Electronic Colloquium on Computational Complexity, Report No. 14 (2023)

Depth-3 Circuit Lower Bounds for k-OV

Tameem Choudhury
Department of Computer Science and Engineering, IIT Hyderabad

Karteek Sreenivasaiah
Department of Computer Science and Engineering, IIT Hyderabad

Abstract

The 2-Orthogonal Vectors (2-OV) problem is the following: given two tuples A and B of n vectors each of dimension d, decide if there exists a vector $u \in A$, and $v \in B$ such that u and v are orthogonal. This problem, and its generalization k-OV defined analogously for k tuples, are central problems in the area of fine-grained complexity. Informally speaking, one of the major conjectures in fine-grained complexity is that deciding k-OV requires time $\Omega(n^k d)$.

In this paper, we are interested in unconditional lower bounds against k-OV, but for weaker models of computation than the general Turing Machine. In particular, we are interested in circuit lower bounds to computing k-OV by Boolean circuit families of depth 3 of the form OR-AND-OR, or equivalently, a *disjunction of CNFs*.

We show that for all $k \leq d$, any disjunction of t-CNFs computing k-OV requires size $\Omega((n/t)^k)$. In particular, when k is a constant, any disjunction of k-CNFs computing k-OV needs to use $\Omega(n^k)$ CNFs. This matches the brute-force construction. Thus for each fixed $k \geq 2$, the complexity of computing k-OV as a disjunction of k-CNFs is $\Theta(n^k)$. Our results partially resolve a conjecture by Kane and Williams [16] (page 12, conjecture 10) about depth-3 AC^0 circuits computing 2-OV.

As a secondary result, we show an exponential lower bound on the size of AND-OR-AND circuits computing 2-OV when d is very large. Since 2-OV reduces to k-OV by projections trivially, this lower bound works against k-OV as well.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Problems, reductions and completeness

Keywords and phrases fine grained complexity, k-OV, circuit lower bounds, depth-3

1 Introduction

The area of fine-grained complexity is a branch of computational complexity that studies the complexity of functions with a finer lens than the usual approach that makes a coarse distinction between polynomial time and super-polynomial time. The area has been focused on functions in P that find uses in a variety of contexts. In the seminal paper by Vassilevska Williams and Williams [24], they show eight problems that are subcubic time equivalent to one another. Hence a truly subcubic time algorithm for any one of these problems will also imply a subcubic algorithm for the others.

The holy grail of computation complexity is to show *unconditional* lower bounds to resources used in computing an *explicit* function. Unfortunately, the state of affairs in terms of unconditional lower bounds for computation, in its full generality, is rather bleak. The best known unconditional lower bounds for the running time of computing an explicit function are merely linear. Even for functions such as SAT that do not have any polynomial time running algorithms till date, we do not know how to show super-linear lower bounds. We do know from the time hierarchy theorem[1] that there are languages in $\mathsf{DTIME}(n^2)$ that are not in $\mathsf{DTIME}(n^c)$ for any $c < 2$. However the languages constructed in a proof of the time hierarchy are not natural, and not as explicit as we would like. Results such as [24] and

[1] Such hierarchy theorems go through for the unit cost RAM model as well.

ISSN 1433-8092

[7] that show equivalences among several important functions help in identifying candidate functions that might witness the time hierarchy theorem for their time class. One such candidate function for quadratic time[2] is the *2-Orthogonal Vectors problem*.

The 2-Orthogonal Vectors problem $\mathsf{2\text{-}OV}_{n,d}$ is defined as follows: Given as input two tuples $A \subseteq \{0,1\}^d$ and $B \subseteq \{0,1\}^d$ of n vectors each, decide if there is a vector $a \in A$ and a vector $b \in B$ such that a and b are orthogonal. A natural generalization of this problem is the function $\mathsf{k\text{-}OV}_{n,d}$ where the input has k such tuples and the task is to decide if there is a choice of k vectors, one from each tuple, that are disjoint. The problems 2-OV and k-OV have emerged as central problems in fine-grained complexity. An important hypothesis is that no deterministic, or randomized, algorithm computing $\mathsf{2\text{-}OV}_{n,d}$ can run in time $O(n^{2-\epsilon}\,\mathrm{poly}(d))$ for any $\epsilon > 0$. This is essentially saying that the brute force algorithm is also the best. Interestingly, Ryan Williams in [22], shows that under the *strong exponential time hypothesis* (SETH)[3], 2-OV (3-OV) requires $n^{2-o(1)}$ time ($n^{3-o(1)}$ time respectively).

