Near-Optimal Set-Multilinear Formula Lower Bounds

Deepanshu Kush
Department of Computer Science, University of Toronto, Canada

Shubhangi Saraf
Department of Computer Science and Department of Mathematics, University of Toronto, Canada

Abstract

The seminal work of Raz (J. ACM 2013) as well as the recent breakthrough results by Limaye, Srinivasan, and Tavenas (FOCS 2021, STOC 2022) have demonstrated a potential avenue for obtaining lower bounds for general algebraic formulas, via strong enough lower bounds for *set-multilinear* formulas.

In this paper, we make progress along this direction by proving *near-optimal* lower bounds against low-depth as well as unbounded-depth set-multilinear formulas. More precisely, we show that over any field of characteristic zero, there is a polynomial f computed by a polynomial-sized set-multilinear branching program (i.e., f is in *set-multilinear VBP*) defined over $\Theta(n^2)$ variables and of degree $\Theta(n)$, such that any product-depth Δ set-multilinear formula computing f has size at least $n^{\Omega(n^{1/\Delta}/\Delta)}$. Moreover, we show that any unbounded-depth set-multilinear formula computing f has size at least $n^{\Omega(\log n)}$.

If such strong lower bounds are proven for the iterated matrix multiplication (IMM) polynomial or rather, any polynomial that is computed by an *ordered* set-multilinear branching program (i.e., a further restriction of set-multilinear VBP), then this would have dramatic consequences as it would imply super-polynomial lower bounds for general algebraic formulas (Raz, J. ACM 2013; Tavenas, Limaye, and Srinivasan, STOC 2022).

Prior to our work, either only weaker lower bounds were known for the IMM polynomial (Tavenas, Limaye, and Srinivasan, STOC 2022), or similar strong lower bounds were known but for a hard polynomial not known to be even in set-multilinear VP (Kush and Saraf, CCC 2022; Raz, J. ACM 2009).

By known *depth-reduction* results, our lower bounds are essentially tight for f and in general, for any hard polynomial that is in set-multilinear VBP or set-multilinear VP. Any asymptotic improvement in the lower bound (for a hard polynomial, say, in VNP) would imply super-polynomial lower bounds for general set-multilinear *circuits*.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Algebraic Complexity, Set-multilinear, Formula Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.CCC.2023.15

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2023/017/

Acknowledgements We would like to thank Srikanth Srinivasan for several helpful and insightful discussions. Our early discussions with him were what inspired much of this work.

1 Introduction

1.1 Background on Algebraic Complexity

Algebraic Complexity Theory is the study of the complexity of computational problems which can be described as computing a multivariate polynomial $P(x_1, \ldots, x_N)$ over some elements $x_1, \ldots, x_N$ lying in a fixed field $\mathbb{F}$. Several fundamental computational tasks such as computing the determinant, permanent, matrix product, etc., can be represented using

38th Computational Complexity Conference (CCC 2023).
Editor: Amnon Ta-Shma; Article No. 15; pp. 15:1–15:33
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

this framework. The natural computational models that we investigate in this setting are models such as *algebraic circuits*, *algebraic branching programs*, and *algebraic formulas*, all of which employ the natural algebraic operations in $\mathbb{F}[x_1, \dots, x_N]$ to compute P.

An *algebraic circuit* over a field $\mathbb{F}$ for a multivariate polynomial $P(x_1, \dots, x_N)$ is a directed acyclic graph (DAG) whose internal vertices (called gates) are labeled as either + (sum) or × (product), and leaves (vertices of in-degree zero) are labeled by the variables x_i or constants from $\mathbb{F}$. A special output gate (the root of the DAG) represents the polynomial P. If the DAG happens to be a tree, such a resulting circuit is called an *algebraic formula.* The size of a circuit or formula is the number of nodes in the DAG. We also consider the product-depth of the circuit, which is the maximum number of product gates on a root-to-leaf path. The class VP (respectively, VF) then is defined to be the collection of all polynomials having at most polynomially large degree which can be computed by polynomial-sized circuits (respectively, formulas).

An *algebraic branching program* is a layered DAG with two special nodes in it: a start-node and an end-node. All edges of the ABP go from layer $\ell - 1$ to layer ℓ for some ℓ (say start-node is the unique node in layer 0) and are labeled by a linear polynomial. Every directed path γ from start-node to end-node computes the monomial P_γ, which is the product of all labels on the path γ. The ABP computes the polynomial $P = \sum_\gamma P_\gamma$, where the sum is over all paths γ from start-node to end-node. Its size is simply the number of nodes in the DAG, its *depth* is the length of the longest path from the start-node to the end-node, and *width* is the maximum number of nodes in any layer. The class VBP then is defined to be the collection of all polynomials which can be computed by polynomial-sized branching programs[1].

The complexity of these models is measured by the size, which serves as an indicator of the time complexity of computing the polynomial. The product-depth measures the extent to which this computation can be made parallel. As these models are supposed to construct a formal polynomial P, they are *syntactic* models of computation. This is unlike a *Boolean* circuit, which is only required to model specific truth-table constraints. The problem of proving algebraic circuit lower bounds is therefore widely considered to be easier than its Boolean counterpart. Indeed, we know that proving VP $\neq$ VNP, the algebraic analog of the P vs NP problem, is implied by the latter separation in the non-uniform setting ([5]). Similarly, proving super-polynomial lower bounds for algebraic formulas is the algebraic analogue of the NC^1 vs NP problem and is also considered to be one of the central challenges in algebraic complexity theory. We refer the reader to [32] for a more elaborate survey of this topic and for the formal definitions of the algebraic complexity classes VF, VBP, VP, and VNP.

1.2 A Recent Breakthrough

Much like in the Boolean setting, the problem of showing lower bounds for *general* algebraic circuits (or even formulas) has remained elusive. However, some remarkable progress has been made very recently by Limaye, Srinivasan, and Tavenas ([23]) who in a spectacular breakthrough, showed the first super-polynomial lower bounds for algebraic formulas of *all* constant depths. Prior to their work, the best known lower bound ([18]) even for product-depth 1 (or $\Sigma\Pi\Sigma$ formulas) was only almost-cubic. This is in stark contrast with the Boolean setting, in which we have known strong constant-depth lower bounds for many

[1] The inclusions VF $\subseteq$ VBP $\subseteq$ VP follow.

decades [3, 11, 38, 13, 31, 34]. Constant-depth formulas are critical to the study of algebraic complexity theory, as unlike the Boolean setting, strong enough bounds against them are known to yield VP $\neq$ VNP ([2]). This helps put into perspective the importance of the work [23].

The crucial step in the proof of the [23] result is to first establish super-polynomial lower bounds for a certain restricted class of (low-depth) algebraic formulas, namely *set-multilinear* formulas which we now define along with other important circuit models. A polynomial is said to be homogeneous if each monomial has the same total degree and *multilinear* if every variable occurs at most once in any monomial. Now, suppose that the underlying variable set is partitioned into d sets $X_1, \ldots, X_d$. Then the polynomial is said to be *set-multilinear* with respect to this variable partition if each monomial in P has *exactly* one variable from each set. Note that a set-multilinear polynomial is both multilinear and homogeneous. Next, we define different models of computation corresponding to these variants of polynomials classes. An algebraic formula/branching program/circuit is set-multilinear with respect to a variable partition $(X_1, \ldots, X_d)$ if each internal node in the formula/branching program/circuit computes a set-multilinear polynomial[2]. Multilinear and homogeneous formulas/branching programs/circuits are defined analogously.

Several well-studied and interesting polynomials happen to be set-multilinear. For example, the determinant and the permanent polynomials, the study of which is profoundly consequential to the field of algebraic complexity theory, are set-multilinear (with respect to the column variables). Another well-studied polynomial, namely the Iterated Matrix Multiplication polynomial, is also set-multilinear. The polynomial $\mathrm{IMM}_{n,d}$ is defined on $N = dn^2$ variables, where the variables are partitioned into d sets $X_1, \ldots, X_d$ of size n^2, each of which is represented as an $n \times n$ matrix with distinct variable entries. The polynomial $\mathrm{IMM}_{n,d}$ is defined to be the polynomial that is the $(1,1)$-th entry of the product matrix $X_1 X_2 \cdots X_d$. Note that hence, this polynomial *precisely* captures the computational power of a branching program of width n and depth d and is "complete" for the class VBP. This polynomial has a simple divide-and-conquer-based set-multilinear formula of size $n^{O(\log d)}$, and more generally for every $\Delta \leq \log d$, a set-multilinear formula of product-depth Δ and size $n^{O(\Delta d^{1/\Delta})}$, and circuit[3] of size $n^{O(d^{1/\Delta})}$. Even without the set-multilinearity constraint, no significantly better upper bound is known. It is reasonable to conjecture that this simple upper bound is tight up to the constant in the exponent.

The lower bounds in [23] for general constant-depth algebraic circuits are shown in the following sequence of steps:

1. It is shown that general low-depth algebraic formulas can be transformed to set-multilinear algebraic formulas of low depth, and without much of a blow-up in size (as long as the degree is small). More precisely, any product-depth Δ formula of size s computing a polynomial that is set-multilinear with respect to the partition $(X_1, \ldots, X_d)$ where each $|X_i| \leq n$, can be converted to a set-multilinear formula[4] of product-depth 2Δ and size $\mathrm{poly}(s) \cdot d^{O(d)}$. Such a "set-multilinearization" of general formulas of small degree was already shown before in [28] (which we describe soon in more detail); however, the main contribution of [23] here is to prove this *depth-preserving* version of it.

[2] Of course, a non-root node need not be set-multilinear with respect to the *entire* variable partition. Nevertheless, here we demand that it must be set-multilinear with respect to some *subset* of the collection $\{X_1, \ldots, X_d\}$.

[3] Any product-depth Δ (set-multilinear) circuit of size s can be simulated by a product-depth Δ (set-multilinear) formula of size $s^{2\Delta}$. Hence, any constant-depth formula lower bound automatically yields a corresponding circuit lower bound.

[4] There is also an intermediate "homogenization" step which we skip describing here for the sake of brevity.

2. Strong lower bounds are then established for low-depth set-multilinear circuits (of small enough degree). More precisely, any set-multilinear circuit C computing $\mathrm{IMM}_{n,d}$ (where $d = O(\log n)$) of product-depth Δ must have size at least $n^{d^{\exp(-O(\Delta))}}$. This combined with the first step yields the desired lower bound for general constant-depth circuits.

Given Raz's set-multilinearization of formulas of small degree that we alluded to, and this description of the set-multilinear formula lower bounds from [23] when $d = O(\log n)$, it is evident the "small degree" regime is inherently interesting to study – as it provides an avenue, via *hardness escalation*, for tackling one of the grand challenges of algebraic complexity theory, namely proving super-polynomial lower bounds for general algebraic formulas. However, we shall now see that even the large degree regime can be equally consequential in this regard.

1.3 The Large Degree Regime

Consider a polynomial P that is set-multilinear with respect to the variable partition $(X_1, \dots, X_d)$ where each $|X_i| \le n$. In this paper, we shall focus on studying set-multilinear formula complexity in the regime where d and n are *polynomially* related (as opposed to say, the assumption $d = O(\log n)$ described above). We now provide some background and motivation for studying this regime.

In follow-up work [36], the same authors showed the first super-polynomial lower bound against unbounded-depth set-multilinear formulas computing $\mathrm{IMM}_{n,n}$[5]. As is astutely described in [36], studying the set-multilinear formula complexity of IMM is extremely interesting and consequential even in the setting $d = n$ because of the following reasons:

- $\mathrm{IMM}_{n,n}$ is a *self-reducible* polynomial i.e., it is possible to construct formulas for $\mathrm{IMM}_{n,n}$ by recursively using formulas for $\mathrm{IMM}_{n,d}$ (for any $d < n$). In particular, if we had formulas of size $n^{o(\log d)}$ for $\mathrm{IMM}_{n,d}$ (for some $d < n$), this would imply formulas of size $n^{o(\log n)}$ for $\mathrm{IMM}_{n,n}$. In other words, an optimal $n^{\Omega(\log n)}$ lower bound for $\mathrm{IMM}_{n,n}$ implies $n^{\omega_d(1)}$ lower bounds for $\mathrm{IMM}_{n,d}$ for any $d < n$.
- Raz in [28] showed that if an N-variate set-multilinear polynomial of degree d has an algebraic formula of size s, then it also has a set-multilinear formula of size $\mathrm{poly}(s) \cdot (\log s)^d$. In particular, for a set-multilinear polynomial P of degree $d = O(\log N / \log\log N)$, it follows that P has a formula of size $\mathrm{poly}(N)$ if and only if P has a set-multilinear formula of size $\mathrm{poly}(N)$. Thus, having $N^{\omega_d(1)}$ set-multilinear formula size lower bounds for such a low degree would imply super-polynomial lower bounds for general formulas.

In particular, proving the optimal $n^{\Omega(\log n)}$ set-multilinear formula size lower bound for $\mathrm{IMM}_{n,n}$ would have dramatic consequences as it would yield general formula lower bounds (and more specifically, the separation $\mathrm{VF} \subsetneq \mathrm{VBP}$). To this end, the authors in [36] are able to show a weaker bound of the form $(\log n)^{\Omega(\log n)}$ instead. Even though it is the case that "simply" improving the base of this exponent from $\log n$ to n yields general formula lower bounds, it seems that we are still far from achieving it. Indeed, as is observed in [36], we do not even have the optimal $n^{\Omega(\sqrt{n})}$ lower bound for $\mathrm{IMM}_{n,n}$[6] when product-depth $\Delta = 2$. For constant (or low) product-depths (i.e., when $\Delta \le \log n$), [36] shows a set-multilinear formula size lower bound of $(\log n)^{\Omega(\Delta n^{1/\Delta})}$ for $\mathrm{IMM}_{n,n}$ (while we expect the lower bound to be $n^{\Omega(n^{1/\Delta})}$).

[5] Note that for $\mathrm{IMM}_{n,n}$, each X_i has size n^2, not n. But the important thing for us here is that the degree, n, is polynomially related to this parameter.

[6] This is known for set-multilinear (and even multilinear) $\Sigma\Pi\Sigma\Pi$ circuits (see [10, 19]), but those are only special cases of general product-depth 2 circuits, which are $\Sigma\Pi\Sigma\Pi\Sigma$.

The best set-multilinear lower bound we know for any explicit polynomial of degree $\Theta(n)$ and in $\mathrm{poly}(n)$ variables and for any constant $\Delta \geq 2$ is indeed $n^{\Omega(n^{1/\Delta})}$, from a recent work by the authors ([22]). However, the polynomial for which these bounds are obtained is *not* $\mathrm{IMM}_{n,n}$. The "hard" polynomial in this work is $\mathrm{NW}_{n,n}$, which comes from the class of so-called Nisan-Wigderson design-based polynomials[7] and is known to be in VNP, but not known to be even in VP. The authors are also able to establish an $n^{\Omega(\log n)}$ set-multilinear formula size lower bound for $\mathrm{NW}_{n,n}$ in the unbounded-depth setting. By far the most striking problem left open by this work is to "simplify" the hard polynomial from $\mathrm{NW}_{n,n}$ to $\mathrm{IMM}_{n,n}$.

We remark that such a line of simplification has been successful in other contexts in algebraic complexity theory. Indeed, for several lower bounds for algebraic circuit classes in the past, a lower bound was initially shown for the NW polynomial and then with additional effort, was shown to also hold for the IMM polynomial. For instance, [17] showed a lower bound of $n^{\Omega(\sqrt{n})}$ for the top fan-in of a $\Sigma\Pi^{[O(\sqrt{n})]}\Sigma\Pi^{[\sqrt{n}]}$ circuit computing the NW polynomial, which was subsequently shown for IMM by [10]. Similarly, [15] showed an $n^{\Omega(\sqrt{d})}$ size lower bound for homogeneous depth-4 algebraic formulas for the NW polynomial, which was then shown for IMM later in [21]. In our context of set-multilinear formula lower bounds, such a simplification from NW to IMM would be especially momentous as it would directly lead to general formula lower bounds. In this paper, although we are presently unable to simplify all the way down to IMM, we manage to make significant progress along this direction.

