathrm{Vars}(\Psi_i) = \mathrm{Supp}(\Psi_i)$ for $1 \leq i \leq k-1$ and $\mathrm{Vars}(\Psi_k) = S \setminus \left(\cup_{i=1}^{k-1} \mathrm{Vars}(\Psi_i) \right)$.

It is easy to check that the above labelling is indeed a valid variable-set labelling. $\square$

We will use the following structural result that converts any small-depth multilinear circuit to a small-depth syntactic multilinear formula without a significant blowup in size.

Lemma 4 (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.*

2.5 Useful Probabilistic Estimates

We will need the Chernoff bound [Che52, Hoe63] for sums of independent Boolean random variables. We use the version from the book of Dubhashi and Panconesi [DP09, Theorem 1.1].

Theorem 5 (Chernoff bound). *Let $W_1, \ldots, W_n$ be independent $\{0,1\}$-valued random variables and let $W = \sum_{i \in [n]} W_i$. Then we have the following.*

1. For any $\varepsilon > 0$,

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

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

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

We will need the following simple fact that reduces the problem of proving a large deviation bound to the problem of bounding the probability of a conjunction of events.

Proposition 6. *Let $W_1, \ldots, W_n$ be n $\{0,1\}$-valued random variables (not necessarily independent) and let $W = (W_1, \ldots, W_n)$. Assume that for some $p \in [0,1]$ and for any $\beta \in \{0,1\}^n$ we have $\Pr[W = \beta] \leq p$. Then, we have $\Pr[\sum_i W_i \leq r] \leq n^r \cdot p$.*

Proof. For any $\beta \in \{0,1\}^n$, let weight of β (denoted $\mathrm{wt}(\beta)$) be $\sum_{i=1}^n \beta(i)$. It is now easy to see that

$$\Pr[\sum_i W_i \leq r] = \sum_{k=0}^r \left(\sum_{\substack{\beta \in \{0,1\}^n \\ \mathrm{wt}(\beta)=k}} \Pr[W = \beta] \right) \leq \sum_{k=0}^r \binom{n}{k} \cdot p \leq n^r \cdot p.$$

The first inequality comes from the fact that there are $\binom{n}{k}$ many bit vectors β in $\{0,1\}^n$ of weight exactly k and $\Pr[W = \beta] \leq p$ for each of them, and the last inequality comes from an upper bound of n^r on $\sum_{k=0}^r \binom{n}{k}$.

$\square$

For a Boolean random variable W, define the *unbias* of W by

$$\text{Unbias}(W) = \min\{\Pr[W = 0], \Pr[W = 1]\}$$

and *bias* of W by

$$\text{Bias}(W) = \left|\Pr[W = 0] - \Pr[W = 1]\right| = \left|\mathbb{E}[(-1)^W]\right|.$$

Note that $\text{Unbias}(W) \in [0, 1/2]$. The following fact follows directly from the aforementioned definitions.

Proposition 7. *For a Boolean random variable W, $\text{Bias}(W) + 2 \cdot \text{Unbias}(W) = 1$.*

The following fact is folklore, but we state it here in the exact form we need and prove it for completeness.

Proposition 8. *Let $W_1, \ldots, W_n$ be independent $\{0,1\}$-valued random variables and assume $W = \bigoplus_{j \in [n]} W_j$. Then $\text{Unbias}(W) \geq \min\{(1/2) \cdot \sum_j \text{Unbias}(W_j), 1/10\}$.*

Proof. From Proposition 7, we get that $\text{Unbias}(W) = \frac{1 - \text{Bias}(W)}{2} = \frac{1 - |\mathbb{E}[(-1)^W]|}{2}$. We also get the following.

$$\left|\mathbb{E}[(-1)^W]\right| = \left|\mathbb{E}[(-1)^{\bigoplus_{j \in [n]} W_j}]\right| = \left|\mathbb{E}[(-1)^{\sum_{j \in [n]} W_j}]\right| = \left|\prod_{j \in [n]} \mathbb{E}[(-1)^{W_j}]\right| = \prod_{j \in [n]} \text{Bias}(W_j).$$

Thus,

$$\text{Unbias}(W) = \frac{1 - \prod_{j \in [n]}(1 - 2 \cdot \text{Unbias}(W_j))}{2} \geq \frac{1 - \prod_{j \in [n]} e^{-2 \cdot \text{Unbias}(W_j)}}{2}$$
$$= \frac{1 - e^{-2(\sum_{j \in [n]} \text{Unbias}(W_j))}}{2}.$$

If $\sum_{j \in [n]} \text{Unbias}(W_j) \geq \frac{1}{2}$, then $\text{Unbias}(W) \geq \frac{1 - e^{-1}}{2} \geq \frac{1}{10}$. Else, $\text{Unbias}(W) \geq \frac{\sum_{j \in [n]} \text{Unbias}(W_j)}{2}$. The latter follows from the inequality $e^x \leq 1 - \frac{x}{2}$ for all $x \in [0, 1/2]$. This proves the proposition. $\square$

Let $A_1, \ldots, A_N$ be any N independent random variables taking values over any finite set. Let $B_1, \ldots, B_M$ be M *Boolean* random variables with $B_i = f_i(A_j : j \in S_i)$ for some function f_i and some $S_i \subseteq [N]$. We say $B_1, \ldots, B_M$ are read-k for $k \geq 1$ if each $j \in [N]$ belongs to at most k many sets S_i.

A result of Janson [Jan04] yields concentration bounds for sums of read-k Boolean random variables. We use the following form of the bound that appears in the result of Gavinsky, Lovett, Saks and Srinivasan [GLSS15].

Theorem 9 ([GLSS15]). *If $B_1, \ldots, B_M$ are read-k Boolean random variables and $B = \sum_i B_i$, then*

$$\Pr[B \leq \frac{1}{2} \boldsymbol{E}[B]] \leq \exp(-\Omega(\boldsymbol{E}[B]/k)).$$

3 The Hard polynomials and their restrictions

3.1 The Hard polynomials

The hard polynomial for circuits of product-depth Δ will be defined to be the polynomial computed by a suitable ABP. The definition is by induction on the product depth Δ. We also define some auxiliary variable sets that will be useful later when we restrict these polynomials.

Let $m \in \mathbb{N}$ be a growing parameter. We will define for each $\Delta \geq 1$ an ABP $G^{(\Delta)}$ on a variable set $X^{(\Delta)}$ of size n_Δ, along with some auxiliary variable sets $Y^{(\Delta)}$ and $Z^{(\Delta)}$. The parameter n_Δ itself is defined inductively as follows. Let $n_0 = 8$ and for each $\Delta \geq 1$, define

$$n_\Delta = 2 \cdot m \cdot n_{\Delta-1}. \tag{2}$$

Observe that $n_\Delta = 8 \cdot (2m)^\Delta$ for $\Delta \geq 1$.

The Variable sets. For each $\Delta \geq 0$, we will define three disjoint variable sets $X^{(\Delta)}, Y^{(\Delta)}$ and $Z^{(\Delta)}$. We will have $|X^{(\Delta)}| = n_\Delta$ for each Δ.

Let $X^{(0)}$ be the set $\{x_{i,j}^{(0)} \mid i \in [4], j \in [2]\}$, $Y^{(0)} = \{y_1^{(0)}, y_2^{(0)}\}$ and $Z^{(0)} = \{z_1^{(0)}, z_2^{(0)}\}$. Note that $|X^{(0)}| = n_0 = 8$.

Given $X^{(\Delta)}, Y^{(\Delta)}$ and $Z^{(\Delta)}$, we now define $X^{(\Delta+1)}, Y^{(\Delta+1)}$ and $Z^{(\Delta+1)}$ as follows.

Clones, segments and half-segments. For each $i \in [m]$ and $j \in [2]$, let $X_{i,j}^{(\Delta+1)}$, $Y_{i,j}^{(\Delta+1)}$ and $Z_{i,j}^{(\Delta+1)}$ be pairwise disjoint copies of $X^{(\Delta)}, Y^{(\Delta)}$ and $Z^{(\Delta)}$ respectively. Define $X^{(\Delta+1)} = \bigcup_{i,j} X_{i,j}^{(\Delta+1)}$; note that $|X^{(\Delta+1)}| = 2m|X^{(\Delta)}| = n_{\Delta+1}$ as desired. Define $Y^{(\Delta+1)}$ to be $\bigcup_i Y_{i,1}^{(\Delta+1)} \cup Y_{i,2}^{(\Delta+1)} \cup \{y_i^{(\Delta+1)}\}$ and $Z^{(\Delta+1)}$ to be $\bigcup_i Z_{i,1}^{(\Delta+1)} \cup Z_{i,2}^{(\Delta+1)} \cup \{z_i^{(\Delta+1)}\}$, where $y_i^{(\Delta+1)}$ and $z_i^{(\Delta+1)}$ are fresh variables.

By the inductive definition of the sets $X^{(\Delta+1)}$ above, we see that $X^{(\Delta+1)}$ is made up of $2m$ copies of $X^{(\Delta)}$, which we denote by $X_{i,j}^{(\Delta+1)}$ for $i \in [m]$ and $j \in [2]$. Using this fact inductively, we see that for any $t \in \{0, \ldots, \Delta+1\}$, $X^{(\Delta+1)}$ contains $(2m)^t$ copies of $X^{(\Delta+1-t)}$. Each such copy is uniquely labelled by a tuple $\omega = ((i_1, j_1), \ldots, (i_t, j_t)) \in ([m] \times [2])^t$; we denote this copy by $X_\omega^{(\Delta+1)}$ and call this an $X^{(\Delta+1-t)}$-*clone*. In a similar way, we see that for each $\omega \in ([m] \times [2])^t$, $Y^{(\Delta+1)}$ and $Z^{(\Delta+1)}$ contain copies of $Y^{(\Delta+1-t)}$ and $Z^{(\Delta+1-t)}$ respectively, which we denote as $Y_\omega^{(\Delta+1)}$ and $Z_\omega^{(\Delta+1)}$ respectively, and call $Y^{(\Delta+1-t)}$-clones and $Z^{(\Delta+1-t)}$-clones respectively.

Say $X_\omega^{(\Delta')}$ is an $X^{(\Delta)}$-clone for some $\Delta' \geq \Delta \geq 1$. Then we refer to $X_{(\omega,(i,j))}^{(\Delta')}$ (which is an $X^{(\Delta-1)}$-clone) as the (i,j)th *half-segment* of $X_\omega^{(\Delta')}$. Further, we refer to $X_{(\omega,(i,1))}^{(\Delta')} \cup X_{(\omega,(i,2))}^{(\Delta')}$ as the ith *segment* of $X_\omega^{(\Delta')}$, which we will denote $X_{(\omega,i)}^{(\Delta')}$; note that $X_\omega^{(\Delta')}$ is a union of its m segments. Similarly a $Y^{(\Delta)}$-clone $Y_\omega^{(\Delta')}$ (resp. a $Z^{(\Delta)}$-clone $Z_\omega^{(\Delta')}$) is a disjoint union of m segments $Y_{(\omega,i)}^{(\Delta')}$ (resp. $Z_{(\omega,i)}^{(\Delta')}$) for $i \in [m]$, the ith of which is made up of two half-segments $Y_{(\omega,(i,1))}^{(\Delta')}$ and $Y_{(\omega,(i,2))}^{(\Delta')}$ (resp. $Z_{(\omega,(i,1))}^{(\Delta')}$ and $Z_{(\omega,(i,2))}^{(\Delta')}$) and an additional fresh variable that we denote $y_{(\omega,i)}^{(\Delta')}$ (resp. $z_{(\omega,i)}^{(\Delta')}$). We refer to a set of the form $X_{(\omega,i)}^{(\Delta')}$ as an $X^{(\Delta)}$-segment[9] and a set of the form $X_{(\omega,(i,j))}^{(\Delta')}$ as an $X^{(\Delta)}$-half-segment (similarly $Y^{(\Delta)}$-segment, $Y^{(\Delta)}$-half-segment, $Z^{(\Delta)}$-segment and $Z^{(\Delta)}$-half-segment).

---

[9]The reader may want to read "$X^{(\Delta)}$-segment" as "a segment of (a clone of) $X^{(\Delta)}$" and similarly for other segments and half-segments.
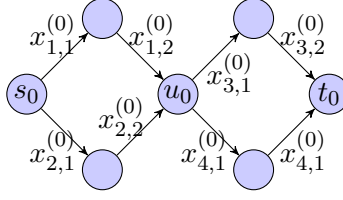
Figure 7: The ABP $G^{(0)}$. All edges go from left to right.

To summarize, given any $\Delta' \geq \Delta \geq 0$ the set $X^{(\Delta')}$ contains $(2m)^{\Delta'-\Delta}$ many $X^{(\Delta)}$-clones which are indexed by elements of the set $([m] \times [2])^{\Delta'-\Delta}$. If $\Delta \geq 1$, each such $X^{(\Delta)}$-clone contains m many $X^{(\Delta)}$-segments, which are indexed by the set $([m] \times [2])^{\Delta'-\Delta} \times [m]$; and each segment contains two $X^{(\Delta-1)}$-clones, also referred to as $X^{(\Delta)}$-half-segments. Finally, these definitions extend naturally to $Y^{(\Delta')}$- and $Z^{(\Delta')}$-clones.

The Hard polynomial. The hard polynomial for product-depth Δ is defined by induction on Δ.

Define $G^{(0)}$ to be an ABP as follows. Assume that $G^{(0)}$ has source vertex s_0, sink vertex t_0 and an intermediate vertex u_0. The graph of the ABP consists of two disjoint paths π_1 and π_2 of length 2 each from s_0 to u_0 and two disjoint paths π_3 and π_4 of length 2 each from u_0 to t_0 (see Figure 7). The edges of π_i are labelled by the distinct variables $x_{i,1}^{(0)}$ and $x_{i,2}^{(0)}$. Let $X^{(0)}$ be the set $\{x_{i,j}^{(0)} \mid i \in [4], j \in [2]\}$ of variables that appear in $G^{(0)}$. Note that $G^{(0)}$ computes a polynomial over the variable set $X^{(0)}$ defined above.

Now fix any $\Delta \geq 0$. Given $G^{(\Delta)}$, an ABP on variable set $X^{(\Delta)}$, we inductively define the ABP $G^{(\Delta+1)}$ an ABP on variable set $X^{(\Delta+1)}$ as follows.

- For each $i \in [m]$ and $j \in [2]$, let $G_{i,j}^{(\Delta+1)}$ be a copy of $G^{(\Delta)}$ on the variable set $X_{i,j}^{(\Delta+1)}$.

- Let $H_i^{(\Delta+1)}$ be the ABP obtained by composing $G_{i,1}^{(\Delta+1)}$ and $G_{i,2}^{(\Delta+1)}$ in parallel. This ABP is defined over variable set $X_i^{(\Delta+1)}$.

- Finally, let $G^{(\Delta+1)}$ be the ABP obtained by composing $H_1^{(\Delta+1)}, \ldots, H_m^{(\Delta+1)}$ in series, in that order.

From the definition of $G^{(\Delta)}$ it is easy to observe the following properties.

Proposition 10. *For any $\Delta \geq 0$, $G^{(\Delta)}$ has a unique source, say s_Δ, and a unique sink, say t_Δ. Also, each edge in $G^{(\Delta)}$ appears on some source to sink path.*

We define $P^{(\Delta)}$ to be the multilinear polynomial in $\mathbb{F}[X^{(\Delta)}]$ computed by the ABP $G^{(\Delta)}$. We can also note the following properties of the polynomial computed by $G^{(\Delta)}$.

Proposition 11 (Properties of $P^{(\Delta)}$).   *1. $P^{(0)}$ is the polynomial $\sum_{i_1 \in \{1,2\}, i_2 \in \{3,4\}} x_{i_1,1}^{(0)} \cdot x_{i_1,2}^{(0)} \cdot x_{i_2,1}^{(0)} \cdot x_{i_2,2}^{(0)}$.*

   2. For each $\Delta \geq 0$,

$$P^{(\Delta+1)}(X^{(\Delta+1)}) = \prod_{i \in [m]} \left(P^{(\Delta)}(X_{i,1}^{(\Delta+1)}) + P^{(\Delta)}(X_{i,2}^{(\Delta+1)}) \right)$$

3. *The polynomial $P^{(0)}$ is computed by a $\Sigma\Pi$ multilinear formula of size $O(1)$. For each $\Delta \geq 1$, $P^{(\Delta)}$ can be computed by a syntactic multilinear $(\Pi\Sigma)^\Delta\Pi$ formula of size $O(n_\Delta)$.*

As with the variable sets, we see that for any $\Delta' \geq \Delta \geq 1$, and for each $\omega \in ([m] \times [2])^{\Delta'-\Delta}$, the ABP $G^{(\Delta')}$ contains a corresponding copy of $G^{(\Delta)}$ on the variable set $X_\omega^{(\Delta')}$. We denote this copy of $G^{(\Delta)}$ by $G_\omega^{(\Delta')}$. The ABP $G_\omega^{(\Delta')}$ is obtained by composing in series the ABPs $H_{(\omega,1)}^{(\Delta')}, \ldots, H_{(\omega,m)}^{(\Delta')}$ (defined on the segments $X_{(\omega,1)}^{(\Delta')}, \ldots, X_{(\omega,m)}^{(\Delta')}$ respectively) in that order.