In the absence of techniques that can show unconditional lower bounds, two natural directions of research emerge: (i) Conditional lower bounds to help us understand connections between various such problems, and "bottlenecks" to better algorithms. (ii) Unconditional lower bounds for weaker models of computation.

The first line of research has seen a tremendous body of results. There are numerous fine-grained reductions, and lower bounds, conditioned on SETH, and the hardness of functions such as $\mathsf{2\text{-}OV}_{n,d}$, and $\mathsf{k\text{-}OV}_{n,d}$. In the 2018 survey [23], Vassilevska Williams aptly describes it as *"an explosion of hardness results based on OV"*, and lists nineteen problems whose complexity is connected to that of k-OV. The fact that better algorithms for so many problems would imply better algorithms for k-OV, is perhaps not surprising. Intuitively, the k-OV function looks "canonical" in a certain sense, and so has managed to hide itself inside several other problems that look quite different at the surface. These include seemingly unrelated problems such as Longest Common Subsequence [1], Edit Distance [2], Fréchet distance [4, 5], Regular Expressions Matching [3], to name a few. Their survey [23] is an excellent source for those looking for a thorough treatment of fine-grained complexity, and in particular, this line of research.

The second direction, of showing lower bounds against weaker models of computation, seems to be lacking the same attention. To the best of our knowledge, the only paper that addresses this line is that of Kane and Williams [16]. In their paper they show tight lower bounds for formulas and branching programs computing 2-OV. We do not know any non-trivial lower bounds for computing 2-OV by models stronger than branching programs.

Note that if a uniform circuit family of bounded fan-in, and size $O(s(n,d))$ computes $\mathsf{k\text{-}OV}_{n,d}$, then an algorithm that simply evaluates the circuit, computes $\mathsf{k\text{-}OV}_{n,d}$ in time $O(s(n,d))$. So if the k-OV hypothesis is true, then any uniform circuit family computing $\mathsf{k\text{-}OV}_{n,d}$ must have size $\Omega(n^k)$. This begs the question:

What is the largest class of circuits for which we can show $\Omega(n^k\,poly(d))$ size lower bounds against computing $\mathsf{k\text{-}OV}_{n,d}$?

The largest class of circuits for which we have any non-trivial lower bounds at all (against computing an explicit function) is $\mathsf{ACC}^0[p]$ (gates from $\{\wedge, \vee, \neg, \mathrm{mod}_p\}$ for any prime p, unbounded fan-in, $O(1)$-depth, where a mod_p gate outputs 1 if and only if the number of

[2] We are being imprecise here so as to remain informal. The input length of $\mathsf{2\text{-}OV}_{n,d}$ is actually nd. So "quadratic in n" is not the same as $\mathsf{DTIME}(n^2)$

[3] [14],[6]For every $\epsilon > 0$, $\exists k$ such that k-SAT problem on n variables cannot be solved in $O(2^{(1-\epsilon)n})$ time

ones in the input is not $0 \mod p$). In fact we know exponential lower bounds against these classes of circuits. Hence a good target to aim for is to show that $\mathsf{k\text{-}OV}_{n,d}$ requires $\mathsf{ACC}^0[p]$ circuits, or just AC^0 circuits (gates from $\{\wedge, \vee, \neg\}$, unbounded fan-in, $O(1)$-depth) of size $\Omega(n^k \operatorname{poly}(d))$. A good starting point is to study depth-3 AC^0 circuits.

The best known lower bound against depth-3 AC^0 circuits is $2^{\Omega(\sqrt{n})}$ for computing majority. This bound is tight, and can be obtained by several classic techniques from the 80s including the switching lemma by Håstad [12], the polynomial method by Razborov [19] and Smolensky [20], and finite-limit vectors by [13]. One of the most important problems in circuit complexity is to prove $2^{\omega(n/\log\log n)}$ lower bounds to the size of depth-3 AC^0 circuits computing an explicit function. This would imply superlinear lower bounds against $O(\log n)$ depth circuits (of bounded fan-in) due to the depth reduction procedure described by Valiant [21] (alternatively, see Chapter 11 of Jukna [15]). With the aim of making progress on this front, Goldreich and Wigderson proposed a new framework in [10] where they define a new model of arithmetic circuits that use *multilinear gates*, as opposed to allowing gates computing sum or product alone, and a new complexity measure on this model. The main motivation being that lower bounds to their complexity measure implies lower bounds to a specific class of Boolean depth-3 circuits that they call *D-canonical*. The best lower bounds obtained for this class of depth-3 Boolean circuits, using their framework, is $\Omega(2^{n^{3/5}})$ by Goldreich and Tal [9]. In fact, the brute force depth-3 AC^0 circuits computing the negation of k-OV, described later in equation 3, bears close resemblance to D-canonical circuits since it is a product of set-multilinear functions, but over the Boolean algebra, as opposed to $\mathsf{GF}(2)$.