1.4 Our Results

The main result of this work is the following statement.

▶ **Theorem 1.** *Let N be a growing parameter and Δ be a constant integer. Then, over any field of characteristic zero, there is an explicit polynomial P_N defined over $N = \Theta(n^2)$ variables with degree $d = \Theta(n)$ that is set-multilinear with respect to the variable partition $X = (X_1, \ldots, X_d)$ where each $|X_i| = n$ and such that:*

- *there is a $\mathrm{poly}(N)$-size set-multilinear branching program computing P_N (i.e., its every internal node computes a set-multilinear polynomial),*
- *any set-multilinear formula of product-depth Δ computing P_N must have size at least $N^{\Omega(d^{1/\Delta})}$, and*
- *further, any set-multilinear formula of arbitrary product-depth P_N must have size at least $N^{\Omega(\log d)}$.*

▶ Remark 2. Similar to [22], the lower bound in Theorem 1 is actually $d^{\Omega(d^{1/\Delta}/\Delta)}$, where d is the degree of the underlying polynomial, and it holds as long as degree $d \leq n$ and the product-depth $\Delta \leq \log d/\log\log d$ (the details are deferred to the proof of Theorem 19 in Section 4).

A few more remarks are in order. First, given that $\mathrm{IMM}_{n,n}$ is complete for the class VBP as described in Section 1.2, one might expect that Theorem 1 immediately implies such a lower bound for $\mathrm{IMM}_{n,n}$ as well, thereby obtaining general formula lower bounds. Curiously, however, this is not the case. This is because the underlying reduction from P_N to $\mathrm{IMM}_{n,n}$ destroys the set-multilinearity of the formula, and hence the set-multilinear formula lower bounds no longer apply. Nevertheless, we observe that if we can make the hard polynomial in

[7] The name is inspired from [24].

Theorem 1 be computable by a polynomial-sized *ordered* set-multilinear branching program[8], then that does yield the desired lower bounds for $\mathrm{IMM}_{n,n}$. More precisely, given a variable partition $(X_1, \ldots, X_d)$ (say with each $|X_i| \leq n$), we say that a set-multilinear branching program of width n and depth d is *ordered* with respect to $(X_1, \ldots, X_d)$ if for each $\ell \in [d]$, all edges of the ABP from layer $\ell - 1$ to layer ℓ are labeled using a linear form *in* X_ℓ. Let $f(X_1, \ldots, X_d)$ be a polynomial that can be computed by an ordered set-multilinear ABP A. Then, given any set-multilinear formula computing $\mathrm{IMM}_{n,d}$ (say over the variables $\{x_{i,j}^{(\ell)}\}$ where $i, j \in [n]$ and $1 \leq \ell \leq d$), we immediately obtain a set-multilinear formula of the same size for f by replacing $x_{i,j}^{(\ell)}$ with the linear form $e_{i,j}^{(\ell)}$, where $e_{i,j}^{(\ell)}$ is the label of the edge in A between the i-th node of layer $\ell - 1$ and j-th node of layer ℓ. As a consequence, any lower bound on the size of a set-multilinear formula computing f yields a lower bound for one computing $\mathrm{IMM}_{n,d}$. We conclude that replacing P_N in Theorem 1 by any polynomial that is computable by an ordered set-multilinear ABP would yield general formula lower bounds! This observation raises the question of the relative power of ordered vs general set-multilinear ABPs – we leave this as an intriguing open problem (see Section 5).

We also remark that obtaining this precise bound is interesting also when viewed through the lens of *depth-reduction.* Tavenas ([35]), building on several prior works ([2, 20]), showed that any algebraic circuit of $\mathrm{poly}(N)$ size computing a homogeneous N-variate polynomial of degree d can be converted to a homogeneous circuit of product-depth[9] Δ of size $N^{O(d^{1/\Delta})}$. It easily follows from the proof that this depth reduction preserves syntactic restrictions. That is, if we start with a syntactically set-multilinear circuit, the resulting product-depth Δ circuit is also syntactically set-multilinear. Therefore, because P_N has a polynomial-sized set-multilinear circuit (in particular, a set-multilinear ABP), it follows that it has a product-depth Δ set-multilinear formula of size $N^{O(d^{1/\Delta})}$. Furthermore, by classical depth-reduction results[10], it follows that a size s, degree d set-multilinear circuit can be simulated by a set-multilinear formula of size $s^{O(\log d)}$. Hence, the lower bounds we obtain for P_N in Theorem 1 – in both, the constant and unbounded-depth settings – are *asymptotically optimal.* In fact, the precise bound in Theorem 1 is also *sharp* in the sense that any asymptotic improvement in its exponent for *any* constant Δ (say, for a set-multilinear polynomial in VNP) would imply super-polynomial set-multilinear circuit lower bounds (i.e., set-multilinear VP $\neq$ set-multilinear VNP), which would be quite a strong and exciting result, as it would demonstrate considerable progress towards the VP vs VNP problem.

On a related note, in [22], the authors posed a question about the possibility of obtaining *improved* depth-reduction bounds for set-multilinear circuits. More specifically, it was observed that if any asymptotic improvement in the exponent on the $N^{O(d^{1/\Delta})}$ bound for general circuits from [35] could be shown to hold for set-multilinear circuits in the setting of Theorem 1 (i.e., when $N = \Theta(d^2)$), then combined with the lower bounds for $\mathrm{NW}_{n,n}$, this would imply super-polynomial set-multilinear circuit lower bounds. It was noted that [10] rules out the possibility of obtaining a stronger reduction to depth-4, or $\Sigma\Pi\Sigma\Pi$ circuits, as it shows an $n^{\Omega(\sqrt{n})}$ size lower bound for set-multilinear depth-4 circuits computing $\mathrm{IMM}_{n,n}$, which of course has small polynomial-sized set-multilinear circuits. Nevertheless, there could still be the possibility of obtaining improved depth-reduction statements for product-depths 2 (which is $\Sigma\Pi\Sigma\Pi\Sigma$ and hence more general than depth-4) or higher, and combining it

[8] Interestingly, this model, along with that of read-once oblivious ABPs (or ROABPs), has been studied quite extensively in the polynomial identity testing (PIT) literature; see [9, 1, 12].

[9] The result is stated in [35] for $\Sigma\Pi\Sigma\Pi$ circuits but the proof can be appropriately modified for larger product-depths.

[10] See [37] and then, [4] for an adaptation to the set-multilinear setting.

with the lower bound for $\mathrm{NW}_{n,n}$ to obtain general set-multilinear circuit lower bounds. We answer this question in the negative and remark that Theorem 1 implies that an improved depth-reduction bound for set-multilinear circuits is impossible (at least when N and d are polynomially related).

We also point out the differences in the quality of the best lower bounds known in the closely related (and more general) *multilinear* setting. Despite the multilinear formula model receiving significant attention in the literature[11], to the best of our knowledge, the best known lower bound for a polynomial of degree n over $\mathrm{poly}(n)$ variables even for product-depth 2 multilinear formulas[12] is only $2^{\Omega(\sqrt{n})}$ ([7]), and generalizes as $2^{\Omega(\Delta n^{1/\Delta})}$ for higher $\Delta \leq \log n$. In contrast, using the terminology of [23], the lower bounds that we obtain for constant product-depth *set-multilinear* formulas in this paper (and indeed, in [22]) are stronger, *non-FPT bounds.* Furthermore, we point out that even solely the third item of Theorem 1 i.e., an $n^{\Omega(\log n)}$ lower bound for a set-multilinear polynomial of degree n over $\mathrm{poly}(n)$ variables computable by a small set-multilinear branching program, is a new result – as far as we know, it is not implied by any prior work. For example, though [27] shows that the $n \times n$ determinant and permanent require $n^{\Omega(\log n)}$ multilinear formula size, these polynomials are not actually known to have small set-multilinear (or even multilinear) circuits – in fact, they are conjectured not to ([25]).

We now move on to the second result of this paper. As noted earlier, prior to this work, the "hard" polynomial for which we had the same lower bounds as Theorem 1 was not known to be even in VP. In the result below, we construct a set-multilinear polynomial in VP matching the bounds of [22].

▶ **Theorem 3.** *Let N be a growing parameter and Δ be a constant integer. Then, over any field of characteristic zero, there is an explicit polynomial Q_N defined over $N = \Theta(n^2)$ variables with degree $d = \Theta(n)$ that is set-multilinear with respect to the variable partition $X = (X_1, \dots, X_d)$ where each $|X_i| = n$ and such that:*

- *there is a $\mathrm{poly}(N)$-size set-multilinear circuit computing Q_N,*
- *any set-multilinear formula of product-depth Δ computing Q_N must have size at least $N^{\Omega(d^{1/\Delta})}$, and*
- *further, any set-multilinear formula of arbitrary product-depth Q_N must have size at least $N^{\Omega(\log d)}$.*

▶ Remark 4. Similar to Theorem 1, the lower bound in Theorem 3 is actually $d^{\Omega(d^{1/\Delta}/\Delta)}$, where d is the degree of the underlying polynomial, and it holds as long as degree $d \leq n$ and the product-depth $\Delta \leq \log d / \log\log d$ (the details are deferred to the proof of Theorem 10 in Section 3).

Evidently, despite already being a new result, Theorem 3 is subsumed by Theorem 1. However, as we shall see in Section 3 and also in the proof overview below, this construction and the associated lower bound argument is simpler than that of Theorem 1. Moreover, this argument will be quite instructive and helpful for the reader to ease into the proof of the main result (Theorem 19 in Section 4).

[11] See [25, 30, 7, 6, 16] for results in the bounded-depth setting and [27, 8, 14, 19] for results in the unbounded-depth setting. Note that however, in many of these works, the "hard" polynomial is not set-multilinear and as such, the corresponding lower bounds do not even apply in our setting.

[12] The situation is significantly better for $\Delta = 1$ (or multilinear $\Sigma\Pi\Sigma$ formulas) as [16] shows a lower bound of $n^{\Omega(d)}$ – in fact, for $\mathrm{IMM}_{n,d}$.

1.5 Proof Overview and Relation to Prior Work

In this subsection, we give an overview of the proof techniques used in both Theorems 1 and 3. We divide the subsection into two parts: the first part discusses the construction of our hard polynomial in VP (which is mainly inspired from a result ([29]) of Raz and Yehudayoff) and the second part discusses the construction of our hard polynomial in VBP (which relies upon the *arc-partition* framework of Dvir, Malod, Perifel, and Yehudayoff ([8])).

VP Construction

We shall first discuss an overview of the proof of Theorem 3. At a high-level, our overall proof techniques are similar to that of many known lower bounds. We work with a measure that is known to be small for all polynomials computed by small enough set-multilinear formulas (suitably so in the bounded and unbounded-depth settings) from the work [22], where it is also shown to be large for the NW polynomial. These *partial derivative measures* were introduced by Nisan and Wigderson in [25], who used them to prove the constant-depth set-multilinear formula lower bounds we discussed earlier. [23, 36] use a particular variant of this measure and this measure is in turn inspired from these works.

Given a variable partition $(X_1, \ldots, X_d)$, the idea is to label each set of variables X_i as "positive" or "negative" uniformly at random. Let $\mathcal{P}$ and $\mathcal{N}$ denote the set of positive and negative indices respectively, and let $\mathcal{M}^{\mathcal{P}}$ and $\mathcal{M}^{\mathcal{N}}$ denote the sets of all set-multilinear monomials over $\mathcal{P}$ and $\mathcal{N}$ respectively. For a polynomial f that is set-multilinear over the given variable partition $(X_1, \ldots, X_d)$, the measure then is simply the rank of the "partial derivative matrix" $\mathcal{M}(f)$ whose rows are indexed by the elements of $\mathcal{M}^{\mathcal{P}}$ and columns indexed by $\mathcal{N}^{\mathcal{P}}$, and the entry of this matrix corresponding to a row m_1 and a column m_2 is the coefficient of the monomial $m_1 \cdot m_2$ in the given polynomial. We remark that though this was inspired by the measure and the techniques from [23], it is also reminiscent of the measure used in [26, 27] to prove multilinear formula lower bounds. [22] shows that indeed for the NW polynomial, the matrix $\mathcal{M}(NW)$ *always* has full-rank (at least when conditioned on the event $|\mathcal{P}| = |\mathcal{N}|$).

In proving Theorem 3, our main contribution is to construct a set-multilinear polynomial Q such that $\mathcal{M}(Q)$ always has full-rank – but in addition, Q is computable by a small set-multilinear circuit. For this, we turn to the literature on the multilinear setting for inspiration. In [26], Raz constructed a multilinear polynomial g computable by a small multilinear circuit and showed a super-polynomial (general-depth) multilinear formula size bound for it. The measure used was the rank of a matrix defined analogously to $\mathcal{M}(\cdot)$. Our starting point for constructing Q was a *simplification* of g (which we call h) by Raz and Yehudayoff ([29]) using Dyck words[13], which we shall describe in more detail in Section 3. One idea that is key in these constructions is the introduction of *auxiliary* variables: $h = h(X, \Lambda)$ is defined over an *original* variable set $X = \{x_1, \ldots, x_n\}$ and an *auxiliary* variable set Λ of poly(n) size. It is then shown that the matrix associated to $h(X)$ has full-rank (i.e., h has "large" measure), when viewed as a matrix over the extended field $\mathbb{F}(\Lambda)$. In other words, the auxiliary variables assist in showing that its matrix (whose entries are now polynomials over variables in Λ) is non-singular.

While attempting to "set-multilinearize" the construction of $h(X, \Lambda)$ in order to define our $Q(X_Q, \Lambda_Q)$ (say where $X_Q = (X_1, \ldots, X_d)$ and each $|X_i| = n$), we were able adapt the correct (i.e., a set-multilinear) dependence of Q on the X_Q-variables (from that of h

[13] [29] does not actually explicitly use Dyck words in its construction, but we benefited from its exposition given in [32].

on X) in a relatively straightforward manner – this involved picking the right “gadget” or “building block” in the set-multilinear setting, which turns out to the *inner product gadget* (see Observation 8). Essentially, the simple observation that if X_1 is labeled “positive” and X_2 is labeled “negative”, then the $n \times n$ matrix corresponding to the inner product polynomial $X_1 \cdot X_2$ is full-rank allows us to “build” more complicated and higher-degree full-rank polynomials, similar to how h is “built” by [29]. However, the main hurdle that we encountered while trying to construct Q using the construction of h was to achieve the correct dependence on the auxiliary variables. The issue is that if we introduce too many *sets* of auxiliary variables (i.e., if $\Lambda_Q = (\Lambda_1, \ldots, \Lambda_{d'})$ and $d' = \omega(d)$), then the degree of the polynomial blows up and because we work over the extended field $\mathbb{F}(\Lambda)$, the final quantitative expression for the lower bound in the constant-depth case of Theorem 3 suffers (in fact, it becomes even worse than the aforementioned lower bound of [7] in the constant-depth multilinear formula model). As a consequence, we need to be judicious in our use of the auxiliary variables – we highlight some of the finer details later on in Section 3. For this reason, the analysis of our hard polynomial being full-rank ends up being more intricate than [29]. In turn, this leads to the demand that the characteristic of $\mathbb{F}$ be zero – although we suspect that this assumption should not be necessary; see the discussion in Section 5.