3.2 Restrictions

We define a random multilinear[10] $(X^{(\Delta+1)}, Y^{(\Delta+1)}, Z^{(\Delta+1)})$-restriction $\rho^{(\Delta+1)}$ by an inductively defined sampling process.

Base case. Let $\rho^{(0)}$ be defined as follows: $\rho^{(0)}(x_{1,1}^{(0)}) = y_1^{(0)}, \rho^{(0)}(x_{2,1}^{(0)}) = z_1^{(0)}, \rho^{(0)}(x_{3,1}^{(0)}) = y_2^{(0)}, \rho^{(0)}(x_{4,1}^{(0)}) = z_2^{(0)}$ (see Figure 3.3). That is, we set the first variable in each of the paths $\pi_1, \pi_2, \pi_3, \pi_4$ to a distinct variable in $Y^{(0)} \cup Z^{(0)}$. Also, we set $\rho^{(0)}(x_{i,2}^{(0)}) = 1$ for each $i \in [2]$. That is, we set all the remaining variables to the constant 1. (Note that $\rho^{(0)}$ is in fact a *deterministic* $(X^{(0)}, Y^{(0)}, Z^{(0)})$-restriction.) It can be checked that $\rho^{(0)}$ is multilinear.

Inductive case. Let us now assume that we have defined the process for sampling the random multilinear $(X^{(\Delta)}, Y^{(\Delta)}, Z^{(\Delta)})$-restriction $\rho^{(\Delta)}$. We define $\rho^{(\Delta+1)}$ by the following sampling process.

For each $i \in [m]$, we sample a random multilinear $(X_i^{(\Delta+1)}, Y_i^{(\Delta+1)}, Z_i^{(\Delta+1)})$-restriction $\rho_i^{(\Delta+1)}$ independently using the general sampling process described below. We then define $\rho^{(\Delta+1)}$ as a composition of these restrictions, i.e. $\rho^{(\Delta+1)} = \rho_1^{(\Delta+1)} \circ \cdots \circ \rho_m^{(\Delta+1)}$ (as defined in Section 2.1). Clearly, $\rho^{(\Delta+1)}$ is multilinear since each $\rho_i^{(\Delta+1)}$ is multilinear.

We now give a general sampling process to sample a random multilinear (X, Y, Z)-restriction where X is an $X^{(\Delta+1)}$-segment and Y and Z are the corresponding $Y^{(\Delta+1)}$- and $Z^{(\Delta+1)}$-segments respectively.

For the remainder of this section, let $\Delta' \geq \Delta + 1 \geq 1$ be arbitrary and for some $\omega \in ([m] \times [2])^{\Delta'-\Delta-1}$ and $i \in [m]$, let X denote the ith segment $X_{(\omega,i)}^{(\Delta')}$ of the $X^{(\Delta+1)}$-clone $X_\omega^{(\Delta')}$ and let $Y = Y_{(\omega,i)}^{(\Delta')}$, $Z = Z_{(\omega,i)}^{(\Delta')}$. For any $j \in [2]$, let $X_j, Y_j,$ and Z_j denote the $X^{(\Delta)}$-clones $X_{(\omega,(i,j))}^{(\Delta')}, Y_{(\omega,(i,j))}^{(\Delta')}$ and $Z_{(\omega,(i,j))}^{(\Delta')}$ respectively, and let G_j denote the ABP $G_{(\omega,(i,j))}^{(\Delta')}$.

Let y, z denote the variables $y_{(\omega,i)}^{(\Delta')}$ and $z_{(\omega,i)}^{(\Delta')}$ in the sets Y and Z respectively (recall that we have $Y = Y_1 \cup Y_2 \cup \{y\}$ and $Z = Z_1 \cup Z_2 \cup \{z\}$). Let H denote the ABP $H_{(\omega,i)}^{(\Delta')}$ (recall that H is the parallel composition of G_1 and G_2).

We show now how to sample for any such X, Y, Z a random multilinear (X, Y, Z)-restriction ρ.

Sampling Algorithm $\mathcal{A}$ for (X, Y, Z)-restriction ρ.

E_1: Set all the variables from the set X_2 to 0. For the variables in X_1, sample a random (X_1, Y_1, Z_1)-restriction ρ_1 using the sampling procedure for $\rho^{(\Delta)}$. (Recall that X_1, Y_1, Z_1 are $X^{(\Delta)}$-, $Y^{(\Delta)}$-, and $Z^{(\Delta)}$-clones respectively.) Set variables in X_1 according to ρ_1.

---

[10]See Section 2.1 for the definition.

16

E_2: This is the same as E_1 except that the roles of X_1 and X_2 are exchanged. Formally, we set all the variables from the set X_1 to 0 and apply a random (X_2, Y_2, Z_2)-restriction ρ_2, sampled using the sampling procedure for $\rho^{(\Delta)}$, to X_2.

E_3: Choose variables x_1 and x_2 independently and uniformly at random from X_1 and X_2 respectively. The variable x_j ($j \in [2]$) labels a unique edge, say e_j, in ABP G_j; let π_{e_j} denote the lexicographically smallest source to sink path in $G_{i,j}^{(\Delta+1)}$ containing e_j. For any variable $x \in X$ which does not label an edge on either of the paths π_{e_1} or π_{e_2}, set x to the constant 0.

Set x_1 to y and x_2 to z.

For any edge e other than e_1, e_2 which lies on either π_{e_1} or π_{e_2}, set the variable labelling e to 1.

3.3 Properties of $\rho^{(\Delta)}$

Fix some $\Delta \geq 1$. For a given random choice of $\rho^{(\Delta)}$, we use $\tilde{Y}$ and $\tilde{Z}$ to denote $\text{Img}(\rho^{(\Delta)}) \cap Y^{(\Delta)}$ and $\text{Img}(\rho^{(\Delta)}) \cap Z^{(\Delta)}$ respectively. Note that these are *random* sets.

Given any $f \in \mathbb{F}[X^{(\Delta)}]$, the polynomial $f|_{\rho^{(\Delta)}}$ belongs to the set $\mathbb{F}[\tilde{Y} \cup \tilde{Z}]$. By the multilinearity of $\rho^{(\Delta)}$, if f is multilinear, then so is $f|_{\rho^{(\Delta)}}$.

Lemma 12. *With probability* 1, $\text{relrk}_{(\tilde{Y}, \tilde{Z})}(P^{(\Delta)}|_{\rho^{(\Delta)}}) = 1$.

Proof. We prove the lemma by induction on Δ.

The base case corresponds to $\Delta = 0$, in which case $\tilde{Y} = Y^{(0)} = \{y_1^{(0)}, y_2^{(0)}\}$ and $\tilde{Z} = Z^{(0)} = \{z_1^{(0)}, z_2^{(0)}\}$ deterministically. From the definition of $P^{(0)}$ and $\rho^{(0)}$, we know that $P^{(0)}|_{\rho^{(0)}} = (y_1^{(0)} + z_1^{(0)})(y_2^{(0)} + z_2^{(0)})$, which can easily be seen to have relative rank 1 w.r.t. the partition $(\tilde{Y}, \tilde{Z})$.
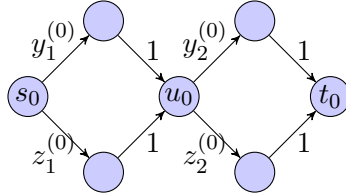

Figure 8: The effect of $\rho^{(0)}$ on $G^{(0)}$

Recall (Proposition 11) that for each $\Delta \geq 1$,

$$P^{(\Delta)}(X^{(\Delta)}) = \prod_{i \in [m]} \left(P^{(\Delta-1)}(X_{i,1}^{(\Delta)}) + P^{(\Delta-1)}(X_{i,2}^{(\Delta)}) \right)$$

Let $Q_i^{(\Delta)}$ be equal to $P^{(\Delta-1)}(X_{i,1}^{(\Delta)}) + P^{(\Delta-1)}(X_{i,2}^{(\Delta)})$. The random restriction $\rho^{(\Delta)}$ is defined to be the composition $\rho_1^{(\Delta)} \circ \cdots \circ \rho_m^{(\Delta)}$ where each $\rho_i^{(\Delta)}$ is a random multilinear $(X_i^{(\Delta)}, Y_i^{(\Delta)}, Z_i^{(\Delta)})$-restriction sampled according to the algorithm $\mathcal{A}$ described in Section 3.2. Thus, we have for any fixing of $\rho_i^{(\Delta)}$ ($i \in [m]$),

$$P^{(\Delta)}|_{\rho^{(\Delta)}} = \prod_{i \in [m]} Q^{(\Delta)}|_{\rho_i^{(\Delta)}}$$

17

and hence by Proposition 2, we have

$$\mathrm{relrk}_{(\tilde{Y},\tilde{Z})}(P^{(\Delta)}|_{\rho^{(\Delta)}}) = \prod_{i\in[m]} \mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(Q^{(\Delta)}|_{\rho_i^{(\Delta)}})$$

where $\tilde{Y}_i := \mathrm{Img}(\rho_i^{(\Delta)}) \cap Y_i^{(\Delta)}$ and $\tilde{Z}_i := \mathrm{Img}(\rho_i^{(\Delta)}) \cap Z_i^{(\Delta)}$. So it suffices to argue that each term in the above product is 1 with probability 1.

Fix an $i \in [m]$ and consider $\mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(Q^{(\Delta)}|_{\rho_i^{(\Delta)}})$. There are three possibilities for the sampling algorithm $\mathcal{A}$ in choosing $\rho_i^{(\Delta)}$.

Say $\mathcal{A}$ picks option E_1. Then, all variables in $X^{(\Delta)} \setminus X_{i,1}^{(\Delta)} = X_{i,2}^{(\Delta)}$ are set to 0 and $\rho_i^{(\Delta)}$ is simply a copy of $\rho^{(\Delta-1)}$ defined w.r.t. the sets $(X_{i,1}^{(\Delta)}, Y_{i,1}^{(\Delta)}, Z_{i,1}^{(\Delta)})$. Thus, by induction $\mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(Q^{(\Delta)}|_{\rho_i^{(\Delta)}}) = \mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(P^{(\Delta-1)}|_{\rho_i^{(\Delta)}}) = 1$. A similar reasoning works when $\mathcal{A}$ picks option E_2.

In case $\mathcal{A}$ picks E_3 then $Q_i^{(\Delta)} = (y_i^{(\Delta)} + z_i^{(\Delta)})$ and $\tilde{Y}_i = \{y_i^{(\Delta)}\}, \tilde{Z}_i = \{z_i^{(\Delta)}\}$. It is then easily checked that $\mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(Q^{(\Delta)}|_{\rho_i^{(\Delta)}}) = 1$. This completes the induction and proves the lemma. $\square$

4 The Main Result

The main result of this paper is the following.

Theorem 13. *Let $m, \Delta \in \mathbb{N}$ be growing parameters with $\Delta = m^{o(1)}$.[11] Assume that the polynomials $P^{(\Delta)}(X^{(\Delta)})$ are as defined in Section 3.1. Then any multilinear circuit C of product-depth Δ computing $P^{(\Delta)}$ must have a size of at least $\exp(m^{\Omega(1)}) = \exp(n_\Delta^{\Omega(1/\Delta)})$.*

The following corollary is immediate from the lower bound in Theorem 13 and the formula upper bound in Proposition 11.

Corollary 14. *Assume $\Delta = \Delta(n) = o(\log n / \log\log n)$. For all large enough $n \in \mathbb{N}$, there is an explicit multilinear polynomial on n variables that has a multilinear formula of size $O(n)$ and product-depth $\Delta(n) + 1$ but no multilinear circuit of size at most $\exp(n^{\Omega(1/\Delta)})$ and product-depth at most $\Delta(n)$.*

Proof. We can find $m = m(n)$ so that that $\Delta = m^{o(1)}$ and the corresponding $n_\Delta \in [\sqrt{n}, n]$ for large enough n. We now apply Proposition 11 and Theorem 13 to obtain the result. $\square$

We also have a similar result for depth instead of product-depth.

Corollary 15. *Assume $\Delta = \Delta(n) = o(\log n / \log\log n)$. For all large enough $n \in \mathbb{N}$, there is an explicit multilinear polynomial on n variables that has a multilinear formula of size n and depth $\Delta + 1$ but no multilinear circuit of size $\exp(n^{\Omega(1/\Delta)})$ and depth at most Δ.*

Proof. Let $\Delta' = \lfloor (\Delta+1)/2 \rfloor$.

Fix $m = m(n)$ so that that $\Delta = m^{o(1)}$ and the corresponding $n_{\Delta'} \in [\sqrt{n}, n/2]$ for large enough n. We define the explicit polynomial to be either $P^{(\Delta')}$ or the sum of two copies of $P^{(\Delta'-1)}$ depending on whether Δ is even or odd.

Assume that Δ is even. Then, $\Delta = 2\Delta'$. In this case, the explicit polynomial is $P^{(\Delta')}$ which has a $(\Pi\Sigma)^{\Delta'}\Pi$ formula of size $O(n_{\Delta'}) = O(n)$ by Proposition 11. Note that a $(\Pi\Sigma)^{\Delta'}\Pi$ formula is of depth $2\Delta' + 1 = \Delta + 1$. This gives the upper bound.

---

[11]Since $n_\Delta = O(m)^\Delta$, this is equivalent to requiring that $\Delta = o(\log n_\Delta / \log\log n_\Delta)$.

For the lower bound, we use Theorem 13. Any circuit C of size at most s and depth at most $\Delta = 2\Delta'$ can be converted to one of size at most s and product-depth at most Δ' as follows. If C contains two $\times$-gates Ψ_1 and Ψ_2 where Ψ_1 feeds into Ψ_2, we merge Ψ_1 and Ψ_2. Repeated applications of this procedure yields a circuit of depth at most $2\Delta'$ in which no input-to-output path can contain consecutive $\times$-gates. This circuit must have product-depth at most Δ'. Clearly, this process can only reduce the size of the circuit. By Theorem 13, we see that if C computes $P^{(\Delta')}$ it must have a size of at least $\exp(n_{\Delta'}^{\Omega(1/\Delta')}) = \exp(n^{\Omega(1/\Delta)})$.

Now consider the case when Δ is odd. Then $\Delta = 2\Delta' - 1$. In this case, we define the explicit polynomial to be $Q = P^{(\Delta'-1)}(X_1) + P^{(\Delta'-1)}(X_2)$ where X_1 and X_2 are disjoint copies of $X^{(\Delta'-1)}$. Note that the number of variables in Q is $2n_{\Delta'-1} \leq n_{\Delta'} \leq n$ and by Proposition 11, Q has a $\Sigma(\Pi\Sigma)^{\Delta'-1}\Pi$ formula of size $O(n_{\Delta'-1}) = O(n)$. Note that a $\Sigma(\Pi\Sigma)^{\Delta'-1}\Pi$ formula is of depth $2(\Delta'-1) + 2 = \Delta + 1$. This gives the upper bound.

For the lower bound, we proceed as was done in the case where Δ is even. Like before, we can assume that the most efficient circuit C of depth Δ for Q has the property that no path in C contains consecutive $\times$-gates. Further, since Q is easily seen to be an irreducible polynomial, we can also assume that the output gate is a $+$-gate. This implies that the product-depth of C is at most $\lfloor \Delta/2 \rfloor \leq \Delta' - 1$. Applying Theorem 13 now yields the lower bound as in the even case. $\qquad\square$

Theorem 13 is proved by induction on the parameter Δ. For the purposes of induction, we need to prove a more technical statement from which we can easily infer Theorem 13. We give this technical statement below.

Recall that for a random mutlilinear (X, Y, Z)-restriction ρ, $\tilde{Y} = \mathrm{Img}(\rho) \cap Y$ and $\tilde{Z} = \mathrm{Img}(\rho) \cap Z$. Note that these are *random* sets. Now, given a multilinear polynomial $f \in \mathbb{F}[X' \cup Y' \cup Z']$ where $X' \subseteq X$ and Y', Z' are disjoint sets of variables, the polynomial $f|_\rho$ is a multilinear polynomial in $\mathbb{F}[\tilde{Y} \cup \tilde{Z} \cup Y' \cup Z']$.

Theorem 16. *Let $m, k, \Delta, M \in \mathbb{N}$ be growing positive integer parameters with $k = m^{0.05}$ and $M \geq m/10$. Let $\varepsilon > 0$ be such that $\varepsilon \geq 1/M^{0.25}$. Let $\Delta' \geq \Delta$ be such that $\Delta' \leq m^{0.001}$.*

Let $X_{(\omega_1, j_1)}^{(\Delta')}, \ldots, X_{(\omega_M, j_M)}^{(\Delta')}$ be arbitrary distinct $X^{(\Delta)}$-segments. Let $X = \bigcup_{i \in [M]} X_{(\omega_i, j_i)}^{(\Delta')}, Y = \bigcup_{i \in [M]} Y_{(\omega_i, j_i)}^{(\Delta')},$ and $Z = \bigcup_{i \in [M]} Z_{(\omega_i, j_i)}^{(\Delta')}.$

Assume that $X' = \bigcup_{i \in [M]} X_i'$ where for each $i \in [M]$, $X_i' \subseteq X_{(\omega_i, j_i)}^{(\Delta')}$ satisfying $|X_i'| \geq \varepsilon \cdot |X_{(\omega_i, j_i)}^{(\Delta')}|$. Let F be any $(\Sigma\Pi)^\Delta\Sigma$ syntactially multilinear variable-labelled[12] formula with $\mathrm{Vars}(F) = X' \dot\cup Y' \dot\cup Z'$ where Y' and Z' are arbitrary sets of variables that are disjoint from X'. Assume that the size of F is $s \leq \exp(k^{0.1}/\Delta^2)$.