More recently, the status of depth-3 $\mathsf{AC}^0[\oplus]$ circuits got an update. The lower bound for computing majority using depth-3 $\mathsf{AC}^0[\oplus]$ circuits was improved from $2^{\Omega(n^{1/4})}$ to $2^{\Omega(\sqrt{n})}$ by Oliveira, Santhanam and Srinivasan [18]. This closed the gap between upper and lower bounds upto a logaritmic factor in the exponent.

While techniques such as the switching lemma and the polynomial method work in a "bottom-up" fashion, the techniques in [13] is a "top-down" approach specifically for depth-3 AC^0 circuits. To the best of our knowledge, the only top-down strategies for circuit lower bounds are the *Karchmer-Wigderson game* by Karchmer and Wigderson [17], the *discriminator lemma* for depth-2 threshold circuits by Hajnal, Masse, Pudlák, Szegedy, Turán [11], and *finite-limits* by Håstad, Jukna, Pudlak [13]. Our results in this paper can be seen as a non-trivial application of the techniques of Håstad, Jukna, Pudlak [13].

Kane and Williams [16] conjecture that any depth-3 AC^0 circuit computing $\mathsf{2\text{-}OV}_{n,d}$ requires $\Omega(n^2)$ wires (see page 12, conjecture 10 in [16]). Observe that $\mathsf{2\text{-}OV}_{n,d}$ (and $\mathsf{k\text{-}OV}_{n,d}$) can be computed by $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuits with $2n^2d$ wires (and kn^kd wires respectively):

$$\mathsf{2\text{-}OV}_{n,d}(A, B) = \bigvee_{i_1, i_2 \in [n]} \bigwedge_{j \in [d]} (\neg a_{i_1}[j] \vee \neg b_{i_2}[j]) \tag{1}$$

$$\mathsf{k\text{-}OV}_{n,d}(A_1, \ldots, A_k) = \bigvee_{i_1, \ldots, i_k \in [n]} \bigwedge_{j \in [d]} (\neg a_{i_1}[j] \vee \cdots \vee \neg a_{i_k}[j]) \tag{2}$$

Hence, informally, their conjecture for $\mathsf{2\text{-}OV}_{n,d}$, and by extension $\mathsf{k\text{-}OV}_{n,d}$, is that the brute-force circuit is also the best.

A second important question in [16] is about generalizing lower bounds from 2-OV to k-OV. As they have noted, generalizing their lower bounds to $k > 2$ would beat the state of the art in branching program lower bounds. Our results for depth-3 AC^0 circuits generalize to $k > 2$, and scale well when the bottom fan-in is bounded.

Our Results

In this paper, we show lower bounds against the size of depth-3 AC^0 circuit families computing $\mathsf{k\text{-}OV}_{n,d}$ with the gates on the bottom layer restricted to having small fan-in. Our main result is the following:

▶ **Theorem 1.** *For all $k \le d$, any* $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ *circuit with bottom fan-in t computing* $\mathsf{k\text{-}OV}_{n,d}$ *requires top fan-in $\Omega\left((\frac{n}{t})^k\right)$.*

Circuit families of the type $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ can also be understood as a *disjunction of CNFs*. Therefore Theorem 1 is equivalent to the following statement:

"Any disjunction of t-CNFs computing $\mathsf{k\text{-}OV}_{n,d}$ *requires size $\Omega(n/t)^k$."*

(Here, by '*t-CNF*', we mean a CNF whose clauses have at most t literals, and by '*size*' we mean the number of CNFs being used.)

The brute-force circuit described earlier in equation 2 for $\mathsf{k\text{-}OV}_{n,d}$, is a disjunction of n^k many k-CNFs, and the lower bound from Theorem 1 for this model is $\Omega((n/k)^k)$. Hence for all constant $k > 1$, the complexity of computing $\mathsf{k\text{-}OV}_{n,d}$ as a disjunction of k-CNFs is $\Theta(n^k)$.