VBP Construction

For proving Theorem 1, we build upon the work of Dvir, Malod, Perifel, and Yehudayoff ([8]), who showed the first separation between multilinear branching programs and formulas. That is, they constructed an n-variate polynomial F that can be computed by a small multilinear branching program, but needs multilinear formulas of size $n^{\Omega(\log n)}$ to compute. Our overall strategy is to adapt their approach to our set-multilinear setting – however, there are some inherent difficulties in doing so because of the nature of the very strong bounds sought in the low-depth setting (which was not an issue for [8] as this setting was not considered in that work). In what follows, we provide an overview of the *arc-partition* framework of [8], state it in our set-multilinear setting, and describe the additional challenges we face with the adaptation.

The proof of Theorem 1 consists of two parts: (i) constructing a small set-multilinear ABP computing a polynomial G and (ii) showing that any set-multilinear multilinear formula computing G must be very large (appropriately so in the constant and general-depth settings). The two parts have opposing demands: In part (i) we wish to make the polynomial G simple enough so that a small ABP can compute it, whereas in part (ii) we will need to rely on the hardness of G to prove lower bounds. One might wonder if we can get away with using the same rank measure that was defined above for the VP construction in order to meet these two demands. However, as far as we know, full-rank polynomials (in the sense described above) may also require super-polynomial sized set-multilinear ABPs. [8] were faced with a similar challenge: full-rank multilinear polynomials (say with respect to the aforementioned analogous measure of [26]) may also require super-polynomial sized multilinear ABPs. Thus, in order to prove a separation between multilinear ABPs and formulas, they sought a property which is *weaker* than being full-rank but is still useful enough for proving lower bounds. One of the main ideas in their proof is an ingenious construction of a special subset of partitions, called *arc-partitions*, which is sufficiently powerful to carry through the lower bound proof and, at the same time, simple enough to carry part (i) of the proof. In this context, a partition simply refers to a particular “positive”/“negative” labelling of the variable sets $X_1, \ldots, X_d$. The point is that the support of the distribution over these *arc-partitions* turns out to be much smaller than the support of the uniform distribution over such labellings that was used as our measure in the VP construction. Nevertheless, after overcoming some hurdles that we soon describe, we are able to adapt their argument to show that every *arc-full-rank* polynomial f (i.e., the matrix $\mathcal{M}(f)$ is always full-rank, but now defined only with respect to the labellings coming from this special *arc-partition* distribution, instead of the uniform distribution) must have very large set-multilinear formulas – appropriately so in the constant and general-depth settings.

Let us now describe this family of partitions (stated in our set-multilinear setting) and its advantages. More specifically, we will describe a distribution over partitions (or labellings, as explained above). The partitions that will have positive probability of being obtained in this distribution will be called arc-partitions. The distribution is defined according to the following (iterative) sampling algorithm. Position the d variable sets on a cycle with d nodes so that there is an edge between i and $i+1$ modulo d. Start with the arc $[L_1, R_1] = \{1, 2\}$ (an arc is a connected path on the cycle). At step $t > 1$ of the process, maintain a partition of the arc $[L_t, R_t]$. "Grow" this partition by first picking a pair uniformly at random out of the three possible pairs $\{L_t - 2, L_t - 1\}, \{L_t - 1, R_t + 1\}, \{R_t + 1, R_t + 2\}$, and then choosing a labelling (or partition) Π on this pair i.e., assigning one of them "positive" and the other "negative" uniformly at random. After $d/2$ steps, we have chosen a partition of the d variable sets into two disjoint, equal-size sets of variables $\mathcal{P}$ and $\mathcal{N}$.

Given these arc-partitions of [8], let us now briefly describe how we obtain the desired optimal lower bounds in the constant-depth setting. In part (i) of the proof, in order to construct an arc-full-rank set-multilinear branching program, we face similar challenges as we did in the VP construction – but similarly, a more careful use of the auxiliary variables comes to the rescue. Next, we show that with high probability over the arc-partition distribution, the rank of a polynomial computed by a product-depth Δ set-multilinear formula is (appropriately) small. This is done via a proof by induction on Δ. We separately show that each summand C_i of $C = C_1 + \cdots + C_t$ for a product-depth Δ formula C has small enough rank, yielding the desired bound by the sub-additivity of rank. There are two cases: either C_i already has a factor of very large degree (i.e., at least $\sim d^{\frac{\Delta-1}{\Delta}}$, which allows us to use the inductive hypothesis for $\Delta - 1$) or otherwise, we argue that we may assume that it has many factors (roughly $K \sim d^{1/\Delta}$ many) of a similar degree. It is this inductive argument (and specifically, the first case) that forces us to work with an arc-partition over a larger D-cycle (where $D \geq d$) – one of the reasons contributing to a more nuanced analysis than [8]. In the second case, the many factors then define a "non-redundant" K-coloring of the d variable sets. This is simply a (partial) mapping $C_i : [D] \to [K]$ so that the pre-images of every color $k \in [K]$ are not too small (and of similar sizes). A color k is said to be "balanced" with respect to a partition Π if the number of "positive" variable sets of color k is roughly the same as the number of "negative" variable sets of color k. Now, for a given coloring C_i, if we choose a random partition Π from the set of all partitions, simple properties of the hyper-geometric distribution imply that the probability that all colors in C_i are "balanced" is at most $p = d^{-\Omega(K)} = d^{-\Omega(d^{1/\Delta})}$. This bound, in turn, proves a roughly $1/p = d^{\Omega(d^{1/\Delta})}$ lower bound for the size of product-depth Δ set-multilinear formulas for the VP construction (Theorem 3). Following a similar overall outline, we adapt the [8] argument to show that for any "non-redundant" partial K-coloring C_i, for a random arc-partition, the probability that all colors in C_i are "balanced" is at most $d^{-\Omega(K)}$ as well. This turns out to be significantly more difficult than showing it for a random partition (from the set of all partitions). Furthermore, because we seek such strong and optimal bounds in the low-depth setting, the analysis turns out to be more intricate (Section 4.4 in particular). Throughout Section 4 where we formally prove Theorem 1, we have suitably placed remarks to point out the locations where we require a different technical or conceptual argument than [8].

2 Preliminaries

2.1 Relative Rank and its Properties

We first describe the notation that we need to define the measures that we use to prove Theorems 1 and 3.

▶ **Definition 5** (Relative Rank Measure of [23, 36]). *Let f be a polynomial that is set-multilinear with respect to the variable partition $(X_1, X_2, \ldots, X_d)$ where each set is of size n. Let $w = (w_1, w_2, \ldots, w_d)$ be a tuple (or word) of non-zero real numbers such that $2^{|w_i|} \in [n]$ for all $i \in [d]$. For each $i \in [d]$, let $X_i(w)$ be the variable set obtained by removing arbitrary variables from the set X_i such that $|X_i(w)| = 2^{|w_i|}$, and let $\overline{X}(w)$ denote the tuple of sets of variables $(X_1(w), \ldots, X_d(w))$. Corresponding to a word w, define $\mathcal{P}_w := \{i \mid w_i > 0\}$ and $\mathcal{N}_w := \{i \mid w_i < 0\}$. Let $\mathcal{M}_w^{\mathcal{P}}$ be the set of all set-multilinear monomials over a subset of the variable sets $X_1(w), X_2(w), \ldots, X_d(w)$ indexed by $\mathcal{P}_w$, and similarly let $\mathcal{M}_w^{\mathcal{N}}$ be the set of all set-multilinear monomials over these variable sets indexed by $\mathcal{N}_w$.*

Define the 'partial derivative matrix' matrix $\mathcal{M}_w(f)$ whose rows are indexed by the elements of $\mathcal{M}_w^{\mathcal{P}}$ and columns indexed by the elements of $\mathcal{N}_w^{\mathcal{P}}$ as follows: the entry of this matrix corresponding to a row m_1 and a column m_2 is the coefficient of the monomial $m_1 \cdot m_2$ in f. We define

$$\mathrm{relrk}_w(f) := \frac{\mathrm{rank}(\mathcal{M}_w(f))}{\sqrt{|\mathcal{M}_w^{\mathcal{P}}| \cdot |\mathcal{M}_w^{\mathcal{N}}|}} = \frac{\mathrm{rank}(\mathcal{M}_w(f))}{2^{\frac{1}{2}\sum_{i\in[d]}|w_i|}}.$$

▶ **Definition 6.** *For any tuple $w = (w_1, \ldots, w_t)$ and a subset $S \subseteq [t]$, we shall refer to the sum $\sum_{i\in S} w_i$ by w_S. And by $w|_S$, we will refer to the tuple obtained by considering only the elements of w that are indexed by S. We denote by $\mathbb{F}_{\mathrm{sm}}[\mathcal{T}]$ the set of set-multilinear polynomials over the tuple of sets of variables $\mathcal{T}$.*

The following is a simple result that establishes various useful properties of the relative rank measure.

▷ Claim 7 ([23]).

1. (Imbalance) Say $f \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$. Then, $\mathrm{relrk}_w(f) \le 2^{-|w_{[d]}|/2}$.
2. (Sub-additivity) If $f, g \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$, then $\mathrm{relrk}_w(f+g) \le \mathrm{relrk}_w(f) + \mathrm{relrk}_w(g)$.
3. (Multiplicativity) Say $f = f_1 f_2 \cdots f_t$ and assume that for each $i \in [t]$, $f_i \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w|_{S_i})]$, where $(S_1, \ldots, S_t)$ is a partition of $[d]$. Then

$$\mathrm{relrk}_w(f) = \prod_{i\in[t]} \mathrm{relrk}_{w|_{S_i}}(f_i).$$

2.2 Inner Product Gadget

We crucially need the following observation to construct the hard polynomials in Theorems 1 and 3.

▶ **Observation 8.** *Let $n = 2^k$ and $X_1 = \{x_{1,1}, \ldots, x_{1,n}\}$ and $X_2 = \{x_{2,1}, \ldots, x_{2,n}\}$ be two disjoint sets of variables. Then, for any* symmetric *word $w \in \{k, -k\}^2$ (i.e., where $w_1 + w_2 = 0$) and for the inner product "gadget" $f = X_1 \cdot X_2 = \sum_{i=1}^{n} x_{1,i}x_{2,i}$, $\mathrm{relrk}_w(f) = 1$ i.e., $\mathcal{M}_w(f)$ is full-rank.*

3 A Hard Set-multilinear Polynomial in VP

3.1 Description of the Polynomial

Let d be an even integer and let $X = (X_1, \ldots, X_d)$ be a collection of sets of variables where each $|X_i| = n$, and similarly, let $Y = (Y_1, \ldots, Y_d)$ be a distinct collection of sets of variables where each $|Y_i| = n$. We shall refer to the Y-variables as the *auxiliary* variables. For i and $j \in \{1, \ldots, d\}$, let $X_i \cdot X_j$ denote the inner-product quadratic form $\sum_{k=1}^{n} x_{ik} x_{jk}$. Here, we shall assume that $X_i = \{x_{i,1}, \ldots, x_{i,n}\}$ and $Y_i = \{y_{i,1}, \ldots, y_{i,n}\}$.

For two integers $i \in \mathbb{N}$ and $j \in \mathbb{N}$, we denote $[i,j] = \{k \in \mathbb{N} : i \leq k \text{ and } k \leq j\}$ and call such a set an *interval.* For every interval $[i,j] \subseteq [d]$, we define a polynomial $f_{i,j}(X,Y) \in \mathbb{F}_{\mathrm{sm}}[X_i, \ldots, X_j, Y_i, \ldots, Y_j]$ as follows:

$$f_{i,j} = \begin{cases} y_{i,j} y_{j,i} (X_i \cdot X_j) & \text{if } j = i+1 \\ 0 & \text{if } j - i \text{ is even} \\ y_{i,j} y_{j,i} (X_i \cdot X_j) \cdot f_{i+1,j-1} + \sum_{r=i+1}^{j-1} f_{i,r} f_{r+1,j} & \text{otherwise} \end{cases}$$

▶ Remark 9. As described in Section 1.5, other than the use of the inner product gadget, one key difference between $f_{i,j}$ and the construction in [29] is that it uses fewer auxiliary variables. More specifically, while [29] had a "fresh" auxiliary variable for every choice of i, r, j in the sum, we are unable to afford that not only because it destroys the set-multilinearity of the polynomial but most importantly, because of the aforementioned degree blow-up. This is also the reason why more straightforward attempts to "set-multilinearize" [29] such as by adding two "copies" y_0 and y_1 for each of their auxiliary variables y (where intuitively y_0 and y_1 correspond to setting y as 0 or 1 respectively in their argument) do not work.

The following is a more precise and general version of Theorem 3 that is stated in Section 1. We also incorporate Remark 4 here and show our lower bound for any degree $d \leq n$. Theorem 3 follows from the special case $d = n$.

▶ **Theorem 10.** *Let $n = 2^k$, and suppose $d \leq n$ be an even integer that is large enough*[14]*, and $1 \leq \Delta \leq \log d / \log \log d$ be any positive integer. Let X_i, Y_i denote the sets of n variables $\{x_{i,j} : j \in [n]\}$ and $\{y_{i,j} : j \in [n]\}$ respectively and let X, Y be the tuples $(X_1, \ldots, X_d)$ and $(Y_1, \ldots, Y_d)$. Then,*

- *there is a $\mathrm{poly}(n,d)$-size set-multilinear circuit computing $F_{n,d} = f_{1,d}(X,Y)$ as defined above,*
- *any set-multilinear formula of product-depth Δ computing $F_{n,d}$ must have size at least $d^{\Omega(d^{1/\Delta}/\Delta)}$, and*
- *further, any set-multilinear formula of arbitrary product-depth computing $F_{n,d}$ must have size at least $d^{\Omega(\log d)}$.*

3.2 Proof of Hardness

Note that the first item in Theorem 10 follows immediately from the recursive definition of $f_{1,d}$ (notice that there are only up to d^2 many distinct intervals of $[d]$). For proving the next two items, we invoke the *symmetric word* framework of [22]. The following couple of lemmas help establish that the relative rank measure with respect to symmetric words is (suitably) small for low-depth and general-depth set-multilinear formulas, respectively.

[14] We only need d to be larger than some absolute constant.

▶ **Lemma 11** ([22]). *Let C be a set-multilinear formula of product-depth $1 \leq \Delta \leq \log d/\log\log d$ of size at most s which computes a polynomial (over any fixed field) that is set-multilinear with respect to the partition $(X_1, \dots, X_d)$ where each $|X_i| = n$. Let $w \in \{k, -k\}^d$ be chosen uniformly at random. Then, we have*

$$\mathrm{relrk}_w(C) \leq s \cdot 2^{-\frac{kd^{1/\Delta}}{20}}$$

with probability at least $1 - s \cdot d^{-\frac{d^{1/\Delta}}{12\Delta}}$.

▶ **Lemma 12** ([22]). *Let F be a set-multilinear formula of size at most s which computes a polynomial (over any fixed field) that is set-multilinear with respect to the partition $(X_1, \dots, X_d)$ where each $|X_i| = n$. Let $w \in \{k, -k\}^d$ be chosen uniformly at random. Then, we have*

$$\mathrm{relrk}_w(F) \leq s \cdot 2^{-\frac{k\log d}{20}}$$

with probability at least $1 - s \cdot d^{-\frac{\log d}{60}}$.