For each $i \in [M]$, let ρ_i be an independent multilinear random restriction obtained by using the sampling algorithm $\mathcal{A}$ described in Section 3.2 to sample a $(X_{(\omega_i, j_i)}^{(\Delta')}, Y_{(\omega_i, j_i)}^{(\Delta')}, Z_{(\omega_i, j_i)}^{(\Delta')})$-restriction. Let $\rho = \rho_1 \circ \cdots \circ \rho_M$ be the resulting (X, Y, Z)-random restriction. Let $\tilde{Y}' = \rho(X') \cap Y$ and $\tilde{Z}' = \rho(X') \cap Z$.[13]

Then, we have

$$\Pr_\rho[\mathrm{relrk}_{(Y' \cup \tilde{Y}', Z' \cup \tilde{Z}')}(F|_\rho) \geq \exp(-k^{0.1}/\Delta)] \leq \Delta \exp(-k^{0.1}/\Delta).$$

Remark 17. *Note that the $X^{(\Delta)}$-segments that appear in the statement of Theorem 16 are $X^{(\Delta)}$-segments contained in $X^{(\Delta')}$ for some $\Delta' \geq \Delta$. This general form of the theorem is needed*

---

[12]See Section 2.4 for the definition of variable-labelled formulas.

[13] Note that these are *random* sets. Also note that for each fixing of ρ, the restricted formula $F|_\rho$ computes a multilinear polynomial in $\mathbb{F}[\tilde{Y}' \cup \tilde{Z}' \cup Y' \cup Z']$

when we are proving lower bounds for multilinear formulas of product-depth Δ'. During the course of the inductive proof, the depth of the circuit reduces iteratively and at some intermediate point in the proof when the depth is down to Δ, we end up in the situation described in Theorem 16.

We postpone the proof of the above theorem to Section 5, and deduce Theorem 13 from it below.

Proof of Theorem 13. Let C be any multilinear circuit of product-depth at most Δ computing $P^{(\Delta)}$. Let s denote the size of C. By Lemma 4, there is a $(\Sigma\Pi)^{\Delta}\Sigma$ syntactic multilinear formula F of size $s' = s^{O(\Delta)}$ computing $P^{(\Delta)}$. We can assume that F does not use any variables outside $X^{(\Delta)}$ since such variables may be safely set to 0 without affecting the polynomial being computed.

Recall that $X^{(\Delta)}$ is the disjoint union of its segments $X_1^{(\Delta)}, \ldots, X_m^{(\Delta)}$. The random restriction $\rho^{(\Delta)}$ from Section 3.2 is defined to be $\rho^{(\Delta)} = \rho_1^{(\Delta)} \circ \cdots \circ \rho_m^{(\Delta)}$ where each $\rho_i^{(\Delta)}$ is an independent $(X_i^{(\Delta)}, Y_i^{(\Delta)}, Z_i^{(\Delta)})$-restriction sampled using the algorithm $\mathcal{A}$. Let $\tilde{Y}$ and $\tilde{Z}$ denote $\mathrm{Img}(\rho^{(\Delta)}) \cap (\bigcup_i Y_i^{(\Delta)})$ and $\mathrm{Img}(\rho^{(\Delta)}) \cap (\bigcup_i Z_i^{(\Delta)})$ respectively.

Consider the restricted polynomials $P' = P^{(\Delta)}|_\rho$ and $F' = F|_\rho$. By Lemma 12, we know that $\mathrm{relrk}_{(\tilde{Y}, \tilde{Z})}(P') = 1$ with probability 1. On the other hand, applying Theorem 16 with $M = m$, $\varepsilon = 1$, $\Delta' = \Delta = m^{o(1)}$ and $Y' = Z' = \emptyset$, we get

$$\Pr_\rho[\mathrm{relrk}_{(\tilde{Y}, \tilde{Z})}(F') \geq \exp(-k^{0.1}/\Delta)] \leq \Delta \exp(-k^{-0.1}/\Delta) = o(1)$$

if $s' \leq \exp(k^{0.1}/\Delta^2)$. The final inequality above follows from the fact that $\Delta \leq m^{o(1)}$ and hence $\Delta \exp(-k^{-0.1}/\Delta) = \Delta \exp(-m^{\Omega(1)}/\Delta) = o(1)$.

Since the formula F' computes P', we must therefore have $s' > \exp(k^{0.1}/\Delta^2) = \exp(m^{\Omega(1)}/\Delta^2)$ which yields $s = \exp(m^{\Omega(1)}/\Delta^3)$. Since $\Delta = m^{o(1)}$, this means that $s = \exp(m^{\Omega(1)}) = \exp((n_\Delta)^{\Omega(1/\Delta)})$. □

5 Proof of Theorem 16

The proof of Theorem 16 will be by induction on the product-depth Δ. The base case $\Delta = 1$ is handled in Section 5.1.

For the induction case $\Delta > 1$, we proceed as follows. Let F be any formula as in the statement of Theorem 16. We can write

$$F = \sum_{i \in [s]} F_i \tag{3}$$

where each F_i is a $\Pi(\Sigma\Pi)^{\Delta-1}\Sigma$ variable-labelled syntactic multilinear formula of size at most s. Further, note (see Section 2.4) that $\mathrm{Vars}(F_i) = \mathrm{Vars}(F)$ for each i by the property of the variable-labelling $\mathrm{Vars}(\cdot)$.

We claim that it suffices to show the following. For any $\Pi(\Sigma\Pi)^{\Delta-1}\Sigma$ variable-labelled syntactic multilinear formula F' with $\mathrm{Vars}(F') = X' \cup Y' \cup Z'$ of size at most $s \leq \exp(k^{0.1}/\Delta^2)$, we have

$$\Pr_\rho[\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) \geq \exp(-k^{0.1}/(\Delta-1))] \leq \Delta \exp(-k^{0.1}/(\Delta-1)). \tag{4}$$

Assuming (4) for now, we proceed as follows. Applying (4) to the formulas $F_1, \ldots, F_s$ and using a union bound, we have

$$\Pr_\rho[\exists i \in [s], \; \mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F_i|_\rho) \geq \exp(-k^{0.1}/(\Delta-1))] \leq \Delta \exp(-k^{0.1}/(\Delta-1)) \cdot s \leq \Delta \exp(-k^{0.1}/\Delta)$$

where for the last inequality, we have used the fact that $s \leq \exp(k^{0.1}/\Delta^2)$. Note that if $\text{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F_i|_\rho) < \exp(-k^{0.1}/(\Delta-1))$ for each $i \in [s]$, the subadditivity of relative rank (Propostion 2 item 2) implies that

$$\text{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F|_\rho) \leq s \cdot \exp(-k^{0.1}/(\Delta-1)) \leq \exp(-k^{0.1}/\Delta),$$

where for the last inequality we have again used $s \leq \exp(k^{0.1}/\Delta^2)$. We have thus shown that

$$\Pr_\rho[\text{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F|_\rho) \geq \exp(-k^{0.1}/\Delta)] \leq \Delta \exp(-k^{0.1}/\Delta)$$

which proves the theorem.

It remains to prove (4), which is the main technical part of the proof. Fix an F' as in (4) and assume that

$$F' = \prod_{i \in [t]} F_i' \tag{5}$$

where the F_i' are the constituent $(\Sigma\Pi)^{(\Delta-1)}\Sigma$ formulas of F'. Recall that the $\text{Vars}(F_i')$ $(i \in [t])$ partition $\text{Vars}(F') = X' \dot\cup Y' \dot\cup Z'$.

For $\delta > 0$, $i \in [t]$ and $j \in [M]$, we say that F_i' is δ-*heavy* in segment X_j if $|\text{Vars}(F_i') \cap X_j| \geq \delta \cdot |X_j|$. We say that X_j is δ-*shattered* if there is no $i \in [t]$ such that F_i' is δ-heavy w.r.t. X_j.

The proof of (4) breaks down into the following three cases. Roughly, in case 1, we can prove (4) directly, whereas in cases 2 and 3, we appeal to the inductive hypothesis.

- **Case 1:** At least $M/2$ segments are $(1/4)$-shattered and further, no F_i' is (ε/k)-heavy in at least M/k segments X_j.

- **Case 2:** At least $M/2$ segments are *not* $(1/4)$-shattered.

- **Case 3:** There is some F_i' $(i \in [t])$ that is (ε/k)-heavy in at least M/k many segments X_j.

We show (4) in each of the above cases in Section 5.2 below. This will conclude the proof of the theorem.

Notation. We now define some notation that will be useful in the remainder of the theorem.

- For brevity, we use X_i, Y_i, and Z_i $(i \in [M])$ to denote the $X^{(\Delta)}$-segment $X^{(\Delta')}_{(\omega_i, j_i)}$, the $Y^{(\Delta)}$-segment $Y^{(\Delta')}_{(\omega_i, j_i)}$ and the $Z^{(\Delta)}$-segment $Z^{(\Delta')}_{(\omega_i, j_i)}$ respectively. The corresponding half-segments of X_i, Y_i and Z_i are denoted $X_{i,j}, Y_{i,j}$ and $Z_{i,j}$ respectively (for $j \in [2]$). Further $X^{(\Delta-1)}$-segments of $X_{i,j}$ are denoted $X_{i,j,p}$ $(p \in [m])$ and similarly we also have $Y_{i,j,p}$ and $Z_{i,j,p}$.

- For each $i \in [M]$, let y_i, z_i denote the variables $y^{(\Delta')}_{(\omega_i, j_i)}$ and $z^{(\Delta')}_{(\omega_i, j_i)}$ in the sets Y_i and Z_i respectively. Recall (see Section 3.1) that we have $Y_i = Y_{i,1} \cup Y_{i,2} \cup \{y_i\}$ and $Z = Z_{i,1} \cup Z_{i,2} \cup \{z_i\}$.

- For each $i \in [M]$, let $\tilde{Y}_i$ and $\tilde{Z}_i$ denote the random sets $\rho_i(X_i) \cap Y_i$ and $\rho_i(X_i) \cap Z_i$ respectively. Let $\tilde{Y}$ and $\tilde{Z}$ denote $\bigcup_{i \in [M]} \tilde{Y}_i$ and $\tilde{Z} = \bigcup_{i \in [M]} \tilde{Z}_i$. Also, let $\tilde{Y}_i'$ and $\tilde{Z}_i'$ denote the random sets $\rho_i(X_i') \cap Y_i$ and $\rho_i(X_i') \cap Z_i$ respectively.

5.1 The base case of Theorem 16

Let F be a $\Sigma\Pi\Sigma$ syntactic multilinear formula over the variable set $X' \cup Y' \cup Z'$. We can write $F = T_1 + \cdots + T_s$ where each T_i is a $\Pi\Sigma$ syntactic multilinear formula. We claim that for any $\Pi\Sigma$ syntactic multilinear formula T, we have

$$\Pr_\rho[\mathrm{relrk}_{(\tilde{Y}'\cup Y', \tilde{Z}'\cup Z')}(T|_\rho) \geq \exp(-2k^{0.1})] \leq \exp(-2k^{0.1}). \tag{6}$$

Assuming (6) we are done, since for any fixing of the random restriction ρ (which is a copy of the restriction $\rho^{(1)}$ from Section 3.2), we have by Proposition 2 Item 2 that

$$\mathrm{relrk}_{(\tilde{Y}'\cup Y', \tilde{Z}'\cup Z')}(F|_\rho) \leq \sum_{i=1}^{s} \mathrm{relrk}_{(\tilde{Y}'\cup Y', \tilde{Z}'\cup Z')}(T_i|_\rho) \leq s \cdot \max_i \left(\mathrm{relrk}_{(\tilde{Y}'\cup Y', \tilde{Z}'\cup Z')}(T_i|_\rho) \right)$$
$$\leq \exp(k^{0.1}) \cdot \max_i \left(\mathrm{relrk}_{(\tilde{Y}'\cup Y', \tilde{Z}'\cup Z')}(T_i|_\rho) \right)$$

and hence

$$\Pr_\rho[\mathrm{relrk}_{(\tilde{Y}'\cup Y', \tilde{Z}'\cup Z')}(F|_\rho) \geq \exp(-k^{0.1})] \leq \Pr_\rho[\exists i \ \mathrm{relrk}_{(\tilde{Y}'\cup Y', \tilde{Z}'\cup Z')}(T_i|_\rho) \geq \exp(-2k^{0.1})]$$
$$\leq \frac{s}{\exp(2k^{0.1})} \leq \exp(-k^{0.1})$$

where we have used (6) and a union bound for the second inequality.

It remains to prove (6). To see this, we proceed as follows. Assume that the output $\times$-gate of T is Φ and let $\Psi_1, \ldots, \Psi_r$ be the $+$-gates feeding into it and let $L_1, \ldots, L_r$ be the linear functions computed by these gates. Recall from Proposition 3 that we can assign to each Ψ_i a variable set $\mathrm{Vars}(\Psi_i) \subseteq X' \cup Y' \cup Z'$ such that $\{\mathrm{Vars}(\Psi_i) \mid i \in [r]\}$ induces a partition of $X' \cup Y' \cup Z'$ and further $\mathrm{Vars}(\Psi_i)$ contains $\mathrm{Supp}(\Psi_i)$.[14] Henceforth, we use $\mathrm{Vars}(L_i)$ to denote $\mathrm{Vars}(\Psi_i)$ for each $i \in [r]$.

We divide the gates $\Psi_1, \ldots, \Psi_r$ into two classes: those gates such that $\mathrm{Vars}(L_i) \cap X' \neq \emptyset$ and those for which $\mathrm{Vars}(L_i) \cap X' = \emptyset$; let t denote the number of gates of the former kind. Without loss of generality, we can assume that $\Psi_1, \ldots, \Psi_t$ are the gates such that $\mathrm{Vars}(L_i) \cap X' \neq \emptyset$ and $\Psi_{t+1}, \ldots, \Psi_r$ the rest.

We write the polynomial T as $T = L_1 \cdots L_t \cdot Q'$ where $Q' = \prod_{i>t} L_i$. For each $i \in [t]$ and $j \in [M]$, let $\hat{Y}_{i,j}$ and $\hat{Z}_{i,j}$ denote the random sets $\rho(\mathrm{Vars}(L_i) \cap X'_j) \cap Y$ and $\rho(\mathrm{Vars}(L_i) \cap X'_j) \cap Z$ respectively. We also denote by $\hat{Y}_{i,0}$ and $\hat{Z}_{i,0}$ the (non-random) sets $\mathrm{Vars}(L_i) \cap Y'$ and $\mathrm{Vars}(L_i) \cap Z'$. Let us define $\hat{Y}_i = \bigcup_{j \in [M]\cup\{0\}} \hat{Y}_{i,j}$ and $\hat{Z}_i = \bigcup_{j \in [M]\cup\{0\}} \hat{Z}_{i,j}$. Finally, let $\hat{Y}_{t+1}$ and $\hat{Z}_{t+1}$ denote the (non-random) sets $\bigcup_{i>t} \mathrm{Vars}(L_i) \cap Y'$ and $\bigcup_{i>t} \mathrm{Vars}(L_i) \cap Z'$.

Note that the sets $\tilde{Y}' \cup Y'$ can be partitioned as $Y' \cup \bigcup_{j \in [M]} \tilde{Y}'_j$ and also as $\bigcup_{i \in [t+1]} \hat{Y}_i = \left(\bigcup_{i \in [t], j \in [M]\cup\{0\}} \hat{Y}_{i,j} \right) \cup \hat{Y}_{t+1}$. A similar statement is true for $\tilde{Z}' \cup Z'$ as well.

Upon the application of the random restriction ρ (as described in Section 3.2), each L_i ($i \in [t]$) restricts to a linear function $L_i|_\rho = \tilde{L}_i \in \mathbb{F}[\hat{Y}_i \cup \hat{Z}_i]$, while the polynomial Q' is unaffected by the restriction ρ. Recall that ρ is the composition of *independent* restrictions $\rho_1, \ldots, \rho_m$ where each ρ_j only affects variables in X_j, for all $j \in [m]$ and the ρ_j's are independent copies of the random restriction ρ (defined in Section 3.2).

---

[14]$\mathrm{Supp}(\Psi_i)$ is the set of variables that actually appear in the subformula rooted at Ψ_i (cf. Section 2.4).

By Proposition 2 items 3 and 1, we know that for any choice of ρ

$$\text{relrk}_{(Y' \cup \tilde{Y}', Z' \cup \tilde{Z}')}(T|_\rho) = \left(\prod_{i \in [t]} \text{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho) \right) \cdot \text{relrk}_{(\hat{Y}_{t+1}, \hat{Z}_{t+1})}(Q')$$

$$\leq \prod_{i \in [t]} \text{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho). \tag{7}$$

To bound $\text{relrk}_{(Y' \cup \tilde{Y}', Z' \cup \tilde{Z}')}(T|_\rho)$, we use (7) along with a case analysis depending on the value of t.

Before we do the case analysis, we will present two technical lemmas which will be useful.

Lemma 18. *For $\tilde{Y}', \tilde{Z}'$ defined as in the statement of Theorem 16, $\Pr_\rho[|\tilde{Y}'| + |\tilde{Z}'| \leq M/200] \leq \exp(-\Omega(M))$.*

Proof. For $j \in [M]$, let v_j be a $\{0,1\}$-valued random variable which takes value 1 if and only if $\tilde{Y}'_j \cup \tilde{Z}'_j \neq \emptyset$. This is a random variable which only depends on ρ_j for each $j \in [M]$, i.e., v_js are independent random variables. Let $v = \sum_j v_j$. Note that $v \leq |\tilde{Y}'| + |\tilde{Z}'|$.

We first claim that for each j, $\Pr_{\rho_j}[\tilde{Y}'_j \cup \tilde{Z}'_j \neq \emptyset]$ is at least $1/48$. To see this, first note that $|X'_j| \geq \varepsilon|X_j| > 0$ (the non-emptiness of $X'_j \subseteq X_j$ is all we will need). Fix any variable $x \in X'_j$. From the description of the sampling algorithm $\mathcal{A}$ for the random restriction ρ, it easily follows that the probability with which ρ_j sets x to a variable in $Y_j \cup Z_j$ is at least $1/48$.[15] Hence, $\mathbf{E}_{\rho_j}[v_j] \geq 1/48$. By linearity of expectation, we get that $\mathbf{E}_\rho[v] \geq M/48$.

Using the Chernoff bound (Theorem 5) we get that $\Pr_\rho[v \leq M/200] \leq \exp(-\Omega(M))$. As $v \leq |\tilde{Y}'| + |\tilde{Z}'|$, we get that $\Pr_\rho[|\tilde{Y}'| + |\tilde{Z}'| \leq M/200] \leq \exp(-\Omega(M))$. $\square$

Lemma 19. *For each $i \in [t]$, $\Pr_\rho[\text{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho) \leq 1/\sqrt{2}] \geq 1/1000$.*

Proof. Fix an L_i for $i \in [t]$. Let us consider the smallest j_0 such that $X'_{j_0} \cap \text{Vars}(L_i) \neq \emptyset$. We consider two possibilities for the relationship between X_{j_0} and $\text{Vars}(L_i)$: either $\text{Vars}(L_i) \cap X'_{j_0} \subsetneq X_{j_0}$ or $X_{j_0} \subseteq \text{Vars}(L_i)$.

We fix ρ_j for all $j \in [M]$ such that $j \neq j_0$. Let $\hat{Y}'_i = \cup_{j \neq j_0} \hat{Y}_{i,j}$ and $\hat{Z}'_i = \cup_{j \neq j_0} \hat{Z}_{i,j}$.

Case I: $X_{j_0} \subseteq \text{Vars}(L_i)$. In this case, if the sampling step for ρ_{j_0} chooses options E_1 or E_2, then from the definition of $\rho^{(0)}$ we can see that $|\hat{Y}_{i,j}| = |\rho(X_{j_0}) \cap Y| = 2$ and similarly $|\hat{Z}_{i,j}| = 2$ as well. This implies that $|\hat{Y}_i|, |\hat{Z}_i| \geq 2$.

On the other hand, we also claim that $\text{rank}(M_{(\hat{Y}_i, \hat{Z}_i)})(L_i|_\rho)) \leq 2$. To see this, note that each $L_i|_\rho$ can be written as $L'_i(\hat{Y}_i) + L''_i(\hat{Z}_i)$ and for any polynomial Q that depends only on the variables in either $\hat{Y}_i$ or $\hat{Z}_i$, we have $\text{rank}(M_{(\hat{Y}_i, \hat{Z}_i)}(Q)) \leq 1$. The subadditivity of matrix rank now implies that $\text{rank}(M_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho)) \leq 2$.

In particular, we see that whenever option E_1 or E_2 is chosen, we have

$$\text{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho) \leq \frac{2}{2^{\frac{1}{2}(|\hat{Y}_i| + |\hat{Z}_i|)}} \leq \frac{1}{2}.$$

Since one of E_1 or E_2 is chosen with probability $2/3$, we get the desired bound in this case.

---

[15]W.l.o.g., say $x \in X_{j,1}$. Then $\mathcal{A}$ sets x to a variable whenever option E_3 is chosen (this happens with probability $1/3$) and further x is the variable chosen in $X_{j,1}$ to set to a variable (this happens with probability $1/16$).

Case II: $\mathrm{Vars}(L_i) \cap X'_{j_0} \subsetneq X_{j_0}$.   As mentioned earlier, we already fixed the restrictions ρ_j for $j \neq j_0$. Thus, the sets $\hat{Y}'_i$ and $\hat{Z}'_i$ have been fixed. Let $b \in \{0,1\}$. Suppose $|\hat{Y}'_i \cup \hat{Z}'_i| \equiv b \mod 2$. We will show that

$$\Pr_{\rho_{j_0}}[|\hat{Y}_{i,j_0} \cup \hat{Z}_{i,j_0}| = (1-b) \mod 2] \geq 1/1000. \tag{8}$$

Assuming (8) for now, we get that with probability at least $1/1000$, $|\hat{Y}_i \cup \hat{Z}_i| = |\hat{Y}_{i,j_0} \cup \hat{Z}_{i,j_0}| + |\hat{Y}'_i \cup \hat{Z}'_i|$ is odd. Therefore using Item 1 of Proposition 2 we are done. This concludes the proof of Lemma 19 assuming (8).

To see (8), let us assume that $b = 1$. (We only present the details for this case. The case of $b = 0$ is similar.) If $b = 1$, we need to show that $|\hat{Y}_{i,j_0} \cup \hat{Z}_{i,j_0}| = |\rho_{j_0}(\mathrm{Vars}(L_i) \cap X'_{j_0}) \cap (Y_{j_0} \cup Z_{j_0})|$ is even with probability $1/1000$. The following three possibilities arise:

(a) $\mathrm{Vars}(L_i) \cap X_{j_0,1} \neq \emptyset$ and $\mathrm{Vars}(L_i) \cap X_{j_0,2} \neq \emptyset$,

(b) $X_{j_0,1} \setminus \mathrm{Vars}(L_i) \neq \emptyset$ and $X_{j_0,2} \setminus \mathrm{Vars}(L_i) \neq \emptyset$, or

(c) $\mathrm{Vars}(L_i) \cap X_{j_0} = X_{j_0,1}$ or $\mathrm{Vars}(L_i) \cap X_{j_0} = X_{j_0,2}$.

In each case we will show that $|\hat{Y}_{i,j_0} \cup \hat{Z}_{i,j_0}|$ is even with probability at least $1/1000$. For (a), if ρ_{j_0} chooses option E_3, then with probability at least $1/8$ one of the variables in $\mathrm{Vars}(L_i) \cap X_{j_0,1}$ (resp. $\mathrm{Vars}(L_i) \cap X_{j_0,2}$) will be set to y_{j_0} (resp. z_{j_0}). The option E_3 is chosen with probability $1/3$. Therefore, with probability at least $1/(8 \cdot 8 \cdot 3) \geq 1/1000$, $|\hat{Y}_{i,j_0} \cup \hat{Z}_{i,j_0}| = 2$ and is thus even.

For (b), observe that if ρ_{j_0} chooses option E_3, then with probability at least $1/8$ one of the variables in $X_{j_0,1} \setminus \mathrm{Vars}(L_i)$ (resp. $X_{j_0,2} \setminus \mathrm{Vars}(L_i)$) will be set to y_{j_0} (resp. z_{j_0}) and all other variables in X_{j_0} are set to constants. As this implies that all variables in $\mathrm{Vars}(L_i)$ are set to constants, again with probability at least $1/(8 \cdot 8 \cdot 3)$, $|\hat{Y}_{i,j_0} \cup \hat{Z}_{i,j_0}| = 0$ and is thus even.

For (c), let us assume that $\mathrm{Vars}(L_i) \cap X_{j_0} = X_{j_0,1}$. In this case, if ρ_{j_0} chooses option E_1, then with probability 1, $|\hat{Y}_{i,j_0} \cup \hat{Z}_{i,j_0}| = 4$. The option E_1 is chosen with probability $1/3$. Similarly, if $\mathrm{Vars}(L_i) \cap X_{j_0} = X_{j_0,2}$ then if ρ_{j_0} chooses option E_2, we have $|\hat{Y}_{i,j_0} \cup \hat{Z}_{i,j_0}| = 4$. Again, option E_2 is chosen with probability $1/3$.

This finishes the proof of (8) in the case that $b = 1$. For the case $b = 0$, we employ a similar case analysis with the following cases.

(a) $\mathrm{Vars}(L_i) \cap X_{j_0,1} \neq \emptyset$ and $X_{j_0,2} \setminus \mathrm{Vars}(L_i) \neq \emptyset$,

(b) $X_{j_0,1} \setminus \mathrm{Vars}(L_i) \neq \emptyset$ and $\mathrm{Vars}(L_i) \cap X_{j_0,2} \neq \emptyset$.

The proof is similar and left to the reader. $\qquad \square$

We now proceed to the proof of (6), which will finish the proof of the base case of Theorem 16. Let us recall that t is the number of $+$-gates such that for all $i \in [t]$, $\mathrm{Vars}(L_i) \cap X' \neq \emptyset$.

$\underline{t \leq M/500 \text{ (Case 1)}}$: This case is quite straightforward. As noted in the proof of Case I of Lemma 19, $\mathrm{rank}(M_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho)) \leq 2$.

In particular, the above yields

$$\mathrm{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho) \leq \frac{2}{2^{\frac{1}{2}(|\hat{Y}_i|+|\hat{Z}_i|)}}$$

and hence using (7), we have

$$\mathrm{relrk}_{(Y' \cup \tilde{Y}', Z' \cup \tilde{Z}')}(T|_\rho) \leq \frac{2^t}{2^{\frac{1}{2}(|\tilde{Y}' \cup Y'|+|\tilde{Z}' \cup Z'|)}} \leq \frac{2^t}{2^{\frac{1}{2}(|\tilde{Y}'|+|\tilde{Z}'|)}}.$$

Using Lemma 18, we know that $|\tilde{Y}'| + |\tilde{Z}'| \leq M/200$ with probability at most $\exp(-\Omega(M))$. Therefore, with all but $\exp(-\Omega(M))$ probability,

$$\text{relrk}_{(Y' \cup \tilde{Y}', Z' \cup \tilde{Z}')}(T) \leq \frac{2^{M/500}}{2^{M/400}} \leq \exp(-2k^{0.1}).$$

Moreover, for our setting of parameters of M and k, $\exp(-\Omega(M)) \leq \exp(-2k^{0.1})$. This proves (6) in the case where $t \leq M/500$.

$\underline{t > M/500 \text{ (Case 2)}:}$ For each $i \in [t]$, we define a Bernoulli random variable V_i as follows: $V_i = 1$ if $\text{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho) \leq 1/\sqrt{2}$ and it is 0 otherwise. By Lemma 19, we know that $\mathbf{E}[V_i] \geq 1/1000$. Let $V = \sum_{i \in [t]} V_i$. Then by linearity of expectation, $\mathbf{E}[V] \geq t/1000$.

Recall that $|X_j| = 16$ for every $j \in [M]$. The multilinearity of the formula T implies that for any $i \neq i'$, $\text{Vars}(L_i) \cap \text{Vars}(L_{i'}) = \emptyset$. Therefore, for each segment X_j the number of L_i such that $\text{Vars}(L_i) \cap X_j \neq \emptyset$ is at most 16. This implies that the Bernoulli random variables V_is are read-16, when viewed as functions of the independent random restrictions $\rho_1, \ldots, \rho_M$. By the read-k Chernoff bound (Theorem 9),[16] we thus obtain

$$\Pr[V < t/2000] \leq \exp(-\Omega(t)) \leq \exp(-\Omega(M)) \leq \exp(-2k^{0.1}).$$

When $V \geq t/2000$, i.e. when at least $t/2000$ many L_i are such that $\text{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho) \leq 1/\sqrt{2}$, we have by (7),

$$\text{relrk}_{(Y' \cup \tilde{Y}, Z' \cup \tilde{Z})}(T) \leq \prod_{i \in [t]} \text{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(L_i|_\rho) \leq \frac{1}{(\sqrt{2})^{t/2000}} \leq \exp(-2k^{0.1}).$$

This proves (6) in the case where $t > M/500$. This completes the proof of the base case.

5.2 The Inductive Case: Proof of (4) in Cases 1-3

The following piece of notation will be useful for the inductive case.

For each $i \in [t]$ and $j \in [M]$, let $\hat{Y}_{i,j}$ and $\hat{Z}_{i,j}$ denote the random sets $\rho(\text{Vars}(F_i') \cap X_j') \cap Y$ and $\rho(\text{Vars}(F_i') \cap X_j') \cap Z$ respectively. We also denote by $\hat{Y}_{i,0}$ and $\hat{Z}_{i,0}$ the (non-random) sets $\text{Vars}(F_i') \cap Y'$ and $\text{Vars}(F_i') \cap Z'$. Let us define $\hat{Y}_i = \bigcup_{j \in [M] \cup \{0\}} \hat{Y}_{i,j}$ and $\hat{Z}_i = \bigcup_{j \in [M] \cup \{0\}} \hat{Z}_{i,j}$.

Note that the sets $\tilde{Y}' \cup Y'$ can be partitioned as $Y' \cup \bigcup_{j \in [M]} \tilde{Y}_j'$ and also as $\bigcup_{i \in [t]} \hat{Y}_i = \bigcup_{i \in [t], j \in [M] \cup \{0\}} \hat{Y}_{i,j}$. A similar statement is true for $\tilde{Z}' \cup Z'$ as well.

5.2.1 Case 1

By renaming the segments if necessary, we assume that $X_1, \ldots, X_{M/2}$ are $(1/4)$-shattered. For each $j \leq M/2$, the restriction ρ_j is sampled according to the algorithm $\mathcal{A}$ described in Section 3.2. We say that a ρ_j is *good* if the algorithm $\mathcal{A}$ decides on option E_3 in sampling ρ_j. For each j, ρ_j is good with probability $1/3$. Let $\mathcal{E}_1$ denote the event that the number of good ρ_j is at most $M/8$. Since different ρ_j's are sampled independently, a Chernoff bound (Theorem 5 Item 1) tells us that

$$\Pr_\rho[\mathcal{E}_1] \leq \exp(-\Omega(M)). \tag{9}$$

---

[16]The use of the read-k Chernoff bound here can be easily circumvented by showing that in fact a constant-fraction of the V_i ($i \in [t]$) are in fact completely independent and the standard Chernoff bound can be applied to their sum. We leave the details to the interested reader.

For each $j \in [M/2]$, we condition on whether or not ρ_j is good. By (9), the probability that $\mathcal{E}_1$ occurs is $\exp(-\Omega(M))$.

Assume that the event $\mathcal{E}_1$ does not occur. By renaming segments once more, we may assume that ρ_j is good for each $j \in [M/8]$. We condition on any choice of the restrictions ρ_j for $j > M/8$; this fixes the sets $\tilde{Y}'_j$ and $\tilde{Z}'_j$ defined above for $j > M/8$.

We now observe the following from the description of the sampling algorithm $\mathcal{A}$, specifically option E_3 of $\mathcal{A}$. Conditioned on our choices so far, each ρ_j for $j \in [M/8]$ is now a random $(X_j, \{y_j\}, \{z_j\})$-restriction. For clarity, we call this conditioned random restriction ρ'_j.

We now proceed to analyzing $\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho)$. Since $\tilde{Y}' \cup Y'$ (resp. $\tilde{Z}' \cup Z'$) can be partitioned as $\bigcup_{i \in [t]} \hat{Y}_i$ (resp. $\bigcup_{i \in [t]} \hat{Z}_i$), we know (Proposition 2 Item 3) that for any choice of ρ,

$$\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) = \prod_{i \in [t]} \mathrm{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(F'_i|_\rho). \tag{10}$$

So we analyze $\mathrm{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(F'_i|_\rho)$ for each $i \in [t]$. To bound this quantity, we recall (Proposition 2 Item 1) that $\mathrm{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(F'_i|_\rho)$ is always at most 1. Further, if $|\hat{Y}_i \cup \hat{Z}_i|$ is odd, then $\mathrm{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(F'_i|_\rho) \leq 1/\sqrt{2}$. This motivates what follows.