Next, we shall show that the hard polynomial $F_{n,d}$ in Theorem 10 has high relative rank (in fact, the maximum possible value – 1) with respect to a symmetric word. For this, we consider an alternate view of these polynomials and require the following notion. For an even integer d, define Dyck(d) to be the collection of all strings (called *Dyck words*) of length d over symbols '(' and ')' that are well-matched in the natural way. More precisely, it is the collection of all strings u of length d such that all prefixes of u contain no more)'s than ('s and the total number of ('s in u equals the total number of). For example, "()()" and "(())" belong to Dyck(4) but not "(()(". Note that for any '(' appearing in a Dyck word, there is a *unique* ')' which "closes" it. Given a Dyck word $u \in$ Dyck(d), we call (i, j) a *matching parenthesis pair* of u if there is '(' in the i-th position of u that is closed by a ')' in the j-th position of u (clearly then, $j - i > 0$ must be odd).

Given a string $u \in$ Dyck(d) and the setup above for defining the polynomials $f_{i,j}$, we can associate to u a *product of inner products* polynomial $IP_u \in \mathbb{F}_{\mathrm{sm}}[X_i, \dots, X_j]$ in the natural way: define IP_u to be the product of all $X_i \cdot X_j$ where (i, j) is a matching parenthesis pair of u. For example, the strings "()()" and "(())" would correspond to the polynomials $(X_1 \cdot X_2) \cdot (X_3 \cdot X_4)$ and $(X_1 \cdot X_4) \cdot (X_2 \cdot X_3)$ respectively. We define y_u analogously: it is the product of all $y_{i,j} \cdot y_{j,i}$ where (i, j) is a matching parenthesis pair of u. So, if $u =$ "(())" $\in$ Dyck(4), then $y_u = y_{1,4}y_{4,1}y_{2,3}y_{3,2}$. The following observation then follows immediately from the recursive definition of $f_{i,j}$.

▶ **Observation 13.** *For every interval $[i, j] \subseteq [d]$ where $j - i$ is odd, there exist constants $c_u \in \mathbb{F}$ corresponding to every $u \in$ Dyck$(j - i + 1)$ such that $f_{i,j}(X, Y) = \sum_{u \in \mathrm{Dyck}(j-i+1)} c_u y_u IP_u$*[15]*. Moreover, if $\mathbb{F}$ has characteristic zero, then every $c_u \neq 0$.*

▷ Claim 14. Let d be a positive even integer. For any $w \in \{-k, k\}^d$ with $w_{[d]} = 0$ (i.e., $|\mathcal{P}_w| = |\mathcal{N}_w|$), there exists a Dyck work $u \in$ Dyck(d) such that for every matching parenthesis pair (i, j) of u, either $i \in \mathcal{P}_w$ and $j \in \mathcal{N}_w$, or $i \in \mathcal{N}_w$ and $j \in \mathcal{P}_w$.

Proof. We prove this by induction on d. The base case $d = 2$ is trivial as there is only a single matching parenthesis pair $(1, 2)$ for which the given condition must indeed hold. Now, suppose $d > 2$ and $w \in \{-k, k\}^d$ is a given word with $w_{[d]} = 0$. Let us refer to $\mathcal{P}_w$ and $\mathcal{N}_w$ as the two "parts" of w and take cases on the membership of 1 and d in these sets.

[15] Strictly speaking, the indices within IP_u and y_u here need to be "translated" appropriately to suit the interval $[i, j]$ (which may not necessarily be $[1, j - i + 1]$).

Case 1: 1 and d are in different parts. By the induction hypothesis, there is a Dyck word $u' \in \mathrm{Dyck}(d-2)$ (corresponding to the subset $\{2, \ldots, d-1\}$ of indices) for which the desired condition holds. Hence, we can simply define u to be the string "(u')" and the claim follows.

Case 2: 1 and d are in the same part. Notice that there exists an index r such that both $w_{[1,r]}$ and $w_{[r+1,d]}$ are 0. Then, we can define u to be the concatenation of the two Dyck words that the induction hypothesis yields for the intervals $[1, r]$ and $[r+1, d]$ respectively and the claim follows. $\lhd$

▶ **Lemma 15.** *Let $n = 2^k$ and $d \leq n$ be an even integer. Over any field $\mathbb{F}$ of characteristic zero, the polynomial $F_{n,d} = f_{1,d} \in \mathbb{F}_{\mathrm{sm}}[X, Y]$ as defined above satisfies the following: For any $w \in \{-k, k\}^d$ with $w_{[d]} = 0$, $\mathcal{M}_w(F_{n,d})$ is full-rank when viewed as a matrix over the field $\mathbb{F}(Y)$, the field of rational functions over the Y variables.*

Proof. Fix a word $w \in \{-k, k\}^d$ with $w_{[d]} = 0$ and let $s \in \mathrm{Dyck}(d)$ be a Dyck word as given by Claim 14. By Observation 13, we know that $F = f_{1,d}$ has the form $\sum_{u \in \mathrm{Dyck}(d)} c_u y_u IP_u$. Consider the polynomial f obtained by plugging in $y_{i,j} = y_{j,i} = 0$ for every i, j such that (i, j) is *not* a matching parenthesis pair of s, and $y_{i,j} = y_{j,i} = 1$ for every i, j such that (i, j) *is* a matching parenthesis pair of s. Observe that the only surviving term from F in f is the one indexed by s. Therefore, to argue that $\mathcal{M}_w(F)$ is full-rank over $\mathbb{F}(Y)$[16], it suffices to show that $\mathcal{M}_w(c_s IP_s)$ is full-rank. As $c_s \neq 0$ by Observation 13, this follows from Observation 8 (the matrix of the inner product gadget has full rank), Claim 14 (w "splits" every matching parenthesis pair of s), and Claim 7 (the multiplicativity of relrk_w). ◀

Let us return to the proof of the last two items of Theorem 10. Let C be a set-multilinear formula of product depth Δ of size s computing $F_{n,d}(X)$ (now interpreted as a formula over the field $\mathbb{F}(Y)$). Suppose $s < d^{\frac{d^{1/\Delta}}{24\Delta}}$. Then, by Lemma 11, with probability at least $1 - d^{-\frac{d^{1/\Delta}}{24\Delta}}$,

$$\mathrm{relrk}_w(C) \leq s \cdot 2^{-\frac{kd^{1/\Delta}}{20}}.$$

But now, we can condition on the event that $w_{[d]} = 0$ (which occurs with probability $\Theta(\frac{1}{\sqrt{d}})$) to establish the existence of a word $w \in \{-k, k\}^d$ with $w_{[d]} = 0$ such that w satisfies $\mathrm{relrk}_w(C) \leq s \cdot 2^{-\frac{kd^{1/\Delta}}{20}}$. This is because of the asymptotic bound $\frac{1}{\sqrt{d}} \gg d^{-\frac{d^{1/\Delta}}{24\Delta}}$, which follows from the given constraints on the parameters d, Δ. Therefore, by Lemma 15,

$$s \geq 2^{\frac{kd^{1/\Delta}}{20}} \cdot \mathrm{relrk}_w(C) = n^{\frac{d^{1/\Delta}}{20}}$$

which contradicts the assumption that $s < d^{\frac{d^{1/\Delta}}{24\Delta}}$. Thus, we conclude that $s \geq d^{\frac{d^{1/\Delta}}{24\Delta}} = d^{\Omega(d^{1/\Delta}/\Delta)}$.

Similarly, to see the final item of Theorem 10, let F be a set-multilinear formula of size s computing $F_{n,d}$ (now interpreted as a formula over the field $\mathbb{F}(Y)$). Suppose $s < d^{\frac{\log d}{120}}$. Then, by Lemma 12, with probability at least $1 - d^{-\frac{\log d}{120}}$,

$$\mathrm{relrk}_w(F) \leq s \cdot 2^{-\frac{k \log d}{20}}.$$

[16] We need to show that its determinant – a polynomial in $\mathbb{F}[Y]$ – is non-zero.

But now, we can condition on the event that $w_{[d]} = 0$ (which occurs with probability $\Theta(\frac{1}{\sqrt{d}}))$ to establish the existence of a word $w \in \{-k, k\}^d$ with $w_{[d]} = 0$ such that w satisfies $\text{relrk}_w(F) \le s \cdot 2^{-\frac{k \log d}{20}}$. This is because of the trivial asymptotic bound $\frac{1}{\sqrt{d}} \gg d^{-\frac{\log d}{120}}$. Therefore, again by Lemma 15,

$$s \ge 2^{\frac{k \log d}{20}} \cdot \text{relrk}_w(F) = n^{\frac{\log d}{20}}$$

which contradicts the assumption that $s < d^{\frac{\log d}{120}}$. Thus, we conclude that $s \ge d^{\frac{\log d}{120}} = d^{\Omega(\log d)}$.

4 A Hard Set-multilinear Polynomial in VBP

4.1 Arc-partition Measure Description

This subsection is adapted from Section 2 of [8]. Let $n = 2^k$, $d \le n$ be an even integer, and let $X = (X_1, X_2, \ldots, X_d)$ be a collection of disjoint sets of n variables each. An *arc-partition* will be a special kind of *symmetric* word $w \in \{-k, k\}^d$ (i.e., a one-to-one map Π from X to $\{-k, k\}^d$). For the purpose of this subsection, the reader can even choose to think of the alphabet of w as $\{-1, 1\}$ (i.e., one "positive" and one "negative" value) – we use $k, -k$ only to remain consistent with Definition 5.

Identify X with the set $\{1, 2, \ldots, d\}$ in the natural way. Consider the d-cycle graph, i.e., the graph with nodes $\{1, 2, \ldots, d\}$ and edges between i and $i + 1$ modulo d. For two nodes $i \neq j$ in the d-cycle, denote by $[i, j]$ the arc between i, j, that is, the set of nodes on the path $\{i, i + 1, \ldots, j - 1, j\}$ from i to j in d-cycle. First, define a distribution $\mathcal{D}_P$ on a family of pairings (a list of disjoint pairs of nodes in the cycle) as follows. A random pairing is constructed in $d/2$ steps. At the end of step $t \in [d/2]$, we shall have a pairing $(P_1, \ldots, P_t)$ of the arc $[L_t, R_t]$. The size of $[L_t, R_t]$ is always $2t$. The first pairing contains only $P_1 = \{L_1, R_1\}$ with $L_1 = 1$ and $R_1 = 2$. Given $(P_1, \ldots, P_t)$ and $[L_t, R_t]$, define the random pair $P_t + 1$ (independently of previous choices) by

$$P_{t+1} = \begin{cases} \{L_t - 2, L_t - 1\} & \text{with probability } 1/3 \\ \{L_t - 1, R_t + 1\} & \text{with probability } 1/3 \\ \{R_t + 1, R_t + 2\} & \text{with probability } 1/3 \end{cases}$$

Define

$$[L_{t+1}, R_{t+1}] = [L_t, R_t] \cup P_{t+1}.$$

So, L_{t+1} is either $L_t - 2$, $L_t - 1$ or L_t, each value is obtained with probability $1/3$, and similarly (but not independently) for R_{t+1}.

The final pairing is $P = (P_1, P_2, \ldots, P_{d/2})$. Denote by $P \sim \mathcal{D}_P$ a pairing distributed according to $\mathcal{D}_P$.

Once a pairing P has been obtained, a word $w \in \{-k, k\}^d$ is obtained by simply randomly assigning $+k$ and $-k$ to the indices of any pair P_i. More formally, for every $t \in [n/2]$, if $P_t = \{i_t, j_t\}$, let with probability $1/2$, independently of all other choices,

$$w_{i_t} = +k \text{ and } w_{j_t} = -k,$$

and with probability $1/2$,

$$w_{i_t} = -k \text{ and } w_{j_t} = +k.$$

Denote by $w \sim \mathcal{D}$ a word in $\{-1, 1\}^n$ that is sampled using this procedure. We call such a word an *arc-partition*. For a pair $P_t = \{i_t, j_t\}$, we refer to i_t and j_t as *partners*.

▶ **Definition 16** (Arc-full-rank). *We say that a polynomial f that is set-multilinear over $X = (X_1, \ldots, X_d)$ is **arc-full-rank** if for every arc-partition $w \in \{-k, k\}^d$,* $\text{relrk}_w(f) = 1$.

4.2 Construction of an Arc-full-rank Polynomial

Below, we describe a simple construction of an ABP that computes an arc-full-rank set-multilinear polynomial. The high-level idea is to construct an ABP in which every path between start-node and end-node corresponds to a specific execution of the random process which samples arc-partitions. Each node in the ABP corresponds to an arc $[L, R]$, which sends an edge to each of the nodes $[L-2, R], [L-1, R+1]$ and $[L, R+2]$. The edges have specially chosen labels that help guarantee full rank with respect to every arc-partition. For simplicity of presentation, we allow the edges of the program to be labeled by degree three polynomials in three variables. This assumption can be easily removed by replacing each edge with a constant-size ABP computing the corresponding degree three polynomial.

Formally, the nodes of the program are even-size arcs in the d-cycle, d an even integer. The start-node of the program is the empty arc $\emptyset$ and the end-node is the whole cycle $[d]$ (both are "special" arcs). Let $X = (X_1, \ldots, X_d)$ be a collection of sets of variables where each $|X_i| = n$, and similarly, let $Y = (Y_1, \ldots, Y_d)$ be a distinct collection of sets of variables where each $|Y_i| = n$ (we shall refer to the Y-variables as *auxiliary* variables). For i and j in $\{1, \ldots, d\}$, let $X_i \cdot X_j$ denote the inner-product quadratic form $\sum_{k=1}^{n} x_{ik}x_{jk}$. Here, we shall assume that $X_i = \{x_{i,1}, \ldots, x_{i,n}\}$ and $Y_i = \{y_{i,1}, \ldots, y_{i,n}\}$.

Construct the branching program by connecting a node/arc of size $2t$ to three nodes/arcs of size $2t+2$. For $t = 1$, there is just one node $[1, 2]$, and the edge from start-node to it is labeled $y_{1,2}y_{2,1}(X_0 \cdot X_1)$. For $t > 1$, the node $[L, R] \supset [1, 2]$ of size $2t < d$ is connected to the three nodes: $[L-2, R], [L-1, R+1]$, and $[L, R+2]$. (It may be the case that the three nodes are the end-node.) The edge labeling is:

- The edge between $[L, R]$ and $[L-2, R]$ is labeled $y_{L-2,L-1}y_{L-1,L-2}(X_{L-2} \cdot X_{L-1})$.
- The edge between $[L, R]$ and $[L-1, R+1]$ is labeled $y_{L-1,R+1}y_{R+1,L-1}(X_{L-1} \cdot X_{R+1})$.
- The edge between $[L, R]$ and $[L, R+2]$ is labeled $y_{R+1,R+2}y_{R+2,R+1}(X_{R+1} \cdot X_{R+2})$.

Consider the ABP thus described, and the polynomial $G = G_{n,d}$ it computes. For every path γ from start-node to end-node in the ABP, the list of edges along γ yields a pairing P; every edge e in γ corresponds to a pair $P_e = \{i_e, j_e\}$ of nodes in d-cycle. Thus,

$$G = \sum_{\gamma} \prod_{e=\{i_e,j_e\}\in\gamma} y_{i_e,j_e}y_{j_e,i_e} \cdot (X_{i_e} \cdot X_{j_e}). \tag{1}$$

where the sum is over all paths γ from start-node to end-node.

▶ Remark 17. There is in fact a one-to-one correspondence between pairings P and such paths γ (this follows by induction on t). Note that this is true only because pairings are tuples i.e., they are *ordered* by definition. Otherwise, it is of course still possible to obtain the same *set* of pairs in a given pairing using multiple different orderings. The sum defining G can be thought of, therefore, as over pairings P.

The following statement summarizes the main useful property of G.