For any $i \in [t]$ and $j \in \{0, \ldots, M\}$, let $\alpha_{i,j} \in \{0,1\}$ be the random variable defined by $\alpha_{i,j} \equiv |\hat{Y}_{i,j} \cup \hat{Z}_{i,j}| \pmod 2$. Note that the variables $\alpha_{i,0}$ and $\alpha_{i,j}$ for $j > M/8$ are actually fixed. Define $\tilde{\alpha}_i \in \{0,1\}$ by $\tilde{\alpha}_i = \bigoplus_{j=0}^{M} \alpha_{i,j}$. Observe that $\tilde{\alpha}_i = 1$ if and only if $|\hat{Y}_i \cup \hat{Z}_i|$ is odd. In particular, by Proposition 2 (item 3), we have $\mathrm{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(F'_i|_\rho) \leq 1/2^{\tilde{\alpha}_i/2}$.

Let Odd_ρ be the integer random variable defined by

$$\mathrm{Odd}_\rho = \sum_{i \in [t]} \tilde{\alpha}_i$$

where the sum is defined over $\mathbb{R}$. By the discussion above and (10), we know that

$$\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) \leq \frac{1}{2^{\mathrm{Odd}_\rho/2}}. \tag{11}$$

We will show below that

$$\Pr_{\rho'_1, \ldots, \rho'_{M/8}} [\mathrm{Odd}_\rho \leq 10 k^{0.1}] \leq \exp(-k^{0.1}). \tag{12}$$

The above, along with (11) implies that whenever the event $\mathcal{E}_1$ does not occur, we have the inequality

$$\Pr_\rho [\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) \geq \exp(-k^{0.1})] \leq \exp(-k^{0.1}).$$

In particular, using (9), we see that

$$\Pr_\rho [\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) \geq \exp(-k^{0.1})] \leq \exp(-k^{0.1}) + \exp(-\Omega(M)) \leq 2\exp(-k^{0.1})$$

which implies (4) and hence finishes the analysis of Case 1. For the last inequality above, we have used the fact $M \geq m/10 \geq k$ for our choice of parameters.

We now prove (12). To do this, we will actually prove a slightly different statement. For any $i \in [t]$, define $\alpha_i = \bigoplus_{j \in [M/8]} \alpha_{i,j}$. Note that $\alpha_i = \tilde{\alpha}_i \oplus \bigoplus_{j \notin [M/8]} \alpha_{i,j}$. For any $\beta_1, \ldots, \beta_t \in \{0,1\}$, we will show that

$$\Pr_{\rho'_1, \ldots, \rho'_{M/8}} [\forall i \in [t] \; \alpha_i = \beta_i] \leq \exp(-k^{0.15}). \tag{13}$$

26

Assuming the above, we obtain a similar statement for $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t$, since for any $\beta_1, \ldots, \beta_t \in \{0,1\}$,

$$\Pr_{\rho'_1, \ldots, \rho'_{M/8}}[\forall i \in [t] \ \tilde{\alpha}_i = \beta_i] = \Pr_{\rho'_1, \ldots, \rho'_{M/8}}[\forall i \in [t] \ \alpha_i = \beta_i \oplus \bigoplus_{j \notin [M/8]} \alpha_{i,j}] \leq \exp(-k^{0.15}).$$

(We have used above the fact that $\alpha_{i,j}$ is fixed for any $j \in \{0, \ldots, M\} \setminus [M/8]$.)

From the above inequality and using Proposition 6, we get

$$\Pr_{\rho'_1, \ldots, \rho'_{M/8}}[\mathrm{Odd}_\rho \leq 10k^{0.1}] \leq \exp(10k^{0.1} \log t - k^{0.15})$$

$$\leq \exp(10k^{0.1} k^{0.02+o(1)} - k^{0.15}) \leq \exp(-k^{0.1})$$

where for the second inequality, we used the fact that $t \leq n_{\Delta'} = O(m)^{\Delta'}$ and hence $\log t = O(\Delta' \log m) \leq m^{0.001+o(1)} = k^{0.02+o(1)}$ as $\Delta' \leq m^{0.001}$ and $m = k^{20}$. This finishes the proof of (12) modulo (13).

The proof of (13) is the main technical statement of this section. From now on, fix some $\beta_1, \ldots, \beta_t \in \{0,1\}$.

Proof outline of (13). From the description of the sampling algorithm $\mathcal{A}$ (specifically option E_3 of $\mathcal{A}$), we observe the following. To sample each ρ'_j, we choose independent random variables $x_{j,1}$ and $x_{j,2}$ uniformly from $X_{j,1}$ and $X_{j,2}$ respectively, and set $x_{j,1}$ to y_j and $x_{j,2}$ to z_j; all other variables in X_j are set deterministically to 0 or 1 according to some rule R (the exact rule R will not be relevant in this argument). We will view this sampling process iteratively via an algorithm $\mathcal{A}'$ described formally below.

Informally, at each step, $\mathcal{A}'$ chooses a suitable $I \subseteq [t]$ and asks for each $j \in [M/8]$ if either of the variables $x_{j,1}$ or $x_{j,2}$ belongs to the set of variables $\bigcup_{i \in I} \mathrm{Vars}(F'_i) \cap X'_j$. If this event does occur for any $j \in [M/8]$, then $\mathcal{A}'$ reveals the restriction ρ'_j entirely, and otherwise, it does not reveal anything else about ρ'_j in this step. In either case, we are able to entirely deduce the values of $\alpha_{i,j}$ for $i \in I$ from the above information about ρ'_j and hence we can also deduce $\alpha_i = \bigoplus_{j \in [M/8]} \alpha_{i,j}$ for each $i \in I$. We show that with reasonable probability, it holds that $\bigoplus_{i \in I} \alpha_i \neq \bigoplus_{i \in I} \beta_i$, which in particular implies that there must be an i such that $\alpha_i \neq \beta_i$. In this case, $\mathcal{A}'$ outputs SUCCESS. If not, the algorithm continues with another iteration of the same procedure. We will show that with high probability, $\mathcal{A}'$ can carry out many such iterations. If either $\mathcal{A}'$ does not find an i such that $\alpha_i \neq \beta_i$ after many iterations, or it cannot carry out too many iterations, then $\mathcal{A}'$ outputs FAILURE. We show that $\mathcal{A}'$ outputs SUCCESS with high probability, which will finish the proof of (13).

The algorithm $\mathcal{A}'$. The algorithm $\mathcal{A}'$ is a general sampling procedure that has the following input-output behaviour.

- **Input**: Sets $S \subseteq [M/8], T \subseteq [t]$, and fixed (i.e. not random) $(X_j, \{y_j\}, \{z_j\})$-restrictions $(\rho'_j)_{j \in [M/8] \setminus S}$. For each $j \in S$, we define $\bar{X}_j = (\bigcup_{i \in T} \mathrm{Vars}(F'_i)) \cap X_j$.

- **Desired Output**: The algorithm samples an independent random restriction ρ'_j for each $j \in S$ as follows. Independent and uniformly random variables $x_{j,1}$ and $x_{j,2}$ are chosen from sets $\bar{X}_{j,1} := \bar{X}_j \cap X_{j,1}$ and $\bar{X}_{j,2} := \bar{X}_j \cap X_{j,2}$ respectively. The variables $x_{j,1}$ and $x_{j,2}$ are set to y_j and z_j respectively, and the remaining variables in X_j are set deterministically to 0 or 1 in accordance with the rule R referenced above.

  Further, the algorithm either outputs SUCCESS or FAILURE, with the guarantee that when it outputs SUCCESS, then we must have $\alpha_i \neq \beta_i$ for some $i \in [t]$.

Note that the problem of sampling $\rho'_1, \ldots, \rho'_{M/8}$ is equivalent to running the algorithm $\mathcal{A}'$ with $S = [M/8]$ and $T = [t]$. In this case, $\bar{X}_j = X_j$ for each $j \in [M/8]$.

The formal description of the algorithm $\mathcal{A}'$ follows.

Algorithm $\mathcal{A}'(S, T, (\rho'_j)_{j \in [M/8] \setminus S})$

1. Find $I \subseteq T$ such that the following holds. Let $F'_I := \prod_{i \in I} F'_i$ and $\mathrm{Vars}(F'_I) := \bigcup_{i \in I} \mathrm{Vars}(F'_i)$.

   (a) For all $j \in S$, $|\mathrm{Vars}(F'_I) \cap \bar{X}_j| \leq (1/3) \cdot |\bar{X}_j|$.

   (b) $|\hat{S}| \leq M/k$, where $\hat{S} := \{j \in S \mid |\mathrm{Vars}(F'_I) \cap \bar{X}_j| > (2\varepsilon/k) \cdot |\bar{X}_j|\}$.

   (c) There is some $j \in S$ such that $|\mathrm{Vars}(F'_I) \cap \bar{X}_j| \geq (\varepsilon/2k) \cdot |\bar{X}_j|$.

   If there are more than one such I, choose the lexicographically least one. If there is no such I or if $S = \emptyset$, output FAILURE. Further, complete the sampling process as follows. For each $j \in S$, the restriction ρ'_j is sampled by choosing variables $x_{j,1}$ and $x_{j,2}$ independently and uniformly from $\bar{X}_{j,1}$ and $\bar{X}_{j,2}$ respectively and setting them to y_j and z_j respectively. Other variables in X_j are set according to the restriction rule R.

2. Initialize the set $S' = \emptyset$.

3. For each $j \in S$, define $\delta_{j,1} \in [0, 1]$ by

$$\delta_{j,1} = \frac{1}{|\bar{X}_{j,1}|} \cdot |\mathrm{Vars}(F'_I) \cap \bar{X}_{j,1}|$$

   and $\delta_{j,2}$ similarly. Let $\delta_j = |\mathrm{Vars}(F'_I) \cap \bar{X}_j|/|\bar{X}_j|$ and define $\delta = \sum_{j \in S} \delta_j$.

4. If $\delta < 1$, define $S'' = \{j \in S \mid \delta_j \geq 1/M^{0.5}\}$. If $\delta \geq 1$, define $S'' = \hat{S}$ where $\hat{S}$ is as defined in Step 1 above, i.e., $S'' = \{j \in S \mid \delta_j \geq (2\varepsilon/k)\}$.

5. For each $j \in S$, do the following independently.

   (a) Sample $b_{j,1} \in \{0, 1\}$ so that $b_{j,1} = 1$ w.p. $\delta_{j,1}$ and 0 otherwise. Similarly, sample $b_{j,2}$ independent of $b_{j,1}$ so that $b_{j,2} = 1$ w.p. $\delta_{j,2}$.

   (b) If either of $b_{j,1}$ or $b_{j,2}$ is 1, or $j \in S''$, do the following.

      i. Add j to the set S'.

      ii. If $b_{j,1} = 1$, sample $x_{j,1}$ uniformly from $\mathrm{Vars}(F'_I) \cap \bar{X}_{j,1}$ and set $x_{j,1}$ to y_j. If $b_{j,1} = 0$, sample $x_{j,1}$ uniformly from $\bar{X}_{j,1} \setminus \mathrm{Vars}(F'_I)$ and set $x_{j,1}$ to y_j.

      iii. Sample $x_{j,2}$ similarly from $\bar{X}_{j,2}$ and set $x_{j,2}$ to z_j.

      iv. Set all other variables in X_j deterministically in accordance with the rule R. (This fixes the $(X_j, \{y_j\}, \{z_j\})$-restriction ρ'_j.)

6. Compute the Boolean variables $\alpha'_{i,j} \in \{0, 1\}$ for each $i \in I$ and $j \in [M/8]$ as follows.

   (a) If $j \notin S$ or $j \in S'$, compute $\alpha'_{i,j}$ from ρ'_j using $\alpha'_{i,j} = |\hat{Y}_{i,j} \cup \hat{Z}_{i,j}| \pmod{2}$.

   (b) If $j \in S \setminus S'$, then set $\alpha'_{i,j} = 0$.

7. For each $i \in I$, let $\alpha'_i = \bigoplus_{j \in [M/8]} \alpha'_{i,j}$. If $\alpha'_i \neq \beta_i$ for any $i \in I$, sample the remaining ρ'_j ($j \in S \setminus S'$) as in Steps 5(b)(ii) and 5(b)(iii), and output SUCCESS.

8. Otherwise, run the algorithm $\mathcal{A}'$ on inputs $(S \setminus S', T \setminus I, (\rho'_j)_{j \notin (S \setminus S')})$.

28

Correctness. Here, we show that $\mathcal{A}'$, on any input S, T, and $(\rho_j')_{j \notin S}$ samples $(\rho_j')_{j \in S}$ according to the desired distribution. Moreover, we show that whenever $\mathcal{A}'$ outputs SUCCESS, it is indeed because the sampled restrictions imply that $\alpha_i \neq \beta_i$ for some $i \in [t]$.

We first argue that the sampled distribution is correct. Note that if the algorithm cannot find a suitable I in Step 1, then the restrictions sampled trivially have the correct distribution. So we assume that $\mathcal{A}$ does not output FAILURE in Step 1.

Now, note that to sample a uniformly random variable a from a finite set A, we may first fix any set $A' \subseteq A$ and sample a random bit $\beta \in \{0, 1\}$ that is 1 with probability $|A'|/|A|$ and depending on whether β is 1 or 0, sample a random element of A' or $A \setminus A'$. This describes how the sampling algorithm $\mathcal{A}'$ samples a random $x_{j,1}$ from $\bar{X}_{j,1}$ (the case of $\bar{X}_{j,2}$ is similar), where the role of the set A' is taken by $\bar{X}_{j,1} \cap \mathrm{Vars}(F_I')$ and the bit $b_{j,1}$ plays the role of β. If $j \in S'$, then the subsequent sampling takes place in Step 5(b), and for $j \in S \setminus S'$, the subsequent sampling takes place in either Step 7 (in case the algorithm outputs SUCCESS) or in a later iteration.

We now argue that an output of SUCCESS means that the algorithm has found an i such that $\alpha_i \neq \beta_i$. This is obvious once we argue that for each $i \in I$, the quantity α_i' computed by the algorithm is equal to the random variable α_i defined above. To argue this, it suffices to show that $\alpha_{i,j}'$ (computed in Step 6) equals $\alpha_{i,j}$ for each $i \in I, j \in [M/8]$. This is obvious for $j \notin S$ or $j \in S'$ from Step 6 and the definition of $\alpha_{i,j}$ above. For $j \in S \setminus S'$, we know that $b_{j,1} = b_{j,2} = 0$ and hence the sampled ρ_j' does not choose any variable from $\mathrm{Vars}(F_I')$ to be either $x_{j,1}$ or $x_{j,2}$. In particular, this implies that $x_{j,1} \notin \mathrm{Vars}(F_I')$ and $x_{j,2} \notin \mathrm{Vars}(F_i')$ for any $i \in I$ and hence, $\alpha_{i,j} = 0$. Thus, even in the case that $j \in S \setminus S'$, we have $\alpha_{i,j} = \alpha_{i,j}'$. This concludes the proof of correctness.

Probability of SUCCESS. We now argue that the algorithm $\mathcal{A}'$ outputs SUCCESS with high probability if started with the initial input of $S = S_0 := [M/8]$ and $T = T_0 := [t]$ (in which case $\bar{X}_j = X_j$ for each j and the input $(\rho_j')_{j \notin S}$ is trivial). This will prove (13).

Consider a run of the algorithm $\mathcal{A}'$ on the above inputs. Each such run can produce many successive iterations on different inputs $(S, T, (\rho_j')_{j \notin S})$. We say that a single iteration of $\mathcal{A}'$ is of Type I if the corresponding value of δ (computed in Step 3) is at least 1 and of Type II otherwise. Note that this is a *deterministic* function of the current inputs S and T.

To show that the algorithm $\mathcal{A}'$ outputs SUCCESS with high probability on (S_0, T_0), we show a more general statement. Call an input $(S, T, (\rho_j')_{j \notin S})$ an (a, b)-*good* input, where $a, b \in \mathbb{N}$, if the following holds.

- The input set S satisfies

$$|S| \geq \frac{M}{8} - \frac{10 \cdot a \cdot M}{k} - 10 \cdot b \cdot M^{0.5}, \tag{14}$$

- and for each $j \in S$, we have

$$|\bar{X}_j| \geq |X_j| \cdot \left(1 - \frac{2\varepsilon}{k} \cdot a - \frac{b}{M^{0.5}}\right). \tag{15}$$

We say an iteration of $\mathcal{A}'$ is (a, b)-good, if its input is (a, b)-good. Informally, we expect an iteration of $\mathcal{A}'$ to be (a, b)-good after at most a iterations of Type I and at most b iterations of Type II (see Claim 21 below).

The first observation is that an algorithm cannot output FAILURE on an (a, b)-good iteration for small a, b.

Define $a_0 = k/100$ and $b_0 = M^{0.45}$.

Claim 20. *The algorithm $\mathcal{A}'$ does not output FAILURE in any (a,b)-good iteration where $a < a_0$ and $b < b_0$.*

Proof. We only need to show that $S \neq \emptyset$ and that $\mathcal{A}'$ is able to find an $I \subseteq T$ with the required properties in Step 1.

Observe that in an (a,b)-good iteration for $a < a_0$ and $b < b_0$, we have by (14)

$$|S| \geq \frac{M}{8} - \frac{10 \cdot (k/100) \cdot M}{k} - 10 \cdot M^{0.45} \cdot M^{0.5} \geq \frac{M}{50} \tag{16}$$

and also for each $j \in S$, by (15)

$$|\bar{X}_j| \geq |X_j| \cdot \left(1 - \frac{2\varepsilon}{k} \cdot \frac{k}{100} - \frac{M^{0.45}}{M^{0.5}}\right) \geq (1 - o(1))|X_j|. \tag{17}$$

First consider the case that there is some $i_0 \in T$ and some $j_0 \in S$ such that $|\mathrm{Vars}(F'_{i_0}) \cap X_{j_0}| \geq (\varepsilon/2k)|X_{j_0}|$. In this case, we claim that the singleton set $I = \{i_0\}$ has the properties required in Step 1 of $\mathcal{A}'$. Property (a) follows from the fact that each X_j ($j \in [M/8]$) is $(1/4)$-shattered and hence for each $j \in [M/8]$

$$|\mathrm{Vars}(F'_{i_0}) \cap \bar{X}_j| = |\mathrm{Vars}(F'_{i_0}) \cap X_j| \leq |X_j|/4 \leq |\bar{X}_j|/3$$

where the last inequality uses (17). Property (c) follows from the choice of i_0, which implies

$$|\mathrm{Vars}(F'_{i_0}) \cap \bar{X}_{j_0}| = |\mathrm{Vars}(F'_{i_0}) \cap X_{j_0}| \geq (\varepsilon/2k) \cdot |X_{j_0}| \geq (\varepsilon/2k) \cdot |\bar{X}_{j_0}|.$$

Finally, to see Property (b), we note that if $|\mathrm{Vars}(F'_{i_0}) \cap \bar{X}_j| > (2\varepsilon/k)|\bar{X}_j|$ for some $j \in S$, then by (17), we also have

$$|\mathrm{Vars}(F'_{i_0}) \cap \bar{X}_j| = (2\varepsilon/k)|\bar{X}_j| > (\varepsilon/k)|X_j|.$$

In other words, F'_{i_0} is (ε/k)-heavy in X_j. However, the assumption in this section (corresponding to Case 1) is that each F'_i is (ε/k)-heavy in at most M/k many X_j. In particular, this shows that the number of j's such that $|\mathrm{Vars}(F'_{i_0}) \cap \bar{X}_j| > (2\varepsilon/k)|\bar{X}_j|$ is at most M/k and hence Property (b) is indeed satisfied. Thus, the singleton set $I = \{i_0\}$ has all the required properties.

From now onwards, we assume that there is no $i_0 \in T$ and $j_0 \in S$ satisfying the above. Thus, we have for each $i \in T$ and $j \in S$,

$$|\mathrm{Vars}(F'_i) \cap X_j| < (\varepsilon/2k)|X_j|. \tag{18}$$

From (18), we know in particular that for each $i \in T$, we have $|\mathrm{Vars}(F'_i) \cap \bar{X}_j| = |\mathrm{Vars}(F'_i) \cap X_j| \leq (\varepsilon/2k)|X_j| \leq (2\varepsilon/k)|\bar{X}_j|$. Thus, the set $\bar{X}_j$ is partitioned by the sets $(\mathrm{Vars}(F'_i) \cap \bar{X}_j)_{i \in T}$, each of relative size at most $(2\varepsilon/k)$.

Now, consider all sets $I \subseteq T$ such that for *some* $j \in S$, we have

$$|\mathrm{Vars}(F'_I) \cap \bar{X}_j| \geq (\varepsilon/2k) \cdot |\bar{X}_j|. \tag{19}$$

Clearly, since each $\bar{X}_j$ is partitioned by the sets $(\mathrm{Vars}(F'_i) \cap \bar{X}_j)_{i \in T}$, there do exist sets I satisfying (19). That is, any such set I satisfies property (c) mentioned in Step 1 of the algorithm $\mathcal{A}'$.

We fix such an I of the smallest possible size and claim that I also satisfies

$$|\mathrm{Vars}(F'_I) \cap \bar{X}_j| \leq (2\varepsilon/k) \cdot |\bar{X}_j| \tag{20}$$

for each $j \in S$. This will therefore prove that this I satisfies properties (a) and (b) in Step 1 of the algorithm $\mathcal{A}'$. (In fact for proving that (a) holds, it suffices to prove that $|\mathrm{Vars}(F'_I) \cap \bar{X}_j| \leq$

$1/3 \cdot |\bar{X}_j|$. Similarly, for proving (b), it suffices to prove (20) for all but M/k many $j \in S$. But we get better bounds in this case.)

To see this, we argue as follows. If $|I| = 1$, then $F'_I = F'_i$ for some $i \in T$. Then (20) follows from (18) and (17). So we may assume $|I| > 1$. In this case, as I is the smallest possible set satisfying (19), we must have $|\text{Vars}(F'_{I'}) \cap \bar{X}_j| < (\varepsilon/2k) \cdot |\bar{X}_j|$ for each $I' \subsetneq I$ and *each* $j \in S$. Thus, we have for any fixed partition of I into two disjoint sets I' and I'' and any $j \in S$,

$$|\text{Vars}(F'_I) \cap \bar{X}_j| \le |\text{Vars}(F'_{I'}) \cap \bar{X}_j| + |\text{Vars}(F'_{I''}) \cap \bar{X}_j| \le (\varepsilon/2k + \varepsilon/2k) \cdot |\bar{X}_j| \le (\varepsilon/k)|\bar{X}_j|$$

proving (20).

This shows that there are suitable sets I satisfying the properties required by Step 1 of the algorithm $\mathcal{A}'$ and hence $\mathcal{A}'$ does not output FAILURE on this input. $\qquad\square$

Informally, as mentioned earlier, we expect an iteration of $\mathcal{A}'$ to be (a, b)-good if there have been at most a iterations of Type I and at most b iterations of Type II before this one. Further, in each good iteration, we have a reasonable probability of success. These points are made precise in the following claim.

Claim 21. *Consider an iteration of $\mathcal{A}'$ on an (a, b)-good input where $a < a_0$ and $b < b_0$.*

- *If the iteration is of Type I, then the probability that the next iteration is not $(a+1, b)$-good is at most $\exp(-\Omega(k))$. Further, the probability of SUCCESS in the current iteration is at least $\eta_1 = 1/100$.*

- *If the iteration is of Type II, then the probability that the input produced for the next iteration is not $(a, b+1)$-good is at most $\exp(-\Omega(k))$. Further, the probability of SUCCESS in the current iteration is at least $\eta_2 = \varepsilon/100k$.*

Proof. Let $(S, T, (\rho'_j)_{j \notin S})$ be the input to this iteration of $\mathcal{A}'$ and $I \subseteq T$ the set chosen in Step 1 of $\mathcal{A}'$. The inputs to the next iteration are $(S_1, T_1, (\rho'_j)_{j \notin S_1})$ where $S_1 = S \setminus S'$, $T_1 = T \setminus I$ and ρ'_j for $j \in S'$ are sampled during the current iteration. For each $j \in S_1$, the set corresponding to $\bar{X}_j$ in the next iteration is denoted by $\bar{X}'_j$; formally, $\bar{X}'_j := \bar{X}_j \setminus \text{Vars}(F'_I)$.

We start with some preliminary observations. Note that by (17), we have $|X_j \setminus \bar{X}_j| = o(|\bar{X}_j|)$ and also as a consequence $|X_{j,\gamma} \setminus \bar{X}_{j,\gamma}| = o(|\bar{X}_j|) = o(|\bar{X}_{j,\gamma}|)$ for any $\gamma \in \{1, 2\}$. In particular, this implies that for any $j \in S$,

$$
\begin{aligned}
\delta_{j,1} + \delta_{j,2} &= \frac{|\text{Vars}(F'_I) \cap \bar{X}_{j,1}|}{|\bar{X}_{j,1}|} + \frac{|\text{Vars}(F'_I) \cap \bar{X}_{j,2}|}{|\bar{X}_{j,2}|} \\
&= \frac{|\text{Vars}(F'_I) \cap \bar{X}_{j,1}|}{((1/2) \pm o(1))|\bar{X}_j|} + \frac{|\text{Vars}(F'_I) \cap \bar{X}_{j,2}|}{((1/2) \pm o(1))|\bar{X}_j|} \\
&= (2 \pm o(1)) \frac{|\text{Vars}(F'_I) \cap \bar{X}_j|}{|\bar{X}_j|} = (2 \pm o(1)) \cdot \delta_j.
\end{aligned}
\tag{21}
$$

Since each variable $b_{j,\gamma}$ ($j \in S$, $\gamma \in \{1, 2\}$) is independently set to 1 with probability $\delta_{j,\gamma}$, we see (using (21)) that for each $j \in S$

$$\Pr[b_{j,1} = 1 \text{ or } b_{j,2} = 1] \le \delta_{j,1} + \delta_{j,2} \le 3\delta_j. \tag{22}$$

We now proceed to the proof of the statement of the claim.

Type I iteration. To show that $(S_1, T_1, (\rho'_j)_{j \notin S_1})$ is $(a+1, b)$-good, we need to show the corresponding versions[17] of (14) and (15).

---

[17]I.e., the version with a replaced by $a+1$ and $\bar{X}_j$ replaced by $\bar{X}'_j$.

Proof of (14). To show (14), note that it suffices to show

$$\Pr[|S'| \geq \frac{10M}{k}] \leq \exp(-\Omega(k)). \tag{23}$$

Note also that in the case of a Type I iteration, $S' = S''' \cup \hat{S}$ where $S''' = \{j \in S \mid b_{j,1} = 1 \text{ or } b_{j,2} = 1\}$. By our choice of I, we know that $|\hat{S}| \leq M/k$. Below, we will bound $|S''' \setminus \hat{S}|$.

Recall that each $b_{j,\gamma}$ ($\gamma \in [2]$) is set to 1 with probability $\delta_{j,\gamma}$. Hence, for each $j \in S$, the probability that $j \in S'''$ is at most $\delta_{j,1} + \delta_{j,2}$, which is at most $3\delta_j$ by (22). Note that for each $j \in S \setminus \hat{S}$, we have $\delta_j \leq (2\varepsilon/k)$. Thus, the expected value of $|S''' \setminus \hat{S}|$ is at most $3\sum_{j \in S \setminus \hat{S}} \delta_j \leq 3 \cdot (2\varepsilon/k) \cdot |S| \leq M/k$, as $|S| \leq M/8$ and $\varepsilon \leq 1$. Thus, by using our bound on $|\hat{S}|$ and a Chernoff bound (Theorem 5), we obtain

$$\Pr[|S'| \geq 10M/k] \leq \Pr[|S''' \setminus \hat{S}| \geq 9M/k] \leq \exp(-\Omega(M/k)) \leq \exp(-k)$$

proving (23). For the final inequality, we have used the fact that $M \geq m/10 = \Omega(k^{20})$.

Proof of (15). Fix any $j \in S_1$. Since $\hat{S} \cap S_1 = \emptyset$, we have $|\text{Vars}(F_I') \cap \bar{X}_j| \leq (2\varepsilon/k)|\bar{X}_j| \leq (2\varepsilon/k) \cdot |X_j|$. Thus, the set $\bar{X}_j' = \bar{X}_j \setminus \text{Vars}(F_I')$ satisfies

$$|\bar{X}_j'| \geq |\bar{X}_j| - (2\varepsilon/k)|X_j| \geq |X_j| \cdot \left(1 - \frac{2\varepsilon}{k} \cdot (a+1) - \frac{b}{M^{0.5}}\right)$$

thus proving (15). This shows that the next iteration is $(a+1, b)$-good except with probability $\exp(-\Omega(k))$.

Probability of SUCCESS. The algorithm outputs SUCCESS whenever it finds an $i \in I$ such that $\alpha_i' \neq \beta_i$. Since, as argued in the correctness proof above, α_i' equals the random variable α_i, the algorithm outputs SUCCESS whenever there is an i such that $\alpha_i \neq \beta_i$ for some $i \in I$.

Define the Boolean random variable α_I to be $\bigoplus_{i \in I} \alpha_i$; similarly, define β_I to be $\bigoplus_{i \in I} \beta_i$. For there to be some $i \in I$ such that $\alpha_i \neq \beta_i$, it suffices to have $\alpha_I \neq \beta_I$, or equivalently, whenever $\alpha_I \oplus \beta_I = 1$. We show that this occurs with probability at least η_1.

Each $\alpha_I = \bigoplus_{j \in S} \alpha_{I,j} \oplus \bigoplus_{j \notin S} \alpha_{I,j}$ where $\alpha_{I,j}$ is defined to be $\bigoplus_{i \in I} \alpha_{i,j}$. Note that for $j \in S$, $\alpha_{I,j}$ is 1 precisely when exactly one among the randomly chosen variables $x_{j,1}$ and $x_{j,2}$ lands up in $\text{Vars}(F_I')$; this happens precisely when $b_{j,1} \oplus b_{j,2} = 1$. Recall that $b_{j,1}$ and $b_{j,2}$ are independent random variables that are 1 with probability $\delta_{j,1}$ and $\delta_{j,2}$ respectively.

We know that $\delta_{j,\gamma} = |\text{Vars}(F_I') \cap \bar{X}_{j,\gamma}|/|\bar{X}_{j,\gamma}|$ for each $\gamma \in [2]$. Since $|\text{Vars}(F_I') \cap \bar{X}_{j,1}| \leq (1/3)|\bar{X}_j|$ (by the choice of I in Step 1 of $\mathcal{A}'$) and $|\bar{X}_{j,\gamma}| \geq ((1/2) - o(1)) \cdot |\bar{X}_j|$ as noted above, we have $\delta_{j,\gamma} \leq (2/3 + o(1))$. In particular, this implies that $\text{Unbias}(b_{j,\gamma}) \geq \min\{\delta_{j,\gamma}, 1 - \delta_{j,\gamma}\} \geq \delta_{j,\gamma}/3$[18].

Thus, Proposition 8 and (21) imply that for each $j \in S$,

$$\text{Unbias}(\alpha_{I,j}) = \text{Unbias}(b_{j,1} \oplus b_{j,2}) \geq \min\{\frac{1}{6}(\delta_{j,1} + \delta_{j,2}), \frac{1}{10}\} \geq \min\{\frac{1}{3}(1 - o(1))\delta_j, \frac{1}{10}\}.$$

Since the random variables $\alpha_{I,j}$ are independent and this is a Type I iteration (meaning that $\sum_{j \in S} \delta_j \geq 1$), we see from Proposition 8 that $\text{Unbias}(\alpha_I) \geq \min\{(1/6)\sum_{j \in S}(1 - o(1)) \cdot \delta_j, 1/10\} \geq 1/100$. This implies that the probability that $\alpha_I \oplus \beta_I = 1$ is at least $\eta_1 = 1/100$ as claimed.

Type II iteration. The proof is, to a large extent, similar to the proof in the Type I case. So we merely point out the main differences.

---

[18]Recall (Section 2.5) that $\text{Unbias}(b_{j,1}) = \min\{\Pr[b_{j,1} = 1], \Pr[b_{j,1} = 0]\}$.

Proof of (14). To show (the corresponding version of) (14), note that it suffices to show

$$\Pr[|S'| \geq 10 \cdot M^{0.5}] \leq \exp(-\Omega(k)). \tag{24}$$

Note that in this case, $S' = S'' \cup S'''$ where S'' is as defined in $\mathcal{A}'$ and $S''' = \{j \in S \mid b_{j,1} = 1 \text{ or } b_{j,2} = 1\}$.

To bound S'', we note that since we are dealing with a Type II iteration, we have $\sum_{j \in S} \delta_j < 1$. In particular, the number of $j \in S$ such that $\delta_j \geq 1/M^{0.5}$ is at most $M^{0.5}$. Hence, we have shown that $|S''| \leq M^{0.5}$.

To bound S''', we note as in the Type I case that the expected size of $|S'''|$ is at most $3 \sum_j \delta_j \leq 3$. In particular, by Theorem 5 (Item 2), we have

$$\Pr[|S'''| \geq M^{0.5}] \leq \exp(-\Omega(M^{0.5})) \leq \exp(-\Omega(k)).$$

Thus, we see that with probability at least $1 - \exp(-\Omega(k))$, both S'' and S''' have size at most $M^{0.5}$ and in this case, $|S'| \leq |S''| + |S'''| \leq 2M^{0.5}$. This proves (24).

Proof of (15). By our choice of S'', the set $S_1 = S \setminus S'$ does not contain any j such that $\delta_j \geq 1/M^{0.5}$. Hence, we see that for all $j \in S_1$, $|\mathrm{Vars}(F_I') \cap \bar{X}_j| \leq |\bar{X}_j|/M^{0.5}$. Hence, we have

$$|\bar{X}_j'| \geq |\bar{X}_j| - \frac{|\bar{X}_j|}{M^{0.5}} \geq |X_j| \cdot \left(1 - \frac{2\varepsilon}{k} \cdot a - \frac{b+1}{M^{0.5}}\right)$$

as desired, for each $j \in S_1$.