▶ **Lemma 18.** *Over any field $\mathbb{F}$ of characteristic zero, the polynomial $G = G_{n,d}$ defined above is arc-full-rank as a set-multilinear polynomial in the variables X over the field $\mathbb{F}(Y)$ of rational functions in Y.*

Proof. Let $w \sim \mathcal{D}$ be an arc-partition. We want to show that $\mathcal{M}_w(G)$ has full rank. The arc-partition w is defined from a pairing $P = (P_1, \ldots, P_{d/2})$ (though as discussed in Remark 17, there could be multiple such P). The pairing P corresponds to a path γ from start-node to end-node. Consider the polynomial f that is obtained by setting every $y_{i,j} = y_{j,i} = 0$ in F

such that $\{i, j\}$ is not a pair in P, and setting every $y_{i,j} = y_{j,i} = 1$ for every pair $\{i, j\}$ in P. Then, it is easy to see that the only terms that survive in (1) correspond to paths (and in turn, pairings) which have the same underlying *set* of pairs as P. As a consequence, f is simply some non-zero constant times a polynomial which is full-rank. $M_w(f)$ being full rank then implies that $M_w(G)$ is also full-rank[17]. ◀

4.3 Bounding relrk_w for Small Set-multilinear Formulas

As discussed in Section 1, the high-level strategy to prove Theorem 1 is to show that the relative rank (with respect to arc-partitions) of our hard polynomial is large (as already established in Lemma 18), while it is small for (small enough) set-multilinear formulas. The remainder of the section is devoted to establishing the latter. Before moving on to it, we shall first state the following more precise and general version of Theorem 1. We also incorporate Remark 2 here and show our lower bound for any degree $d \le n$. Theorem 1 follows from the special case $d = n$.

▶ **Theorem 19.** *Let $n = 2^k$, and suppose $d \le n$ be an even integer that is large enough[18], and $1 \le \Delta \le \log d/ \log\log d$ be any integer. Let X_i, Y_i denote the sets of n variables $\{x_{i,j} : j \in [n]\}$ and $\{y_{i,j} : j \in [n]\}$ respectively and let X, Y be the tuples $(X_1, \ldots, X_d)$ and $(Y_1, \ldots, Y_d)$. Then,*

- *there is a $\mathrm{poly}(n, d)$-size branching program computing $G_{n,d}$ as defined above whose every internal node computes a set-multilinear polynomial,*
- *any set-multilinear formula of product-depth Δ computing $G_{n,d}$ must have size at least $d^{\Omega(d^{1/\Delta}/\Delta)}$.*
- *Further, any set-multilinear formula of arbitrary product-depth computing $G_{n,d}$ must have size at least $d^{\Omega(\log d)}$.*

The following couple of lemmas formalize the high-level idea mentioned before the statement of Theorem 19 – they correspond to the low-depth case and general depth case respectively. Most of the remainder of this section is devoted to the proof of Lemma 20; Lemma 22 has a similar (and in fact, easier) proof and for this reason, we only provide a sketch that is deferred to the appendix.

▶ **Lemma 20.** *Let $\mathbb{K}$ be any field and let $X_1, \ldots, X_D$ be sets of n distinct variables each. Let C be a set-multilinear formula over $\mathbb{K}$ of constant product-depth $\Delta \ge 1$ of size at most s which computes a polynomial over $\mathbb{K}$ that is set-multilinear with respect to the partition $(X_{i_1}, \ldots, X_{i_d})$ where $1 \le i_1 < \cdots < i_d \le D$ and each $|X_i| = n$. Let $w \sim \mathcal{D}$ be an arc-partition sampled from $\{-k, k\}^D$. Then, we have*

$$\mathrm{relrk}_w(C) \le s \cdot 2^{-\frac{kd^{1/\Delta}}{2000}}$$

with probability at least $1 - s \cdot d^{-\frac{d^{1/\Delta}}{10^7\Delta}}$.

▶ Remark 21. Note that in the statement above, we are abusing notation and overloading the relrk_w notation – assume that $\mathrm{relrk}_w(C)$ is defined in the obvious projective manner i.e., if $S = \{i_1, \ldots, i_d\}$, then $\mathrm{relrk}_w(C) := \mathrm{relrk}_{w|_S}(C)$ where $w|_S$ is as defined in Definition 6.

[17] This argument is the same as in the proof of Lemma 15.
[18] We only need d to be larger than some absolute constant.

▶ **Lemma 22.** *Let $\mathbb{K}$ be any field and let $X_1, \ldots, X_d$ be sets of n distinct variables each. Let F be a set-multilinear formula over $\mathbb{K}$ of size at most s which computes a polynomial over $\mathbb{K}$ that is set-multilinear with respect to the partition $(X_1, \ldots, X_d)$ where each $|X_i| = n$. Let $w \sim \mathcal{D}$ be an arc-partition sampled from $\{-k, k\}^d$. Then, we have*

$$\mathrm{relrk}_w(F) \leq s \cdot 2^{-\frac{k \log d}{2000}}$$

with probability at least $1 - s \cdot d^{-\frac{\log d}{10^7 \Delta}}$.

Before moving on to the technical core of this section (the proof of Lemma 20), let us finish the proof of Theorem 19.

Proof of Theorem 19 given Lemmas 20 and 22. Note that the first item follows immediately from the definition of $G_{n,d}$ (see (1)). Let us prove the last two items of Theorem 19. Let C be a set-multilinear formula of product depth Δ of size s computing $G_{n,d}(X)$ (now interpreted as a formula over the field $\mathbb{F}(Y)$). Suppose $s < d^{\frac{d^{1/\Delta}}{2\times 10^7 \Delta}}$. Then, by Lemma 20, for an arc-partition $w \sim \mathcal{D}$ sampled from $\{-k, k\}^d$, it follows that with probability at least $1 - d^{-\frac{d^{1/\Delta}}{2\times 10^7 \Delta}}$,

$$\mathrm{relrk}_w(C) \leq s \cdot 2^{-\frac{kd^{1/\Delta}}{2000}}.$$

Fix such an arc-partition w. By Lemma 18, we have

$$s \geq 2^{\frac{kd^{1/\Delta}}{2000}} \cdot \mathrm{relrk}_w(C) = n^{\frac{d^{1/\Delta}}{2000}}$$

which contradicts the assumption that $s < d^{\frac{d^{1/\Delta}}{2\times 10^7 \Delta}}$. Thus, we conclude that $s \geq d^{\frac{d^{1/\Delta}}{2\times 10^7 \Delta}} = d^{\Omega(d^{1/\Delta}/\Delta)}$.

Similarly, to see the final item of Theorem 19, let F be a set-multilinear formula of size s computing $G_{n,d}$ (now interpreted as a formula over the field $\mathbb{F}(Y)$). Suppose $s < d^{\frac{\log d}{2\times 10^7}}$. Then, for an arc-partition $w \sim \mathcal{D}$ sampled from $\{-k, k\}^d$, by Lemma 22, with probability at least $1 - d^{-\frac{\log d}{2\times 10^7}}$,

$$\mathrm{relrk}_w(F) \leq s \cdot 2^{-\frac{k \log d}{2000}}.$$

Fix such an arc-partition w. Therefore, again by Lemma 18,

$$s \geq 2^{\frac{k \log d}{2000}} \cdot \mathrm{relrk}_w(F) = n^{\frac{\log d}{2000}}$$

which contradicts the assumption that $s < d^{\frac{\log d}{2\times 10^7}}$. Thus, we conclude that $s \geq d^{\frac{\log d}{2\times 10^7}} = d^{\Omega(\log d)}$. ◀

The essential ingredient in the proof of Lemma 20 is a combinatorial proposition which we will call the "many violations lemma". As alluded to in Section 1.5, this is a modification of a corresponding statement in [8] (Lemma 4.1). However, because we are working in the low-depth setting (as opposed to [8]) and because we are seeking such strong and near-optimal lower bounds, we need to make significant changes – this includes introducing new conceptual arguments to tighten the analysis. To state this lemma, we shall reproduce some of the definitions made in Section 4 of [8].

Again, we identify the set of variables $X = (X_1, \ldots, X_D)$ with the D-cycle $\{1, \ldots, D\}$, where addition is modulo D. Let S be a collection of disjoint subsets of the cycle to K parts, namely, $S = (S_1, \ldots, S_K)$ where each $S_k \subset \{1, \ldots, D\}$ and $S_k \cap S_{k'} = \emptyset$ for all $k \neq k'$ in $[K]$.

We also think of $[K]$ as a set of colors, and of S as a (partial) K-coloring of some d nodes of the cycle, where $d = |S_1| + \cdots + |S_K|$. We shall refer to the nodes in the D-cycle outside of S as *uncolored.*

For a pairing P, define the number of k-violations by

$$V_k(P) = \{P_t \in P : |P_t \cap S_k| = 1\}.$$

In words, it is the set of pairs in which one color is k and the other color is different. Fix $\varepsilon = 1/1000$ and denote

$$G(P) = \{k \in [K] : |V_k(P)| \geq d^{\varepsilon}\}.$$

We do not include S as a subscript in these two notations since S will be known from the context (and will be fixed throughout most of the discussion). The next crucial lemma shows that for every fixed non-redundant K-coloring of the cycle, a random pairing has, with high probability, many colors with many violations.

▶ **Lemma 23** (Many Violations Lemma). *For all large enough d and for all integers K in the range $[2d^{\frac{1}{\Delta+1}}/3, 2d^{\frac{1}{\Delta+1}}]$ the following holds: Let $S = (S_1, \ldots, S_K)$ be a collection of disjoint subsets of the D-cycle and suppose that $|S_k| \geq d^{\frac{\Delta}{\Delta+1}}/2$ for all $k \in [K]$. Then,*

$$\mathbb{P}[G(P) \leq K/1000] \leq d^{-K/500},$$

where $P \sim \mathcal{D}_P$.

▶ Remark 24. Other than the differences in parameter ranges, one key difference between the statement above from Lemma 4.1 in [8] is the loosening of the requirement that S be a *partition* of the D-cycle. Note that here, we only demand that S be a collection of disjoint subsets (i.e., some nodes are allowed to remain *uncolored*) – this requirement is indeed key for the inductive proof of Lemma 20 to go through.

Before proving Lemma 23, let us next see that the many violations lemma suffices to prove the relative rank upper bound on low-depth set-multilinear formulas.

Proof of Lemma 20 given Lemma 23. We prove the statement by induction on Δ. Identify the set $\{i_1, \ldots, i_d\}$ with $[d]$.

If $\Delta = 1$, then $C = C_1 + \cdots + C_t$ where each C_i is a product of linear forms. So, for all $i \in [t]$, by Claim 7,

$$\mathrm{relrk}_w(C_i) = \prod_{i=1}^{d} 2^{-\frac{1}{2}|w_j|} = 2^{-\frac{kd}{2}}$$

where in the last step, we used the observation that regardless of the choice of w, $|w_j| = k$ for all $j \in [n]$. Hence, by the sub-additivity of relrk_w, with probability 1, we have

$$\mathrm{relrk}_w(C) \leq s \cdot 2^{-\frac{kd}{2}} \leq s \cdot 2^{-\frac{kd}{2000}}.$$

Next, we assume the statement is true for all formulas of product-depth $\leq \Delta$. Let C be a formula of product-depth $\Delta + 1$. So, C is of the form $C = C_1 + \cdots + C_t$. Using a similar terminology to that in [23] and [22], we say that a sub-formula C_i of size s_i is of type 1 if one of its factors has degree at least $T_\Delta = d^{\frac{\Delta}{\Delta+1}}$, otherwise we say it is of type 2.

Suppose $C_i = C_{i,1} \cdot \dots \cdot C_{i,t_i}$ is of type 1 with, say, $C_{i,1}$ having degree at least T_Δ. Let $w^{i,1}$ be the corresponding word i.e., $w^{i,1} = w|_{S_1}$ if $C_{i,1}$ is set-multilinear with respect to $S_1 \subsetneq [d]$. If it has size $s_{i,1}$, then since it has product-depth at most Δ, it follows by induction that

$$\mathrm{relrk}_w(C_i) \leq \mathrm{relrk}_{w^{i,1}}(C_{i,1}) \leq s_{i,1} \cdot 2^{-\frac{kT_\Delta^{1/\Delta}}{2000}} \leq s_i \cdot 2^{-\frac{kd^{1/(\Delta+1)}}{2000}}$$

with probability at least

$$1 - s_{i,1} \cdot T_\Delta^{-\frac{T_\Delta^{1/\Delta}}{10^7 \Delta}} \geq 1 - s_i \cdot d^{-\frac{d^{1/(\Delta+1)}}{10^7 \Delta} \cdot \frac{\Delta}{\Delta+1}} = 1 - s_i \cdot d^{-\frac{d^{1/(\Delta+1)}}{10^7(\Delta+1)}}.$$

Now suppose that $C_i = C_{i,1} \cdot \dots \cdot C_{i,t_i}$ is of type 2 i.e., each factor $C_{i,j}$ has degree $< T_\Delta$. Note that this forces $t_i > d/T_\Delta = d^{\frac{1}{\Delta+1}}$. As the formula is set-multilinear, $(S_1, \dots, S_{t_i})$ form a partition of $[d]$ where each $C_{i,j}$ is set-multilinear with respect to $(X_\ell)_{\ell \in S_j}$ and C_i is set-multilinear with respect to $(X_\ell)_{\ell \in S}$. Let $w^{i,1}, \dots, w^{i,t_i}$ be the corresponding decomposition, whose respective sums are denoted simply by $w_{S_1}, \dots, w_{S_{t_i}}$.

From the properties of relrk_w (Claim 7), we have

$$\mathrm{relrk}_w(C_i) = \prod_{j=1}^{t_i} \mathrm{relrk}_{w^{i,j}}(C_{i,j}) \leq \prod_{j=1}^{t_i} 2^{-\frac{1}{2}|w_{S_j}|} = 2^{-\frac{1}{2}\sum_{j=1}^{t_i} |w_{S_j}|},$$

from which we observe that the task of upper bounding $\mathrm{relrk}_w(C)$ can be reduced to the task of lower bounding the sum $\sum_{j=1}^{t_i} |w_{S_j}|$, which is established in the following claim. For the sake of convenience, the choice of the alphabet for w below is scaled down to $\{-1, 1\}$.

▷ Claim 25. For large enough d, suppose $(S_1, \dots, S_K)$ is a partition of $[d]$ such that each $|S_j| < T_\Delta = d^{\frac{\Delta}{\Delta+1}}$. Then, we have

$$\mathop{\mathbb{P}}_{w \sim \mathcal{D}} \left[\sum_{j=1}^{K} |w_{S_j}| < \frac{d^{1/(\Delta+1)}}{2000} \right] \leq d^{-\frac{d^{1/(\Delta+1)}}{10^7}}.$$

Here, $\mathcal{D}$ refers to the original distribution i.e., an arc-partition over the D-cycle.

Proof. We first show that without loss of generality, we may assume that each S_j has size "roughly" T_Δ. To see this, we apply the following *clubbing* procedure to the sets in the partition $(S_1, \dots, S_K)$:

- Start with the given partition $(S_1, \dots, S_K)$. At each step in the procedure, we shall "club" two of the sets in the partition according to the following rule.
- If there are two distinct sets S' and S'' in the current partition each of size $< T_\Delta/2$, we remove both of them and add their union $S' \cup S''$ to the partition.
- If the rule above is no longer applicable, then we have at most one set in the current partition of size $< T_\Delta/2$. If there is none, then we halt the procedure here. Otherwise, we union this set with any one of the other sets and then halt.

After the clubbing procedure, we are left with a partition $(S'_1, \dots, S'_{K'})$ of $[d]$ such that $\frac{T_\Delta}{2} \leq |S'_j| \leq \frac{3T_\Delta}{2}$ for each $j \in [K']$, also implying that $\frac{2d^{1/(\Delta+1)}}{3} \leq K' \leq 2d^{1/(\Delta+1)}$. Through a repeated use of the triangle inequality, we see that $\sum_{j=1}^{K'} |w_{S'_j}| \leq \sum_{j=1}^{K} |w_{S_j}|$. Therefore, upper bounding the latter sum is a "smaller" event than upper bounding the former sum. Hence, it suffices to prove the statement of the claim with the assumption that $\frac{T_\Delta}{2} \leq |S_j| \leq \frac{3T_\Delta}{2}$ for each $j \in [K]$ (we henceforth drop the primed notation).