Probability of SUCCESS. We argue as in the Type I case that $\mathrm{Unbias}(\alpha_I)$ is large.

We know by our choice of I that there is at least one j such that $\delta_j \in [\varepsilon/2k, 2\varepsilon/k]$. Hence, exactly as in the Type I case, we see that $\mathrm{Unbias}(\alpha_I) \geq \min\{(1/6)\sum_j(1-o(1))\cdot\delta_j, 1/10\} \geq \varepsilon/12k$. In particular, this implies that the probability that $\alpha_I \oplus \beta_I = 1$ is at least $\eta_2 = \varepsilon/100k$, which yields the statement of the claim. □

We are now ready to upper bound the probability that the algorithm outputs FAILURE on input (S_0, T_0). We prove a more general statement. Assume that $(S, T, (\rho_j')_{j \notin S})$ is any (a, b)-good input to $\mathcal{A}'$ where $a \leq a_0$ and $b \leq b_0$. Then, we claim that

$$\Pr[\mathcal{A}'(S, T, (\rho_j')_{j \notin S}) \text{ outputs FAILURE}] \leq ((a_0-a)+(b_0-b))\cdot\exp(-\Omega(k))+(1-\eta_1)^{a_0-a}+(1-\eta_2)^{b_0-b} \tag{25}$$

where η_1, η_2 are as defined in the statement of Claim 21. We prove the above by downward induction on $a + b$ where $a \leq a_0$ and $b \leq b_0$.

The base case of the induction is defined to be the case when either $a = a_0$ or $b = b_0$. In this case, the claim is trivial since the right hand side of (25) is more than 1.

Now consider the case when $a < a_0$ and $b < b_0$. Let us analyze the behaviour of the algorithm $\mathcal{A}'$ on an (a, b)-good input $(S, T, (\rho_j')_{j \notin S})$. Assume that the current iteration is of Type I (the other case is similar). Claim 20 tells us that the algorithm does not output FAILURE in this iteration. By Claim 21, the probability that the algorithm produces an input (for the next iteration) that is *not* $(a + 1, b)$-good is at most $\exp(-\Omega(k))$; in this case, we give up and assume that the algorithm subsequently will output FAILURE. Finally, by Claim 21, the probability that the algorithm does not output SUCCESS in this round is at most $(1 - \eta_1)$. In particular, this implies that the probability that the algorithm does not output SUCCESS *and* produces an $(a + 1, b)$-good input for the next iteration is also at most $(1 - \eta_1)$. However, conditioned on this

event, we can use the inductive hypothesis to bound the probability of FAILURE. By the union bound and the induction hypothesis, we get

$\Pr[\mathcal{A}'(S, T, (\rho'_j)_{j \notin S})$ outputs FAILURE$]$

$\leq \exp(-\Omega(k)) + (1 - \eta_1) \cdot (((a_0 - (a + 1)) + (b_0 - b)) \cdot \exp(-\Omega(k)) + (1 - \eta_1)^{a_0 - (a+1)} + (1 - \eta_2)^{b_0 - b})$

$\leq \exp(-\Omega(k)) + ((a_0 - (a + 1)) + (b_0 - b)) \cdot \exp(-\Omega(k)) + (1 - \eta_1) \cdot (1 - \eta_1)^{a_0 - (a+1)} + (1 - \eta_2)^{b_0 - b}$

$\leq ((a_0 - a) + (b_0 - b)) \cdot \exp(-\Omega(k)) + (1 - \eta_1)^{a_0 - a} + (1 - \eta_2)^{b_0 - b}$

which completes the induction.

In particular, since the initial input (S_0, T_0) is $(0, 0)$-good, we see that

$\Pr[\mathcal{A}'$ outputs FAILURE on $(S_0, T_0)] \leq (a_0 + b_0) \cdot \exp(-\Omega(k)) + (1 - \eta_1)^{a_0} + (1 - \eta_2)^{b_0}$

$$\leq (a_0 + b_0) \cdot \exp(-\Omega(k)) + \exp(-\Omega(a_0)) + \exp(-\Omega(b_0 \varepsilon / k))$$
$$\leq M \cdot \exp(-\Omega(k)) + \exp(-\Omega(k)) + \exp(-\Omega(M^{0.45} \varepsilon / k))$$
$$\leq \exp(-\Omega(k))$$

where the second inequality follows from the definition of η_1 and η_2, the third from the definition of a_0 and b_0, and the last follows from the fact that $M \leq m^{\Delta'} \leq \exp(k^{0.02 + o(1)})$,[19] $\varepsilon \geq 1/M^{0.25}$, and $M = \Omega(m) = \Omega(k^{20})$. This implies (13) and hence completes the proof of Case 1.

5.2.2 Case 2

By renaming our segments if necessary, we assume that $X_1, \ldots, X_{M/2}$ are the segments that are not $(1/4)$-shattered. That is, for each of these segments X_j ($j \in [M/2]$), there exists an F'_{i_j} ($i_j \in [t]$) such that $\left|\text{Vars}(F'_{i_j}) \cap X_j\right| \geq (1/4) |X_j|$. By an averaging argument, we can say that there exists $b_j \in \{1, 2\}$ such that $\left|\text{Vars}(F'_{i_j}) \cap X_{j, b_j}\right| \geq (1/4) \left|X_{j, b_j}\right|$.

For each $j \leq M/2$, the restriction ρ_j is sampled according to the algorithm $\mathcal{A}$ described in Section 3.2. We first condition on the choice (among the options E_1, E_2, E_3) made by the algorithm $\mathcal{A}$ in sampling each ρ_j. We say that a ρ_j is *good* if the algorithm $\mathcal{A}$ decides on option $E_{b_j} \in \{E_1, E_2\}$ in sampling ρ_j. For each j, ρ_j is good with probability $1/3$. Let $\mathcal{E}_2$ denote the event that no ρ_j is good for $j \in [M/2]$ and let $\bar{\mathcal{E}}_2$ be the complement of $\mathcal{E}_2$. Clearly, we have

$$\Pr_\rho[\mathcal{E}_2] \leq (2/3)^{M/2} \leq \exp(-\Omega(M)). \tag{26}$$

Assume that $\mathcal{E}_2$ does not occur. Then, there is a $j \in [M/2]$ such that ρ_j is good. By renaming segments once more, we can assume that $j = 1$. Further, by renaming the F'_is, we assume that $i_1 = 1$; in particular, we have $|\text{Vars}(F'_1) \cap X_1| \geq |X_1|/4$. Finally, we can similarly assume that $b_1 = 1$, which implies that $|\text{Vars}(F'_1) \cap X_{1,1}| \geq |X_{1,1}|/4$.

We condition on any choice of the restrictions ρ_j for $j > 1$; this fixes the sets $\tilde{Y}'_j$ and $\tilde{Z}'_j$ defined above for $j > 1$. Conditioned on all these events, we see that the random restriction ρ_1 sets all the variables in $X_{1,2}$ to 0 deterministically, and hence can now be considered an $(X_{1,1}, Y_{1,1}, Z_{1,1})$-restriction, which we will call ρ'_1 for clarity. Note that $X_{1,1}, Y_{1,1}$ and $Z_{1,1}$ are $X^{(\Delta-1)}$-, $Y^{(\Delta-1)}$- and $Z^{(\Delta-1)}$-clones respectively.

Recall that the $X^{(\Delta-1)}$-clone $X_{1,1}$ is made up of m $X^{(\Delta-1)}$-segments $X_{1,1,1}, \ldots, X_{1,1,m}$. We similarly have $Y_{1,1} = \bigcup_{j \in [m]} Y_{1,1,j}$ and $Z_{1,1} = \bigcup_{j \in [m]} Z_{1,1,j}$.

The following claim is proved by a standard averaging argument.

---

[19]Note that there are at most $m^{\Delta'}$ many $X^{(\Delta)}$-segments in $G^{(\Delta')}$.

Claim 22. *There exist at least $m/8$ many $j \in [m]$ such that F_1' is $1/8$-heavy in $X_{1,1,j}$ i.e.,*
$|\mathrm{Vars}(F_1') \cap X_{1,1,j}| \geq (1/8) \cdot |X_{1,1,j}|$.

Assuming Claim 22, we will first show how we can invoke the induction hypothesis to prove the desired result.

By renaming if needed, let the $m/8$ segments from Claim 22 be $X_{1,1,1}, \ldots, X_{1,1,m/8}$. The restriction $\rho_1' = \rho_{1,1} \circ \cdots \circ \rho_{1,m}$ where $\rho_{1,\ell}$ for $\ell \in [m]$ is an $(X_{1,1,\ell}, Y_{1,1,\ell}, Z_{1,1,\ell})$-restriction sampled using the algorithm $\mathcal{A}$ in Section 3.2. We further condition on any choice of restrictions $\rho_{1,\ell}$ for all $\ell > m/8$. This fixes the sets $\tilde{Y}_{1,1,\ell}' := \rho_{1,\ell}(\mathrm{Vars}(F_1') \cap X_{1,1,\ell}) \cap Y_{1,1,\ell}$ and $\tilde{Z}_{1,1,\ell}'$ (defined similarly) for all $\ell > m/8$. Let F'' denote the restricted formula thus obtained and ρ_1'' the random restriction after this conditioning (i.e. $\rho_1'' = \rho_{1,1} \circ \cdots \circ \rho_{1,m/8}$).

We proceed to analyze $\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho)$. Since $\tilde{Y}' \cup Y'$ (resp. $\tilde{Z}' \cup Z'$) can be partitioned as $\bigcup_{i \in [t]} \hat{Y}_i$ (resp. $\bigcup_{i \in [t]} \hat{Z}_i$), we know (using Proposition 2 Item 3) that for any choice of ρ consistent with our choices so far,

$$\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) = \prod_{i \in [t]} \mathrm{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(F_i'|_\rho) \leq \mathrm{relrk}_{(\hat{Y}_1, \hat{Z}_1)}(F_1'|_\rho) = \mathrm{relrk}_{(\hat{Y}_1, \hat{Z}_1)}(F''|_{\rho_1''}). \quad (27)$$

To bound the latter term, we invoke the induction hypothesis on F'' with M replaced by $M' = m/8$ and ε replaced by $\varepsilon' = 1/8$. Here ρ'' is a (X'', Y'', Z'')-restriction where

$$X'' = \cup_{j \in [m/8]} X_{1,1,j}$$
$$Y'' = \cup_{j \in [m/8]} Y_{1,1,j}$$
$$Z'' = \cup_{j \in [m/8]} Z_{1,1,j}$$

and $\mathrm{Vars}(F'') = \bar{X} \cup \bar{Y} \cup \bar{Z}$ where

$$\bar{X} = \mathrm{Vars}(F_1') \cap X''$$
$$\bar{Y} = \left(\cup_{j>1} \tilde{Y}_j' \right) \cup \left(\cup_{\ell > m/8} \tilde{Y}_{1,1,j}' \right) \cup \hat{Y}_{1,0}$$
$$\bar{Z} = \left(\cup_{j>1} \tilde{Z}_j' \right) \cup \left(\cup_{\ell > m/8} \tilde{Z}_{1,1,j}' \right) \cup \hat{Z}_{1,0}.$$

Note that by Claim 22, it follows that $|\mathrm{Vars}(F'') \cap X_{1,1,j}| \geq \varepsilon' \cdot |X_{1,1,j}|$ for each $j \in [M']$ and hence the induction hypothesis for product-depth $(\Delta - 1)$ is applicable.

From the induction hypothesis, we see that

$$\Pr[\mathrm{relrk}_{(\hat{Y}_1, \hat{Z}_1)}(F''|_{\rho''}) \geq \exp(-k^{0.1}/(\Delta - 1))] \leq (\Delta - 1) \exp(-k^{0.1}/(\Delta - 1))$$

Using (27), we get

$$\Pr_\rho[\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) \geq \exp(-k^{0.1}/(\Delta - 1)) \mid \bar{\mathcal{E}}_2] \leq (\Delta - 1) \exp(-k^{0.1}/(\Delta - 1)).$$

By (26), we thus have

$$\Pr[\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) \geq \exp(-k^{0.1}/(\Delta - 1))] \leq (\Delta - 1) \exp(-k^{0.1}/(\Delta - 1)) + \exp(-\Omega(M))$$
$$\leq \Delta \exp(-k^{0.1}/(\Delta - 1)).$$

The last inequality holds since $M \geq m/10 > k$. This finishes the proof of Case 2 modulo the (standard) proof of Claim 22, which we give below for completeness.

Proof of Claim 22. For $j \in [m]$, let p_j denote $|\mathrm{Vars}(F_1') \cap X_{1,1,j}|/|X_{1,1,j}| = (m|\mathrm{Vars}(F_1') \cap X_{1,1,j}|)/|X_{1,1}|$. Since $|\mathrm{Vars}(F_1') \cap X_1| \geq (1/4)|X_1|$, we get that $\sum_{j=1}^m p_j \geq m/4$. Assume that the number of j such that $p_j \geq 1/8$ is strictly less than $m/8$. Then, we have

$$\sum_{j \in [m]} p_j \leq \sum_{j:p_j \geq 1/8} 1 + \sum_{j:p_j \leq 1/8} (1/8) < (m/8) + 1/8 \sum_{j:p_j \leq 1/8} 1 \leq m/8 + (1/8) \cdot m = m/4,$$

yielding $\sum_{j \in [m]} p_j < m/4$, contradicting our assumption on $\sum_{j \in [m/8]} p_j$. This proves the claim. $\square$

5.2.3 Case 3

Recall that in Case 3, we have that there is some F_i' ($i \in [t]$) that is (ε/k)-heavy in at least M/k many segments X_j. We will prove (4) in this case using induction. W.l.o.g., assume that $i = 1$.

Since $\tilde{Y}' \cup Y'$ (resp. $\tilde{Z}' \cup Z'$) can be partitioned as $\bigcup_{i \in [t]} \hat{Y}_i$ (resp. $\bigcup_{i \in [t]} \hat{Z}_i$), we know (Proposition 2 Item 3) that for any choice of ρ,

$$\mathrm{relrk}_{(\tilde{Y}' \cup Y', \tilde{Z}' \cup Z')}(F'|_\rho) = \prod_{i \in [t]} \mathrm{relrk}_{(\hat{Y}_i, \hat{Z}_i)}(F_i'|_\rho) \leq \mathrm{relrk}_{(\hat{Y}_1, \hat{Z}_1)}(F_1'|_\rho). \tag{28}$$

It is therefore sufficient to bound $\mathrm{relrk}_{(\hat{Y}_1, \hat{Z}_1)}(F_1'|_\rho)$.

W.l.o.g., we assume that F_1' is (ε/k)-heavy in segments $X_1, \ldots, X_{M/k}$. Each X_j ($j \in [M/k]$) consists of two half-segments, i.e., $X_j = X_{j,1} \cup X_{j,2}$. By averaging, we know that there is some $\gamma \in [2]$ such that F_1' is (ε/k)-heavy in at least half of $X_{1,\gamma}, \ldots, X_{M/k,\gamma}$. W.l.o.g. we assume that $\gamma = 1$. By renaming the segments, let us assume that F_1' is (ε/k)-heavy in $X_{1,1}, \ldots, X_{M/2k,1}$.

For each $j \leq M/2k$, the restriction ρ_j is sampled according to the algorithm $\mathcal{A}$ described in Section 3.2. Let us say that ρ_j is *good* if the algorithm $\mathcal{A}$ decides on option E_1 in sampling ρ_j. For each j, ρ_j is good with probability $1/3$. For each $j \in [M/2k]$, we condition on whether or not ρ_j is good. Let $\mathcal{E}_3$ denote the event that the number of good ρ_j is at most $M/8k$. Since the different ρ_j are sampled independently, a Chernoff bound (Theorem 5 Item 1) tells us that

$$\Pr_\rho[\mathcal{E}_3] \leq \exp(-\Omega(M/k)). \tag{29}$$

Assume that the event $\mathcal{E}_3$ does not occur. By renaming segments once more, we may assume that ρ_j is good for each $j \in [M/8k]$. We condition on any choice of the restrictions ρ_j for $j > M/8k$; this fixes the sets $\tilde{Y}_j'$ and $\tilde{Z}_j'$ for $j > M/8k$. Also, note that since we have conditioned on choosing the option E_1 for sampling restrictions $\rho_1, \ldots, \rho_{M/8k}$, all variables in $X_{j,2}$ ($j \in [M/8k]$) are set to 0 with probability 1. We will therefore think of ρ_j as an $(X_{j,1}, Y_{j,1}, Z_{j,1})$-restriction for the rest of the proof.

We will now prove our result by considering two subcases. Recall that each half-segment $X_{j,1}$ is an $X^{(\Delta-1)}$-clone and hence a union of m $X^{(\Delta-1)}$-segments $X_{j,1,1}, \ldots, X_{j,1,m}$. We will refer to these as *sub-segments* of $X_{j,1}$.

Case (3a): There are at least $M/16k$ many $j \in [M/8k]$ such that F_1' is at least $(\varepsilon/2k)$-heavy in at least $m^{0.5}$ many sub-segments of $X_{j,1}$. Without loss of generality, let us assume that the hypothesis of Case (3a) holds for the half-segments $X_{1,1}, \ldots, X_{M/16k,1}$. That is, F_1' is $(\varepsilon/2k)$-heavy in at least $m^{0.5}$ many sub-segments of $X_{j,1}$ for each $j \in [M/16k]$. Here, we will be able to apply induction on F_1' (which is a $\Sigma(\Pi\Sigma)^{(\Delta-1)}$ formula) to obtain an upper bound on its relative rank under the random restriction.

We will apply the induction hypothesis with M replaced by $M' = M/16k \times m^{0.5}$ and ε replaced by $\varepsilon' = \varepsilon/2k$. We claim that we have $\varepsilon' \geq (1/M')^{0.25}$ and $M' \geq m/10$ (which are

needed to apply induction). The latter inequality is trivial as $M' \geq M \geq m/10$. For the former, note that we have

$$\frac{1}{(\varepsilon')^4} = \frac{(2k)^4}{\varepsilon^4} \leq (2k)^4 M \leq \frac{M\sqrt{m}}{16k} = M'$$

where the first inequality uses the fact that $\varepsilon \geq 1/M^{0.25}$ and the last inequality uses the fact that $k \leq m^{0.05}$. This shows that $\varepsilon' \geq (1/M')^{0.25}$ as required.

We apply induction after some further processing. By renaming if necessary, let us assume that F_1' is heavy in the first $\sqrt{m}$ sub-segments $X_{j,1,1}, \cdots, X_{j,1,m^{0.5}}$ of $X_{j,1}$ for each $j \in [M/16k]$. Conditioned on what is known about ρ so far, for each $j \in [M/8k]$, the restriction ρ_j can be written as $\rho_j = \rho_{j,1} \circ \cdots \circ \rho_{j,m}$ where each $\rho_{j,\ell}$ for $\ell \in [m]$ is an $(X_{j,1,\ell}, Y_{j,1,\ell}, Z_{j,1,\ell})$-restriction sampled as per the sampling algorithm $\mathcal{A}$. We further condition on any choice of the restrictions ρ_{j_1} for all $j_1 > M/16k$ and $\rho_{j_2,\ell}$ for all $j_2 \in [M/16k]$ and $\ell > m^{0.5}$. This fixes the sets $\tilde{Y}_{j_1}'$, $\tilde{Z}_{j_1}'$ $(M/8k < j_1 \leq M/16k)$ and $\tilde{Y}_{j_2,1,\ell}' := \rho(\mathrm{Vars}(F_1') \cap X_{j_2,1,\ell}) \cap Y_{j_2,1,\ell}$ and $\tilde{Z}_{j_2,1,\ell}' := \rho(\mathrm{Vars}(F_1') \cap X_{j_2,1,\ell}) \cap Z_{j_2,1,\ell}$ $(j_2 \in [M/16k], \ell > m^{0.5})$.

Conditioned on our choices so far, the restriction ρ can now be identified an (X'', Y'', Z'')-restriction ρ'' where

$$X'' = \cup_{j \in [M/8k]} \cup_{\ell \in [m^{0.5}]} X_{j,1,\ell}$$
$$Y'' = \cup_{j \in [M/8k]} \cup_{\ell \in [m^{0.5}]} Y_{j,1,\ell}$$
$$Z'' = \cup_{j \in [M/8k]} \cup_{\ell \in [m^{0.5}]} Z_{j,1,\ell}.$$

and the restricted formula, which we denote by F_1'', satisfies $\mathrm{Vars}(F_1'') = \bar{X} \cup \bar{Y} \cup \bar{Z}$ where

$$\bar{X} = \mathrm{Vars}(F_1') \cap X''$$
$$\bar{Y} = \left(\cup_{j > M/16k} \tilde{Y}_j'\right) \cup \left(\cup_{j \in [M/16k]} \cup_{\ell > m^{0.5}} \tilde{Y}_{j,1,\ell}'\right) \cup \hat{Y}_{1,0}$$
$$\bar{Z} = \left(\cup_{j > M/16k} \tilde{Z}_j'\right) \cup \left(\cup_{j \in [M/16k]} \cup_{\ell > m^{0.5}} \tilde{Z}_{j,1,\ell}'\right) \cup \hat{Z}_{1,0}.$$

Based on the discussion above, we can now invoke the induction hypothesis with $M' = (M/16k)m^{0.5}$ and $\varepsilon' = \varepsilon/2k$ on F_1'' to obtain

$$\Pr_{\rho''}[\mathrm{relrk}_{(\hat{Y}_1, \hat{Z}_1)}(F_1''|_\rho) \geq \exp(-k^{0.1}/(\Delta-1))] \leq (\Delta-1)\exp(-k^{0.1}/(\Delta-1)).$$

Since the above holds for an arbitrary sequence of fixings after it was determined that $\mathcal{E}_3$ did not hold, we have thus shown the following.

$$\Pr_{\rho}[\mathrm{relrk}_{(\hat{Y}_1, \hat{Z}_1)}(F_1'|_\rho) \geq \exp(-k^{0.1}/(\Delta-1)) \mid \bar{\mathcal{E}}_3] \leq (\Delta-1)\exp(-k^{0.1}/(\Delta-1)). \tag{30}$$

We will use this below, after the proof of Case 3(b), to finish the proof in this case.

Case (3b): There are at least $M/16k$ many $j \in [M/8k]$ such that F_1' is at least $(\varepsilon/2k)$-heavy in less than $m^{0.5}$ many sub-segments of $X_{j,1}$. W.l.o.g., assume that for each $j \in [M/16k]$, there is a $h_j < \sqrt{m}$ such that F_1' is $(\varepsilon/2k)$-heavy only in the sub-segments $X_{j,1,1}, \ldots, X_{j,1,h_j}$.

For each $j \in [M/16k]$, let W_j be $X_{j,1} \cap \mathrm{Vars}(F_1')$. By our assumption in Case 3, we know that

$$|W_j| \geq \frac{\varepsilon}{k} \cdot |X_{j,1}| \quad \forall j \in [M/16k] \tag{31}$$

Also, we know that for any $\ell \in \{h_j + 1, \ldots, m\}$, $|X_{j,1,\ell} \cap W_j| < \frac{\varepsilon}{2k} \cdot |X_{j,1,\ell}|$ for each $j \in [M/16k]$. Therefore, we get that

$$\sum_{\ell > h_j} |X_{j,1,\ell} \cap W_j| \leq \frac{\varepsilon}{2k} \cdot |X_{j,1}| \quad \forall j \in [M/16k] \tag{32}$$

Using Equations (31) and (32) we get that

$$\sum_{\ell \in [h_j]} |X_{(j,1),\ell} \cap W_j| \geq \frac{\varepsilon}{2k} \cdot |X_{j,1}| \quad \text{for each } j \in [M/16k].$$

Thus by averaging and by using the fact that $h_j \leq m^{0.5}$, we get that for every $j \in [M/16k]$ there exists an $\ell_j \in [h_j]$ such that $|X_{j,1,\ell_j} \cap W_j| \geq \frac{\varepsilon}{2k} \cdot \frac{|X_{j,1}|}{m^{0.5}}$. By renaming if necessary, let $\ell_j = 1$ for all $j \in [M/16k]$.

Notice that in fact $|X_{j,1}| = m \cdot |X_{j,1,\ell}|$ for any $\ell \in [m]$. Therefore, we get

$$|X_{j,1,1} \cap \text{Vars}(F_1')| = |X_{j,1,1} \cap W_j| \geq \frac{\varepsilon}{2k} \cdot \frac{|X_{j,1,1}| \cdot m}{m^{0.5}} = \frac{\varepsilon}{2k} \cdot m^{0.5} \cdot |X_{j,1,1}| \quad \forall j \in [M/16k] \tag{33}$$

We will apply induction on the formula F_1' and the sub-segments $X_{j,1,1}$ ($j \in [M/16k]$). For the induction, the parameter M will thus be replaced by $M' = M/16k$ and ε by $\varepsilon' = \frac{\varepsilon}{2k} \cdot m^{0.5}$ (we can take this ε' by (33) above). To check that induction is possible with these parameters, we need to ensure that $\varepsilon' \geq 1/M'^{0.25}$ and $M' \geq m/10$. The first inequality follows as in Case (3a). Formally, we note that

$$\frac{1}{(\varepsilon')^4} = \frac{(2k)^4}{\varepsilon^4 m^2} \leq \frac{M}{16k} = M'$$

where the inequality follows from the fact that $\varepsilon \geq 1/M^{0.25}$ and $k = m^{0.05}$. This shows that $\varepsilon' \geq (1/M')^{0.25}$.

Now we will show that $M' \geq m/10$. We know that $M \geq m/10$, but in this case we will be able to get a better lower bound on the value of M, which will then give us the intended lower bound on M'. For this, first observe that $\varepsilon' = (\varepsilon/2k) \cdot m^{0.5}$. Note that by (33) and the fact that $|X_{j,1,1} \cap \text{Vars}(F_1')| \leq |X_{j,1,1}|$, we get

$$(\varepsilon/2k) \cdot m^{0.5} \leq 1 \Leftrightarrow \frac{1}{\varepsilon} \geq \frac{\sqrt{m}}{2k}.$$

We also know that $\varepsilon \geq 1/M^{0.25}$ and therefore, $M \geq 1/\varepsilon^4 \geq m^2/(2k)^4$. As $M' = M/16k$, we get that $M' \geq \frac{m^2}{8k \cdot (2k)^4} \geq m$. This gives us (a bound that is slightly better than) the desired bound on M'.

We now apply induction. As in Case (3a), some processing is needed. Note that conditioned on what is known about ρ so far, for each $j \in [M/8k]$, the restriction ρ_j sets all $\rho_j = \rho_{j,1} \circ \cdots \circ \rho_{j,m}$ where each $\rho_{j,\ell}$ for $\ell \in [m]$ is an $(X_{j,1,\ell}, Y_{j,1,\ell}, Z_{j,1,\ell})$-restriction sampled as per the sampling algorithm $\mathcal{A}$. We further condition on any choice the restrictions ρ_{j_1} for all $j_1 > M/16k$ and $\rho_{j_2,\ell}$ for all $j_2 \in [M/16k]$ and $\ell > 1$. This fixes the sets $\tilde{Y}'_{j_1}$, $\tilde{Z}'_{j_1}$ ($M/8k < j_1 \leq M/16k$) and $\tilde{Y}'_{j_2,1,\ell} := \rho(\text{Vars}(F_1') \cap X_{j_2,1,\ell}) \cap Y_{j_2,1,\ell}$ and $\tilde{Z}'_{j_2,1,\ell} := \rho(\text{Vars}(F_1') \cap X_{j_2,1,\ell}) \cap Z_{j_2,1,\ell}$ ($j_2 \in [M/16k], \ell > 1$).

Conditioned on our choices so far, the restriction ρ can be identified with an (X'', Y'', Z'')-restriction ρ'' where

$$X'' = \cup_{j \in [M/16k]} X_{j,1,1}$$
$$Y'' = \cup_{j \in [M/16k]} Y_{j,1,1}$$
$$Z'' = \cup_{j \in [M/16k]} Z_{j,1,1}.$$

and the restricted formula F_1'' satisfies $\mathrm{Vars}(F_1'') = \bar{X} \cup \bar{Y} \cup \bar{Z}$ where

$$\bar{X} = \mathrm{Vars}(F_1') \cap X''$$
$$\bar{Y} = \left(\cup_{j>M/16k}\tilde{Y}_j'\right) \cup \left(\cup_{j\in[M/16k]}\cup_{\ell>1}\tilde{Y}_{j,1,\ell}'\right) \cup \hat{Y}_{1,0}$$
$$\bar{Z} = \left(\cup_{j>M/16k}\tilde{Z}_j'\right) \cup \left(\cup_{j\in[M/16k]}\cup_{\ell>1}\tilde{Z}_{j,1,\ell}'\right) \cup \hat{Z}_{1,0}.$$

Based on the discussion above, we can now invoke the induction hypothesis with M' and ε' (as defined above) on F_1'' to obtain

$$\Pr_{\rho}[\mathrm{relrk}_{(\hat{Y}_1,\hat{Z}_1)}(F_1''|_{\rho''}) \geq \exp(-k^{0.1}/(\Delta-1))] \leq (\Delta-1)\exp(-k^{0.1}/(\Delta-1)).$$

Hence, we obtain as in Case (3a),

$$\Pr_{\rho}[\mathrm{relrk}_{(\hat{Y}_1,\hat{Z}_1)}(F_1'|_{\rho}) \geq \exp(-k^{0.1}/(\Delta-1)) \mid \bar{\mathcal{E}}_3] \leq (\Delta-1)\exp(-k^{0.1}/(\Delta-1)). \tag{34}$$

We now see how to finish the proof in both Cases (3a) and (3b). Using (30) and (34), we see that in each of the cases (3a) and (3b),

$$\Pr_{\rho}[\mathrm{relrk}_{(\hat{Y}_1,\hat{Z}_1)}(F_1'|_{\rho}) \geq \exp(-k^{0.1}/(\Delta-1))] \leq (\Delta-1)\exp(-k^{0.1}/(\Delta-1)) + \Pr_{\rho}[\mathcal{E}_3]$$
$$\leq (\Delta-1)\exp(-k^{0.1}/(\Delta-1)) + \exp(-\Omega(M/k))$$
$$\leq \Delta\exp(-k^{0.1}/(\Delta-1))$$

where the second inequality follows from (29) and the last inequality from the fact that $M = \Omega(m) = \Omega(k^{20})$.

By (28), we thus get

$$\Pr_{\rho}[\mathrm{relrk}_{(\tilde{Y}'\cup Y',\tilde{Z}'\cup Z')}(F'|_{\rho}) \geq \exp(-k^{0.1}/(\Delta-1))] \leq \Delta\exp(-k^{0.1}/(\Delta-1))$$

proving inequality (4) in this case. This completes the proof in Case 4.

References

[Ajt83]   M. Ajtai. σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983. URL: http://www.sciencedirect.com/science/article/pii/0168007283900386, doi:https://doi.org/10.1016/0168-0072(83)90038-6.

[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.

[BC15]   Suman K. Bera and Amit Chakrabarti. A depth-five lower bound for iterated matrix multiplication. In *Conference on Computational Complexity*, volume 33 of *LIPIcs*, pages 183–197. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[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.

[CLS18]   Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. Small-depth multilinear formula lower bounds for iterated matrix multiplication, with applications. In *STACS*, volume 96 of *LIPIcs*, pages 21:1–21:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[COST16]   Xi Chen, Igor Carboni Oliveira, Rocco A. Servedio, and Li-Yang Tan. Near-optimal small-depth lower bounds for small distance connectivity. In *STOC*, pages 612–625. ACM, 2016.

[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.

[FSS84]   Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, Dec 1984. URL: https://doi.org/10.1007/BF01744431, doi:10.1007/BF01744431.

[GKKS14]   Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, December 2014. URL: http://doi.acm.org/10.1145/2629541, doi:10.1145/2629541.

[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.

[GLSS15]   Dmitry Gavinsky, Shachar Lovett, Michael E. Saks, and Srikanth Srinivasan. A tail bound for read-k families of functions. *Random Struct. Algorithms*, 47(1):99–108, 2015. URL: https://doi.org/10.1002/rsa.20532, doi:10.1002/rsa.20532.

[Hås87]   J. Håstad. *Computational limitations of small-depth circuits*. ACM doctoral dissertation award. MIT Press, 1987. URL: https://books.google.co.in/books?id=_hOZAQAAIAAJ.

[Hås16]   Johan Håstad. An average-case depth hierarchy theorem for higher depth. In *FOCS*, pages 79–88. IEEE Computer Society, 2016.

[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.

[HPV77]   John E. Hopcroft, Wolfgang J. Paul, and Leslie G. Valiant. On time versus space. *J. ACM*, 24(2):332–337, 1977. URL: http://doi.acm.org/10.1145/322003.322015, doi:10.1145/322003.322015.

[HRST17]   Johan Håstad, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. *J. ACM*, 64(5):35:1–35:27, 2017.

[Jan04]   Svante Janson. Large deviations for sums of partly dependent random variables. *Random Struct. Algorithms*, 24(3):234–248, May 2004. URL: http://dx.doi.org/10.1002/rsa.v24:3, doi:10.1002/rsa.v24:3.

[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.

[Koi12]   Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012.

[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.

[KS16]   Mrinal Kumar and Ramprasad Saptharishi. Finer separations between shallow arithmetic circuits. In *FSTTCS*, volume 65 of *LIPIcs*, pages 38:1–38:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[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-$NC^2 \neq$ multilinear-NC^1. 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.

[RST15]   Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1030–1048, 2015. URL: https://doi.org/10.1109/FOCS.2015.67, doi:10.1109/FOCS.2015.67.

[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/10.1561/0400000039`.

[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`.

[VSBR83]   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`.

[Yao85]   Andrew C-C. Yao. Separating the polynomial-time hierarchy by oracles. In *Proc. 26th Annual Symposium on Foundations of Computer Science*, pages 1–10, Piscataway, NJ, USA, 1985. IEEE Press. URL: `http://dl.acm.org/citation.cfm?id=4479.4487`.