Applying Lemma 23 to the tuple $(S_1, \dots, S_K)$, we obtain that

$$\mathbb{P}[G(P) \leq K/1000] \leq d^{-K/500}.$$

The idea is to condition on the high probability event that $G(P) > K/1000$. Fix a pairing P with this property. Consider an ordering σ of the colors in $G(P)$. A color k is said to be *bright* with respect to an ordering if there are at least $d^{\varepsilon}/2$ nodes x of color k such that either the partner of x is uncolored or its partner is colored using a color that appears *after* k in the ordering σ. Call an ordering σ of the nodes in $G(P)$ *good* if there are at least $|G(P)|/2$ bright colors with respect to σ. The observation is that for any ordering σ of the colors, either σ itself is good, or its reverse is good. We conclude that given any pairing P, there exists a good ordering of $G(P)$. Fix any such good ordering and let $H(P)$ be the collection of bright colors with respect to this ordering.

Next, notice that if the sum $\sum_{j=1}^{K} |w_{S_j}|$ is at most $\frac{d^{1/(\Delta+1)}}{2000}$, then so is the sum $\sum_{k \in H(P)} |w_{S_k}|$. Let $K' = |H(P)|$ (which is at least $K/2000$ if $G(P) > K/1000$). View the sampling of Π from P as happening in a specific order, according to the order of $k_1, k_2, \dots, k_{K'}$: First define Π on pairs with at least one point with color k_1, then define Π on remaining pairs with at least one point with color k_2, and so forth. When finished with $k_1, \dots, k_{K'}$, continue to define Π on all other pairs.

Conditioned on the event that $G(P) > K/1000$, this implies that $|w_{S_j}| \leq 1$ for each $j \in H(P)$. For every $j \in H(P)$, define E_j to be the event that $|w_{S_{k_j}}| \leq 1$. By choice, conditioned on $E_1, \dots, E_{j-1}$, there are at least $d^{\varepsilon}/2$ pairs P_t so that $|P_t \cap S_{k_j}| = 1$ that are not yet assigned a "positive" or "negative" sign. For every such P_t, the element in $P_t \cap S_{k_j}$ is assigned a positive sign with probability $1/2$, and is independent of any other $P_{t'}$. The probability that a binomial random variable B over a universe of size $U \geq d^{\varepsilon}/2$ and marginals $1/2$ obtains any specific value is at most $O(U^{-1/2}) = O(d^{-\varepsilon/2})$. Hence, for all $j \in H(P)$, by the union bound,

$$\mathbb{P}[E_j | E_1, \dots, E_{j-1}, P] \leq \mathbb{P}_{B}[U/2 - 1 \leq B \leq U/2 + 1] \leq O(3 \cdot d^{-\varepsilon/2}) \leq d^{-\varepsilon/4}.$$

Therefore,

$$\mathbb{P}[|w_{S_{k_j}}| \leq 1 \text{ for all } j \in H(P)] \leq \mathbb{E}[d^{-\varepsilon|H(P)|/4} | G(P) > K/1000] + d^{-K/500} \leq d^{-K/10^7}.$$

Finally, we note that

$$\mathbb{P}_{w \sim \mathcal{D}}\left[\sum_{j=1}^{K} |w_{S_j}| < \frac{d^{1/(\Delta+1)}}{2000}\right] \leq \mathbb{P}[|w_{S_{k_j}}| \leq 1 \text{ for all } j \in H(P)]. \qquad \triangleleft$$

The claim above and the preceding calculation immediately implies that for a sub-formula C_i of type 2,

$$\mathrm{relrk}_w(C_i) \leq s_i \cdot 2^{-\frac{kd^{1/(\Delta+1)}}{2000}}$$

with probability at least $1 - d^{-\frac{d^{1/(\Delta+1)}}{10^7}} \geq 1 - s_i \cdot d^{-\frac{d^{1/(\Delta+1)}}{10^7(\Delta+1)}}$.

Next, by a union bound over $i \in [t]$ and the sub-additivity property of relrk_w, it follows that

$$\mathrm{relrk}_w(C) \leq \mathrm{relrk}_w(C_1) + \dots + \mathrm{relrk}_w(C_t) \leq s_1 \cdot 2^{-\frac{kd^{1/(\Delta+1)}}{2000}} + \dots + s_t \cdot 2^{-\frac{kd^{1/(\Delta+1)}}{2000}} = s \cdot 2^{-\frac{kd^{1/(\Delta+1)}}{2000}}$$

with probability at least $1 - s \cdot d^{-\frac{d^{1/(\Delta+1)}}{10^7(\Delta+1)}}$, which concludes the proof of the lemma. ◀

4.4 Proof of the Many Violations Lemma

Fix some collection of disjoint subsets (or a "partial" coloring) $S = (S_1, \dots, S_K)$ of the D-cycle satisfying the conditions of the lemma. Think of S as a partial function from the D-cycle to the set $[K]$, either assigning a node its color in $[K]$ or leaving it uncolored; $S(i)$ is the color of i. Use the following definition to partition the proof into cases. For a color k, count the number of jumps in it (with respect to the partition S) to be

$$J_k = \{j \in S_k : k = S(j) \neq S(j+1)\},$$

the set of elements j of color k so that $j+1$ is either uncolored or has a color different from k. As mentioned previously, this subsection is adapted from the proof of Lemma 4.1 in [8]. In what follows, we include remarks where we require a more refined analysis than [8] or a different argument to suit the parameter demands of Lemma 23. Overall, we have attempted to provide a more comprehensive and complete exposition to the proof of the many violations lemma.

Case 1: Many colors with many jumps

The high-level idea is that each color with many jumps has many violations because pairs of the form $(j, j+1)$ yield violations as soon as they are constructed.

Assume that for at least $K/2$ colors k, $|J_k| > d^{2\varepsilon}$. Denote by $B \subseteq [K]$ the set of k's that satisfy this inequality. Then, for every k in B, there exists a subset $Q_k \subset J_k$ of size $N = \lceil d^{2\varepsilon} \rceil$. Let

$$Q \coloneqq \bigcup_{k \in B} Q_k.$$

We think of the construction of the (random) pairing P as happening in *epochs*, depending on Q, as follows.

For $t > 0$, define the random variable

$$Q(t) = Q \setminus [L_t - 4, R_t + 4],$$

the set Q after removing a four-neighborhood of $[L_t, R_t]$. For a certain sequence of time steps t, we will define special nodes q_t which lie in this small "cloud" around the arc $[L_t, R_t]$ (i.e., within a distance of 4 on either side of the arc) - it is for these special nodes q_t that the set of pairs (q_t, q_{t+1}) will provide many violations. We now formalize this intuition.

Let $\tau_1 \geq \tau_0 \coloneqq 1$ be the first time t after τ_0 so that the distance between $[L_t, R_t]$ and $Q(\tau_0)$ is at most two. The distance between $[L_{\tau_0}, R_{\tau_0}]$ and $Q(\tau_0)$ is at least five. The size of the arc $[L_t, R_t]$ increases by two at each time step. So, $\tau_1 \geq \tau_0 + 2$. Let q_1 be an element of $Q(\tau_0)$ that is of distance at most two from $[L_{\tau_1}, R_{\tau_1}]$; if there is more than one such q_1, choose arbitrarily. The minimality of τ_1 implies that q_1 is not in $[L_{\tau_1}, R_{\tau_1}]$.

Let $\tau_2 \geq \tau_1$ be the first time t after τ_1 so that the distance between $[L_t, R_t]$ and $Q(\tau_1)$ is at most two. Let q_2 be an element of $Q(\tau_1)$ that is of distance at most two from $[L_{\tau_2}, R_{\tau_2}]$. Define τ_j, q_j for $j > 2$ similarly, until $Q(\tau_j)$ is empty. As long as $|Q(\tau_j)| \geq 8$, we have $|Q(\tau_j + 1)| \geq |Q(\tau_j)| - 8$. This process, therefore, has at least $KN/16$ steps. For $1 \leq j \leq KN/16$, denote by E_j the event that during the time between τ_j and τ_{j+1} the pair $\{q_j, q_j + 1\}$ is added to P. The pair $\{q_j, q_j + 1\}$ is violating color $S(q_j)$. At time τ_j, even conditioned on all the past $P_1, \dots, P_{\tau_j}$, in at most two steps (and before τ_{j+1}) we can add the pair $\{q_j, q_j + 1\}$ to P. For every j, therefore,

$$\mathbb{P}[E_j | P_1, \dots, P_{\tau_j}] \geq (1/3)(1/3) = 1/9.$$

Next, let $N' = \lceil KN/960 \rceil$. We want to show that with high probability, for at least N' many j, the event E_j occurs. There are $\binom{\lfloor KN/16 \rfloor}{\lceil KN/960 \rceil}$ many ways of choosing a set of indices j of size $N - N'$. Subsequently,

$$\begin{aligned}\mathbb{P}[\text{there is } j_1, \dots, j_{N'} \text{ so that } E_{j_1} \cap \dots \cap E_{j_{N'}}] &\geq 1 - \binom{\lfloor KN/16 \rfloor}{\lceil KN/960 \rceil} \cdot \left(\frac{8}{9}\right)^{N-N'} \\ &\geq 1 - \left(\frac{960e}{16}\right)^{N'} \cdot \left(\frac{8}{9}\right)^{60N'} \\ &\geq 1 - c^{N'}\end{aligned}$$

where $0 < c < 1$ is a universal constant. Finally, we argue that if there do exist $j_1, \dots, j_{N'}$ for which the events $E_{j_1}, \dots, E_{j_{N'}}$ occur, then $G(P) \geq K/1000$. To see this, note that the size of every Q_k is N. So, every color k in B can contribute at most N elements to $j_1, \dots, j_{N'}$. If $G(P) < K/1000$, then at most these many colors can contribute larger than d^{ε} (and up to N elements) - combined, at most $KN/1000$ elements. However, there are at least $K/2 - K/1000$ colors which can contribute only up to d^{ε} elements. Again combined, this is not sufficient to cover the N' elements overall (for large enough d), which is a contradiction. Hence,

$$\mathbb{P}[G(P) \geq K/1000] \geq \mathbb{P}[\text{there is } j_1, \dots, j_{N'} \text{ so that } E_{j_1} \cap \dots \cap E_{j'_N}].$$

and the proof follows in this case as $c^{N'} \ll d^{-\Omega(K)}$.

Case 2: Many colors with few jumps

The intuition is that many violations will come from pairs of the form $\{L_t - 1, R_t + 1\}$ in the construction of the pairing. Assume that for at least $K/2$ colors k, $|J_k| \leq d^{2\varepsilon}$. Denote again by $B \subseteq [K]$ the set of k's that satisfy the above inequality. We say that a color k is noticeable in the arc A if

$$d^{\frac{\Delta}{\Delta+1} - 4\varepsilon} \leq |S_k \cap A| \leq |A| - d^{\frac{\Delta}{\Delta+1} - 4\varepsilon}.$$

▷ Claim 26. There are $K' \geq K/2 - 1$ disjoint arcs $A_1, \dots, A_{K'}$ so that for every $j \in [K']$,

1. $|A_j| = m = \lfloor d^{\frac{\Delta}{\Delta+1} - 3\varepsilon} \rfloor$ and,
2. there is a color k_j in B that is noticeable in A_j.

Moreover, the colors $k_1, \dots, k_{K'}$ can be chosen to be pairwise distinct.

Proof. For each color k in B, there are at least $d^{\frac{\Delta}{\Delta+1}}/2$ vertices of color k in the D-cycle and at most $d^{2\varepsilon}$ jumps in the color k. Therefore, there is at least one k-monochromatic arc of size at least $d^{\frac{\Delta}{\Delta+1} - 2\varepsilon}$. Hence, on the D-cycle, there are such monochromatic arcs $I_{k_1}, \dots, I_{k_{|B|}}$ for the colors $k_1, \dots, k_{|B|}$ in B, in this order ($1 < 2 < \dots < D$).

Consider an arc A of size m included in I_{k_1}. Thus $|S_{k_1} \cap A| = m$. If we "slide" the arc A until it is included in I_{k_2}, then $|S_{k_1} \cap A| = 0$. By continuity, there is an intermediate position for the arc A such that $d^{\frac{\Delta}{\Delta+1} - 4\varepsilon} \leq |S_{k_1} \cap A| \leq m - d^{\frac{\Delta}{\Delta+1} - 4\varepsilon}$. This provides the first arc A_1 of the claim.

Sliding an arc inside I_{k_2} to inside I_{k_3} shows that there exists an arc A_2 such that $d^{\frac{\Delta}{\Delta+1} - 4\varepsilon} \leq |S_{k_2} \cap A_2| \leq m - d^{\frac{\Delta}{\Delta+1} - 4\varepsilon}$. The arcs A_1 and A_2 are disjoint: The distance of the largest element of A_1 and the smallest element of I_{k_2} is at most m. The distance of the smallest element of A_2 and the largest element of I_{k_2} is at most m. The size of I_{k_2} is larger than $2m$. Proceed in this way to define $A_3, \dots, A_{|B|-1}$. ◁

Use Claim 26 to divide the construction of the (random) pairing into *epochs*. Denote by $A^{(0)}$ the family of arcs given by the claim. Let τ_1 be the first time t that the arc $[L_t, R_t]$ hits one of the arcs in $A^{(0)}$. Denote by A_1 that arc that is hit at time τ_1 (break ties arbitrarily). Denote by k_1 the color that is noticeable in A_1. Let σ_1 be the first time t so that A_1 is contained in $[L_t, R_t]$. Let $A^{(1)}$ be the subset of $A^{(0)}$ of arcs that have an empty intersection with $[L_{\sigma_1}, R_{\sigma_1}]$. Similarly, let τ_2 be the first time t after σ_1 that the arc $[L_t, R_t]$ hits one of the arcs in $A^{(1)}$. If there are no arc in $A^{(1)}$, define $\tau_2 = \infty$. Denote by A_2 that arc that is hit at time τ_2. Denote by k_2 the color that is noticeable in A_2. Let σ_2 be the first time t so that A_2 is contained in $[L_t, R_t]$. Let $A^{(2)}$ be the subset of $A^{(1)}$ of arcs that have an empty intersection with $[L\sigma_2, R\sigma_2]$. Define $\tau_j, \sigma_j, A_j, k_j, A^{(j)}$ for $j > 2$ analogously. For every $j \geq 1$, denote by E_j the event that during the time between τ_j and τ_{j+1} the number of pairs added that violate color k_j's at most d^{ε}. (If E_j does not hold, then $|V_{k_j}(P)| \geq d^{\varepsilon}$ and $k_j \in G(P)$. The main part of the proof is summarized in the following proposition, whose proof is deferred to Section 4.5.

▶ **Lemma 27** (Chessboard Lemma). *Let $\delta = 0.10$. For every $j \geq 1$, and any choice of pairs $P_1, \ldots, P_{\tau_j}$,*

$$\mathbb{P}[E_j | P_1, \ldots, P_{\tau_j}, |A^{(j-1)}| \geq 3] \leq d^{-\delta}$$

Given this lemma, let us finish the proof of Lemma 23. Define $K'' = \lfloor K'/10 \rfloor$ and let T denote the event that the number of j's for which $|A^{(j)}| \geq 3$ is at least K''. First, we argue that T occurs with high probability.

For any $j \geq 1$, consider the evolution of the arc $[L_t, R_t]$ between the time steps τ_j (when it first hits arc A_j) and σ_j (when it completely engulfs it). During this epoch, let us call the evolution of $[L_t, R_t]$ *in* the "direction" of A_j as *good* (labelled "G") and *away* from the direction of A_j as *bad* ("B"). To this end, for any time step in this epoch, we can *code* the three possible choices for the evolution of $[L_t, R_t]$ as GG (when the arc is grown *in* the direction of A_j), GB (when it is grown equally on either side), or BB (when it is grown *away* from the direction of A_j). Consequently, the evolution of $[L_t, R_t]$ during this epoch can be realized as a sequence consisting of the symbols G and B.

Consider the sequence s of G's and B's obtained by concatenating the sequences corresponding to all the epochs (ignoring the choices made at time steps that do *not* lie in such epochs, i.e., between τ_j and σ_j for some j - as there is no corresponding notion of a "good" direction outside such epochs). The intuition is that if $|A^{(K'')}| < 3$ (i.e., if T does not occur), then there must be an extremely large number of B's compared to G's (i.e., the arc $[L_t, R_t]$ evolves disproportionately in the *bad* direction) in the concatenated string s, which should occur only with a vanishingly small probability.

Consider the sub-string s' of s that corresponds to the choices made only for the nodes in $A^{(0)} \setminus A^{(K'')}$. Note that there are precisely mK'' many G's in s'. Suppose $|A^{(K'')}| = 2$ for concreteness (the cases $|A^{(K'')}| = 1$ and $|A^{(K'')}| = 0$ are similar). This implies that there are $m(K' - 2 - K'')$ many B's in s'. Since only up to mK'' many of these B's may appear as a result of the evolution making a choice of the form GB, it follows that the evolution of $[L_t, R_t]$ must make a choice of the form BB at least $m(K' - 2 - 2K'')/2$ times out of a possible $m(K' - 2)/2$, in order to cover the elements of $A^{(0)} \setminus A^{(K'')}$. Denote $K_1 := (K' - 2)/2$. By the union bound, this probability is at most

$$\mathbb{P}[|A^{(K'')}| = 2] \leq \binom{mK_1}{mK''} \cdot \left(\frac{1}{3}\right)^{m(K_1 - K'')} < c_2^{mK''}$$

for some universal constant $0 < c_2 < 1$. Similarly, we have bounds for both $\mathbb{P}[|A^{(K'')}| = 1]$ and $\mathbb{P}[|A^{(K'')}| = 0]$ and it follows that $\mathbb{P}[T] \geq 1 - c^{mK''}$ for some universal constant $0 < c < 1$.

▶ Remark 28. The argument above for showing that T occurs with high probability differs considerably from [8], where the corresponding event is sketched to occur with probability only at least $1 - dc^{m^{1/3}}$, which is not strong enough for our purposes.

Next, note that

$$\mathbb{P}[G(P) < K/1000] \le \mathbb{P}[G(P) < K/1000 \cap T] + \mathbb{P}[\neg T] \le \mathbb{P}[G(P) < K/1000|T] + \mathbb{P}[\neg T].$$

If $G(P) < K/1000$, then at least $K/2 - K/1000$ colors in B have at most d^{ε} many violations. Since $K'' = \lfloor K'/10 \rfloor < K/2 - K/1000$, in particular, there must exist at least $K''/2$ colors within the *first* K'' colors (here we are using the ordering of colors as provided by Claim 26) for which there are at most d^{ε} many violations. We then obtain the following by conditioning on T, using the union bound.

$$\mathbb{P}[G(P) < K/1000 \cap T] \le 2^{K''} \max_{H=\{j_1<\cdots<j_{K''/2}\}\subset[K'']} \mathbb{P}[E_{j_1}, \ldots, E_{j_{K''/2}} || A^{(K'')}| \ge 3]$$

For a fixed choice of H, by the chain rule and Lemma 27, we have

$$\begin{aligned}&\mathbb{P}[E_{j_1} \cap \cdots \cap E_{j_{K''/2}} || A^{(K'')}| \ge 3] \\ &\quad = \mathbb{P}[E_{j_1}|T] \cdot \mathbb{P}[E_{j_2}|E_{j_1} \cap T] \cdot \cdots \cdot \mathbb{P}[E_{j_{K''/2}}|E_{j_{K''/2-1}} \cap \cdots \cap E_{j_1} \cap T] \\ &\quad \le d^{-\delta K''/2} \le d^{-0.1K'/20} \le d^{-K/400}.\end{aligned}$$

Overall, we conclude that

$$\mathbb{P}[G(P) < K/1000] \le d^{-K/500}.$$

4.5 Proof of the Chessboard Lemma

To prove Lemma 27, we use a different point of view of the random process. We begin by describing this different view, and later describe its formal connection to the distribution on pairings. This subsection is adapted from Section 5 of [8] and closely follows their argument, though with numerous parameter changes to suit our demands.

The view uses two definitions. One is a standard definition of a two-dimensional random walk, and the other is a definition of a "chessboard" configuration in the plane. The proof of the proposition will follow by analyzing the behavior of the random walk on the "chessboard". Let d be as above and m be as defined in Lemma 26. The random walk W on $\mathbb{N}^2$ is defined as follows. It starts at the origin, $W_0 = (0, 0)$. At every step it move to one of three nodes, independently of previous choices,

$$W_{t+1} = \begin{cases} W_t + (0,2) & \text{with probability } 1/3 \\ W_t + (1,1) & \text{with probability } 1/3 \\ W_t + (2,0) & \text{with probability } 1/3 \end{cases}$$

At time t, the L_1-distance of Wt from the origin is thus $2t$.

The "chessboard" is defined as follows. Let $\alpha_1 : [m] \to \{0,1\}$ and $\alpha_2 : [2m] \to \{0,1\}$ be two Boolean functions. The functions α_1, α_2 induce a "chessboard" structure on the board $[m] \times [2m]$. A position in the board $\xi = (\xi_1, \xi_2)$ is colored either white or black. It is colored black if $\alpha_1(\xi_1) \ne \alpha_2(\xi_2)$ and white if $\alpha_1(\xi_1) = \alpha_2(\xi_2)$. We say that the "chessboard" is well-behaved if

1. α_1 is far from constant:
$$d^{\frac{\Delta}{\Delta+1}-4\varepsilon} \leq |\{\xi_1 \in [m] : \alpha_1(\xi_1) = 1\}| \leq m - d^{\frac{\Delta}{\Delta+1}-4\varepsilon}.$$
2. α_1 does not contain many jumps:
$$|\{\xi_1 \in [m-1] : \alpha_1(\xi_1) \neq \alpha_1(\xi_1 + 1)\}| \leq d^{2\varepsilon}$$
3. α_2 does not contain many jumps:
$$|\{\xi_2 \in [2m-1] : \alpha_2(\xi_2) \neq \alpha_2(\xi_2 + 1)\}| \leq d^{2\varepsilon}$$

Consider a random walk W on top of the "chessboard" and stop it when reaching the boundary of the board (i.e., when it tries to make a step outside the board $[m] \times [2m]$). We define a good step to be a step of the form $(1,1)$ that lands in a black block. We will later relate good steps to violating edges. Our goal is, therefore, to show that a typical W makes many good steps.

▶ **Lemma 29.** *Let $\delta = 0.10$ and assume the chessboard is well-behaved. The probability that W makes less than $d^{2\varepsilon}$ good steps is at most $d^{-\delta}$.*

We use this lemma to show Lemma 27.

Proof of Lemma 27 given Lemma 29. Recall that A_j is an arc of size $|A_j| = m = \lfloor d^{\frac{\Delta}{\Delta+1}-3\varepsilon} \rfloor$ so that there is a color k_j satisfying
$$d^{\frac{\Delta}{\Delta+1}-4\varepsilon} \leq |S_k \cap A| \leq |A| - d^{\frac{\Delta}{\Delta+1}-4\varepsilon}. \tag{2}$$

Furthermore, condition on $P_1, \ldots, P_{\tau_j}$, $|A^{(j-1)}| \geq 3$. Assume without loss of generality that R_{τ_j} is in A_j (when L_{τ_j} is in A_j, the analysis is similar). The distance of R_{τ_j} from the smallest element of A_j is at most one (the length of "one step to the right" is between zero and two). We now grow the random interval until σ_j, i.e., as long as R_t stays in A_j. At the same time, L_t performs a movement to the left. Since $|A^{(j-1)}| \geq 3$, there are at least $2m$ steps for L_t to take to the left before hitting A_j. There is a one-to-one correspondence between pairings P and random walks W using the correspondence
$$P_{t+1} = \{L_t - 2, L_t - 1\} \longleftrightarrow W_{t+1} = W_t + (0,2),$$
$$P_{t+1} = \{L_t - 1, R_t + 1\} \longleftrightarrow W_{t+1} = W_t + (1,1),$$
$$P_{t+1} = \{R_t + 1, R_t + 2\} \longleftrightarrow W_{t+1} = W_t + (2,0).$$

Define the function α_1 to be 1 at positions of A_j with color k_j, and 0 at the other positions. Set the function α_2 as to describe the color k_j from L_{τ_j} leftward. The "chessboard" is well-behaved by (2) and since k_j is in the set B defined in case 2 of the proof of Lemma 23 (so there are not many jumps for the color k_j). Finally, if W makes a good step, then the corresponding pair added to P violated color k_j . So, if E_j holds for P, then the corresponding W makes less than $d^{2\varepsilon}$ good steps. Formally, by Lemma 23,
$$\mathbb{P}[E_j | P_1, \ldots, P_{\tau_j}, |A^{(j-1)}| \geq 3] \leq \mathbb{P}[W \text{ makes less than } d^{2\varepsilon} \text{ good steps}] \leq d^{-\delta}. \quad ◀$$

Proof of Lemma 29. Define three events E_R, E_C, E_D, all of which happen with small probability, so that every W that is not in their union makes many good steps.

Call a subset of the board of the form $I \times [2m]$ or $[m] \times I$, where I is a sub-interval, a *region*. The width of a region is the size of I. Let R be the set of regions of width at least $d^{4\varepsilon}$. The size of R is at most $2m^2$. For a region r in R, denote by E_r the event that the number of steps of the form $(1, 1)$ that W makes in r is less than $d^{2\varepsilon}$ *given* that it makes at least $d^{3\varepsilon}$ steps in r. Denote

$$E_R = \bigcup_{r \in R} E_r$$

To estimate the probability of E_r, note that we can simply apply the Chernoff bound to a sum of $d^{3\varepsilon}$ Bernoulli random variables with $p = 1/3$. By the union bound, we conclude that there is a universal constant $0 < c < 1$ such that

$$\mathbb{P}[E_R] \leq c^{d^{3\varepsilon}}.$$

Denote by H the set of all points in the board with L_1-norm at least $m^{5/8}$. At time T the random walk W is distributed along all points in $\mathbb{N}^2$ of L_1-norm exactly T. The distribution of W on this set is the same as that of a random walk on $\mathbb{Z}$ that is started at 0, and moves at every step to the right with probability 1/3, stays in place with probability 1/3 and moves to the left with probability 1/3. The probability that such a random walk on $\mathbb{Z}$ is at a specific point in $\mathbb{Z}$ at time T is at most $O(T^{-1/2})$. Hence, for every point h in H,

$$\mathbb{P}[W \text{ hits } h] \leq O(m^{-5/16}) \leq m^{-1/4}.$$

Call a point $c = (\xi_1, \xi_2)$ in the board a corner if both (ξ_1, ξ_2) and $(\xi_1 + 1, \xi_2 + 1)$ are of the same color $\kappa \in \{\text{black}, \text{white}\}$, but $(\xi_1 + 1, \xi_2)$ and $(\xi_1, \xi_2 + 1)$ are not of color κ. For a corner c, denote by $\Delta(c)$ the $d^{4\varepsilon}$-neighborhood of c in L_1-metric. Denote by Δ the union over all $\Delta(c)$, for corners c in H. Denote by E_C the event that W hits any point in Δ. Since the board is well-behaved, the number of jumps in each of α_1, α_2 is at most $d^{2\varepsilon}$. Therefore, the number of corners is at most $d^{4\varepsilon}$. By the union bound,

$$\mathbb{P}[E_C] \leq O(d^{4\varepsilon} d^{8\varepsilon} m^{-1/4}) \leq d^{-0.112},$$

where in the last step, we plugged in $\varepsilon = 1/1000$ and used $m \geq d^{1/2-3\varepsilon}$. Next, let $m' = \lceil m^{5/8} \rceil$. Define three (vertical) lines: D_1 is the line $\{m'\} \times [2m]$, D_2 is the line $\{2m'\} \times [2m]$ and D_3 is the line $\{m - m'\} \times [2m]$. Denote by E_D the event that W does not cross the line D_3 before stopped (i.e., hitting the boundary of the board). Chernoff's bound implies that there is a universal constant $0 < c < 1$ for which

$$P[E_D] \leq c^m.$$

To conclude the proof by the union bound, it suffices to show that for every W not in $E_R \cup E_C \cup E_D$, the walk W makes at least $d^{2\varepsilon}$ good steps. Fix such a walk W. Since $W \notin E_D$, we know that W crosses the line D_2.

We consider several cases. Define a *block* to be a maximal monochromatic rectangle in the board. The board is thus partitioned into black blocks and white blocks - which is what led [8] to calling it a "chessboard." We now think of the board $[m] \times [2m]$ as drawn in the plane with $(1, 1)$ at the bottom-left corner and $(m, 2m)$ at the upper-right corner.

Case 1: The walk W does not hit any white block after crossing D_1 and before crossing D_2. In this case, all steps taken in the *region* whose left border is D_1 and right border is D_2 are in a black area. The number of such steps is at least $m^{5/8}/2 \gg d^{3\varepsilon}$. Since $W \notin E_R$, the claim holds.

Case 2: The walk W hits a white block after crossing D_1 and before crossing D_2. Let us label the blocks as follows: we associate every block with a pair $\langle \eta_1, \eta_2 \rangle$ where η_1 is between 1 and the number of jumps in α_1 and η_2 is between 1 and the number of jumps in α_2. So, the label of the "bottom-left" is $\langle 1, 1 \rangle$, the label of the block "above" it is $\langle 1, 2 \rangle$ and the label of the block "to its right" is $\langle 2, 1 \rangle$, etc. There are two sub-cases to consider:

Sub-case 1: At some point after crossing D_1 and before crossing D_3, there are two white blocks of the form $\langle \eta_1, \eta_2 \rangle$, $\langle \eta_1 + 1, \eta_2 + 1 \rangle$ so that W intersects both blocks. Let c be the corner between these two blocks (which must exist by definition). Since $W \notin E_C$, we know that W does not visit $\Delta(c)$. Therefore, W must walk in a black area around $\Delta(c)$. Every path surrounding $\Delta(c)$ has length at least $d^{4\varepsilon}$. Since $W \notin E_R$, the claim holds.

Sub-case 2: At all times after crossing D_1 and before crossing D_3, the walk never moves from a white block $\langle \eta_1, \eta_2 \rangle$ to one of the two white blocks $\langle \eta_1 + 1, \eta_2 + 1 \rangle$, $\langle \eta_1 - 1, \eta_2 - 1 \rangle$. Since $W \notin E_D$, this is indeed the last case. The width of a combinatorial rectangle in the board is the size of its "bottom side" (i.e., the corresponding subset of [m]). Let η be the first white block W hits after crossing D_1. Let Σ be the family of black blocks that are to the right but on the same height as η. Define γ as the maximal width of a rectangle of the form $\sigma \cap [0, m - m_0 - 1] \times [2m]$ over all $\sigma \in \Sigma$. Since the board is well-behaved, it follows (from the first condition) that the total width of the black area on the same height as η is at least $d^{\frac{\Delta}{\Delta+1} - 4\varepsilon}$. Also, since we are in case 2, the left border of η is to the left of D_2. Therefore, the total width of the black area to the right of the left border of η and to the left of D_3, on the same height as η is at least $d^{\frac{\Delta}{\Delta+1} - 4\varepsilon} - 3m'$. Therefore, since the number of jumps is at most $d^{2\varepsilon}$,

$$\gamma \geq (d^{\frac{\Delta}{\Delta+1} - 4\varepsilon} - 3m')/d^{2\varepsilon} \gg d^{4\varepsilon}.$$

Since we are in this sub-case, the walk W must "go through" every black block it hits: it can go from bottom side to upper side or from left side to right side (but not from left side to upper side or from bottom side to right side). Consider the behaviour of W after it hits η: starting from a white block, because $W \notin E_D$, it is guaranteed to cross D_3. Therefore, the color of the block that W "exits" from from each *column* must keep alternating between white and black. For each black block in Σ, therefore, there exists a black block in the same column that W crosses horizontally. Focusing on one such black block of width γ, since $W \notin E_R$, the claim holds. ◀

5 Discussion and Open Problems

We conclude by mentioning some interesting directions for future work.

- The most interesting and natural question is to make the hard polynomial in our main result $\text{IMM}_{n,n}$. This would imply super-polynomial algebraic formula lower bounds. As far as we know, it is conceivable that even the complexity measure of [22] as described in Section 3 could be used to prove the lower bound for the $\text{IMM}_{n,n}$ polynomial. While the relative rank of $\text{IMM}_{n,n}$ itself is low, there might be a suitable "restriction" of it such that for a randomly chosen $w \in \{-k, k\}^n$, with reasonably high probability the restriction has large rank. This could then be used to prove the lower bound for $\text{IMM}_{n,n}$ (using Lemma 11 and Lemma 12). Secondly, we point out that perhaps it is more viable to find an *ordered* set-multilinear branching program (as described in Section 1.4) which can be shown to be *arc-full-rank*. This would also lead to general formula lower bounds.

- The discussion in Section 1.4 raises the question of the relative computational power of the ordered vs general set-multilinear branching program models. Clearly, if it is shown that these classes coincide, then it leads to formula lower bounds via Theorem 1. We would like to note here that in fact, exponential lower bounds are known for the *ordered* model (see [4] for a discussion[19]).

References

1 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM J. Comput.*, 44(3):669–697, 2015. doi:10.1137/140975103.

2 Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 67–75. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.32.

3 Miklós Ajtai. $\sum^1{}_1$-formulae on finite structures. *Ann. Pure Appl. Log.*, 24(1):1–48, 1983. doi:10.1016/0168-0072(83)90038-6.

4 Vikraman Arvind and S. Raja. Some lower bound results for set-multilinear arithmetic computations. *Chic. J. Theor. Comput. Sci.*, 2016, 2016. URL: http://cjtcs.cs.uchicago.edu/articles/2016/6/contents.html.

5 Peter Bürgisser. Cook's versus valiant's hypothesis. *Theor. Comput. Sci.*, 235(1):71–88, 2000. doi:10.1016/S0304-3975(99)00183-8.

6 Suryajith Chillara, Christian Engels, Nutan Limaye, and Srikanth Srinivasan. A near-optimal depth-hierarchy theorem for small-depth multilinear circuits. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 934–945. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00092.

7 Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. Small-depth multilinear formula lower bounds for iterated matrix multiplication with applications. *SIAM J. Comput.*, 48(1):70–92, 2019. doi:10.1137/18M1191567.

8 Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19–22, 2012*, pages 615–624. ACM, 2012. doi:10.1145/2213977.2214034.

9 Michael A. Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 243–252. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.34.

10 Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth-4 formulas computing iterated matrix multiplication. *SIAM J. Comput.*, 44(5):1173–1201, 2015. doi:10.1137/140990280.

11 Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Syst. Theory*, 17(1):13–27, 1984. doi:10.1007/BF01744431.

12 Zeyu Guo and Rohit Gurjar. Improved explicit hitting-sets for roabps. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.4.

[19] What they term as a "type-width 1" set-multilinear ABP is an *ordered* set-multilinear ABP for us.

13 Johan Håstad. Almost optimal lower bounds for small depth circuits. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 6–20. ACM, 1986. doi:10.1145/12130.12132.

14 Pavel Hrubes and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Comput. Complex.*, 20(3):559–578, 2011. doi:10.1007/s00037-011-0007-3.

15 Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. *SIAM J. Comput.*, 46(1):307–335, 2017. doi:10.1137/151002423.

16 Neeraj Kayal, Vineet Nair, and Chandan Saha. Separation between read-once oblivious algebraic branching programs (roabps) and multilinear depth-three circuits. *ACM Trans. Comput. Theory*, 12(1):2:1–2:27, 2020. doi:10.1145/3369928.

17 Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 – June 03, 2014*, pages 146–153. ACM, 2014. doi:10.1145/2591796.2591847.

18 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An almost cubic lower bound for depth three arithmetic circuits. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.33.

19 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. On the size of homogeneous and of depth-four formulas with low individual degree. *Theory Comput.*, 14(1):1–46, 2018. doi:10.4086/toc.2018.v014a016.

20 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. doi:10.1016/j.tcs.2012.03.041.

21 Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. *SIAM J. Comput.*, 46(1):336–387, 2017. doi:10.1137/140999335.

22 Deepanshu Kush and Shubhangi Saraf. Improved low-depth set-multilinear circuit lower bounds. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 38:1–38:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.38.

23 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 804–814. IEEE, 2021. doi:10.1109/FOCS52979.2021.00083.

24 Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.

25 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complex.*, 6(3):217–234, 1997. doi:10.1007/BF01294256.

26 Ran Raz. Separation of multilinear circuit and formula size. *Theory Comput.*, 2(6):121–135, 2006. doi:10.4086/toc.2006.v002a006.

27 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):8:1–8:17, 2009. doi:10.1145/1502793.1502797.

28 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40:1–40:15, 2013. doi:10.1145/2535928.

29 Ran Raz and Amir Yehudayoff. Balancing syntactically multilinear arithmetic circuits. *Comput. Complex.*, 17(4):515–535, 2008. doi:10.1007/s00037-008-0254-0.

30 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Comput. Complex.*, 18(2):171–207, 2009. doi:10.1007/s00037-009-0270-8.

31 Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41:333–338, 1987.

32 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. *Github Survey*, 2015. URL: `https://github.com/dasarpmar/lowerbounds-survey`.

33 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3-4):207–388, 2010. `doi:10.1561/0400000039`.

34 Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82. ACM, 1987. `doi:10.1145/28395.28404`.

35 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. `doi:10.1016/j.ic.2014.09.004`.

36 Sébastien Tavenas, Nutan Limaye, and Srikanth Srinivasan. Set-multilinear and non-commutative formula lower bounds for iterated matrix multiplication. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20–24, 2022*, pages 416–425. ACM, 2022. `doi:10.1145/3519935.3520044`.

37 Leslie G. Valiant, Sven Skyum, Stuart J. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12:641–644, 1983.

38 Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 1–10. IEEE Computer Society, 1985. `doi:10.1109/SFCS.1985.49`.

A Proof Sketch of Lemma 22

In this section, we describe the proof of Lemma 22. As mentioned in Section 4, the proof structure is very similar to that of Lemma 20. The setup is similar as well, but we describe it here again for the convenience of the reader.

Again, we identify the set of variables $X = (X_1, \dots, X_d)$ with the d-cycle $\{1, 2, \dots, d\}$, where addition is modulo d. Let S be a partition of the cycle to K parts, namely, $S = (S_1, \dots, S_K)$. We also think of $[K]$ as a set of colors, and of S as a (now "full") coloring of the cycle.

For a pairing P, define the number of k-violations by

$$V_k(P) = \{P_t \in P : |P_t \cap S_k| = 1\}$$

in words, the set of pairs in which one color is k and the other color is different. Fix $\varepsilon = 1/1000$ and denote

$$G(P) = \{k \in [K] : |V_k(P)| \geq d^{\varepsilon}\}$$

We do not include S as a subscript in these two notations since S will be known from the context (and will be fixed throughout most of the discussion). We begin by stating the analogue to Lemma 23, which shows that for every fixed K-coloring of the cycle, a random pairing has, with high probability, many colors with many violations.

▶ **Lemma 30.** *There is a constant $C > 0$ such that for all integers K in the range $[C, d^{1/1000}]$ the following holds: Let $S = (S_1, \dots, S_K)$ be a partition of the d-cycle and suppose that $|S_k| \geq d^{7/8}$ for all $k \in [K]$. Then,*

$$\mathbb{P}[G(P) \leq K/1000] \leq d^{-K/500},$$

where $P \sim \mathcal{D}_P$.

Let us prove Lemma 22 given this lemma.

Proof of Lemma 22 given Lemma 30. We first need the following structural result, whose proof can be extrapolated from [33] , where it is shown for multilinear formulas.

▶ **Lemma 31** (Product Lemma; see [8, 33]). *Assume that F is a formula with at most s leaves, and is set-multilinear with respect to the set partition $(X_1, \ldots, X_d)$. Then, we can write*

$$F = \sum_{i=1}^{s} \prod_{j=1}^{\ell} F_{i,j}$$

where $\ell \geq \log d/100$ and for each $i \in [s]$, the product $F_i = \prod_{j=1}^{\ell} F_{i,j}$ is also set-multilinear. Furthermore, the degree of each $F_{i,j}$ is at least $d^{7/8}$.

Continuing with the proof, let F be a formula as in the statement of Lemma 22. We start by writing $F = \sum_{i=1}^{s} F_i$ in the form given by Lemma 31, so that each $F_i = \prod_{j=1}^{\ell} F_{i,j}$. As each F_i is set-multilinear, $(S_{i,1}, \ldots, S_{i,\ell})$ form a partition of $[d]$ where each $F_{i,j}$ is set-multilinear with respect to $(X_p)_{p \in S_{i,j}}$. Let $w^{i,1}, \ldots, w^{i,\ell}$ be the corresponding decomposition, whose respective sums are denoted simply by $w_{S_{i,1}}, \ldots, w_{S_{i,\ell}}$.

From the properties of relrk_w (Claim 7), we have

$$\mathrm{relrk}_w(F_i) = \prod_{j=1}^{\ell} \mathrm{relrk}_{w^{i,j}}(F_{i,j}) \leq \prod_{j=1}^{\ell} 2^{-\frac{1}{2}|w_{S_j}|} = 2^{-\frac{1}{2}\sum_{j=1}^{\ell}|w_{S_j}|},$$

from which we observe that the task of upper bounding $\mathrm{relrk}_w(F)$ can be reduced to the task of lower bounding the sum $\sum_{j=1}^{\ell} |w_{S_j}|$, which is established in the following claim. For the sake of convenience, the choice of the alphabet for w below is scaled down to $\{-1, 1\}$.

▷ Claim 32. For large enough d, suppose $(S_1, \ldots, S_K)$ is a partition of $[d]$ such that each $|S_j| \geq d^{7/8}$. Then, we have

$$\mathop{\mathbb{P}}_{w \sim \mathcal{D}} \left[\sum_{j=1}^{K} |w_{S_j}| < \frac{\log d}{2000} \right] \leq d^{-\frac{\log d}{10^7}}.$$

Here, $\mathcal{D}$ refers to the original distribution i.e., an arc-partition over the d-cycle.

Proof. The proof is going to be similar to that of Claim 25. Applying Lemma 30 to the tuple $(S_1, \ldots, S_K)$, we obtain that

$$\mathbb{P}[G(P) \leq K/1000] \leq d^{-K/500}.$$

The idea is to condition on the high probability event that $G(P) > K/1000$. Fix a pairing P with this property. Consider an ordering σ of the colors in $G(P)$. A color k is said to be *bright* with respect to an ordering if there are at least $d^{\varepsilon}/2$ nodes x of color k such that either the partner of x is uncolored or its partner is colored using a color that appears *after* k in the ordering σ. Call an ordering σ of the nodes in $G(P)$ *good* if there are at least $|G(P)|/2$ bright colors with respect to σ. The observation is that for any ordering σ of the colors, either σ itself is good, or its reverse is good. We conclude that given any pairing P, there exists a good ordering of $G(P)$. Fix any such good ordering and let $H(P)$ be the collection of bright colors with respect to this ordering.

Next, notice that if the sum $\sum_{j=1}^{K} |w_{S_j}|$ is at most $\frac{\log d}{2000}$, then so is the sum $\sum_{k \in H(P)} |w_{S_k}|$. Let $K' = |H(P)|$ (which is at least $K/2000$ if $G(P) > K/1000$). View the sampling of Π from P as happening in a specific order, according to the order of $k_1, k_2, \ldots, k_{K'}$: First define Π on pairs with at least one point with color k_1, then define Π on remaining pairs with at least one point with color k_2, and so forth. When finished with $k_1, \ldots, k_{K'}$, continue to define Π on all other pairs.

Conditioned on the event that $G(P) > K/1000$, this implies that $|w_{S_j}| \leq 1$ for each $j \in H(P)$. For every $j \in H(P)$, define E_j to be the event that $|w_{S_{k_j}}| \leq 1$. By choice, conditioned on $E_1, \ldots, E_{j-1}$, there are at least $d^{\varepsilon}/2$ pairs P_t so that $|P_t \cap S_{k_j}| = 1$ that are not yet assigned a "positive" or "negative" sign. For every such P_t, the element in $P_t \cap S_{k_j}$ is assigned a positive sign with probability $1/2$, and is independent of any other $P_{t'}$. The probability that a binomial random variable B over a universe of size $U \geq d^{\varepsilon}/2$ and marginals $1/2$ obtains any specific value is at most $O(U^{-1/2}) = O(d^{-\varepsilon/2})$. Hence, for all $j \in H(P)$, by the union bound,

$$\mathbb{P}[E_j | E_1, \ldots, E_{j-1}, P] \leq \mathbb{P}_B[U/2 - 1 \leq B \leq U/2 + 1] \leq O(3 \cdot d^{-\varepsilon/2}) \leq d^{-\varepsilon/4}.$$

Therefore,

$$\mathbb{P}[|w_{S_{k_j}}| \leq 1 \text{ for all } j \in H(P)] \leq \mathbb{E}[d^{-\varepsilon|H(P)|/4} | G(P) > K/1000] + d^{-K/500} \leq d^{-K/10^7}.$$

Finally, we note that

$$\mathbb{P}_{w \sim \mathcal{D}}\left[\sum_{j=1}^{K} |w_{S_j}| < \frac{\log d}{2000}\right] \leq \mathbb{P}[|w_{S_{k_j}}| \leq 1 \text{ for all } j \in H(P)]. \qquad \lhd$$

The claim above and the preceding calculation immediately implies that for every sub-formula F_i of size s_i,

$$\mathrm{relrk}_w(F_i) \leq s_i \cdot 2^{-\frac{k \log d}{2000}}$$

with probability at least $1 - d^{-\frac{\log d}{10^7}} \geq 1 - s_i \cdot d^{-\frac{\log d}{10^7}}$.

Next, by a union bound over $i \in [s]$ and the sub-additivity property of relrk_w, it follows that

$$\mathrm{relrk}_w(F) \leq s \cdot 2^{-\frac{k \log d}{2000}}$$

with probability at least $1 - s \cdot d^{-\frac{\log d}{10^7}}$, which concludes the proof of the lemma. ◀

We shall omit the proof of Lemma 30 here as it is, in fact, a significantly easier adaptation of Lemma 4.1 from [8] than the proof of Lemma 23 – this is because we no longer need to conduct the tighter analysis that was necessary for the low-depth case.