The proof technique used for Theorem 1 is that of [13] which goes through as long as the bottom layer gates are a function of at most t inputs. We describe this in the next subsection. The more general theorem is the following:

▶ **Theorem 2.** *For all $k \le d$, any* $\mathsf{OR} \circ \mathsf{AND} \circ \mathcal{C}$ *circuit where $\mathcal{C} = \{g : \{0,1\}^t \to \{0,1\}\}$ computing* $\mathsf{k\text{-}OV}_{n,d}$ *requires top fan-in $\Omega\left((\frac{n}{t})^k\right)$.*

However, throughout this paper, we stick to either $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuits or $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}$ circuits as the case may be.

It is important to note that the usual trick of using random restrictions to get rid of the bottom fan-in restriction in Theorem 1 is very unlikely to work as it is known that 2-OV becomes easy to compute by AC^0 circuits with high probability under random restrictions [16] (section 3).

As a secondary result, we show an exponential lower bound on the size of $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}$ circuits computing $\mathsf{2\text{-}OV}_{n,d}$ when d is very large:

▶ **Theorem 3.** *For all $k \le d$, any* $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}$ *circuit computing* $\mathsf{2\text{-}OV}_{n,d}$ *requires size $s \in \Omega(\min\{2^k, \left(\frac{d}{nk}\right)^n\})$. In particular, for $k = d/2n$ and $d \in \Omega(n^2)$, $s \in \Omega(2^n)$.*

Since $\mathsf{2\text{-}OV}_{n,d}$ reduces to $\mathsf{k\text{-}OV}_{n,d}$ by projections trivially, the above theorem holds for $\mathsf{k\text{-}OV}_{n,d}$ as well.

Techniques.

We note that throughout this paper, we work with the function $\mathsf{k\text{-}Int}_{n,d}$ defined as the negation of $\mathsf{k\text{-}OV}_{n,d}$. We do this because $\mathsf{k\text{-}Int}_{n,d}$ is a monotone function, and hence allows us several conveniences with regard to notation. Thus our lower bounds to $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}$ circuits computing $\mathsf{k\text{-}Int}_{n,d}$ transfer directly to $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuits computing $\mathsf{k\text{-}OV}_{n,d}$. More formally, $\mathsf{k\text{-}Int}_{n,d}$ is defined as

$$\mathsf{k\text{-}Int}_{n,d}(A_1, \ldots, A_k) = \bigwedge_{i_1,\ldots,i_k \in [n]} \bigvee_{j \in [d]} (a_{i_1}[j] \wedge \cdots \wedge a_{i_k}[j]) \tag{3}$$

Main result. For our main result, the strategy we use is that of *finite limit vectors*. This is a top-down strategy that was used by Håstad, Jukna, and Pudlák in [13] for proving depth-3 AC^0 circuit lower bounds. We briefly describe the approach.

Assume an $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}$ circuit $C = C_1 \wedge \cdots \wedge C_{s(n)}$ computes a function f. Then for any $\mathcal{N} \subseteq f^{-1}(0)$, by an averaging argument, there is a C_i that correctly outputs 0 on at least $1/s$ fraction of inputs in $\mathcal{N}$. Hence showing an upper bound to $C_i^{-1}(0) \cap \mathcal{N}$ implies a lower bound to $s(n)$ as $s \geq |\mathcal{N}|/|C_i^{-1}(0) \cap \mathcal{N}|$.

The technique of *finite limits* by [13] is used to show that C_i cannot be correct on many inputs in $\mathcal{N}$. The idea is to show that if $C_i^{-1}(0) \cap \mathcal{N}$ is large, then we can construct a 1-input y such that for any set of t input positions, it looks identical to *some string* in $C_i^{-1}(0) \cap \mathcal{N}$. Such a string is called a *t-limit* for the set $C_i^{-1}(0) \cap \mathcal{N}$. Then if the bottom gates in C_i can each see only t bits of the input, the string y fools all of them into evaluating to 0 *simultaneously*, and hence C_i will output 0 on y. This is a contradiction since $y \in C^{-1}(1)$ by construction, but $C_i(y) = 0$ implies $C(y) = 0$.

The key idea behind our construction of a t-limit is to first model any subset of *maxterms* of $\mathsf{k\text{-}Int}_{n,d}$ as a k-partite hypergraph such that the maxterms in the subset and the hyperedges are in bijection. Then we construct a t-limit for the case of $\mathsf{2\text{-}Int}_{n,d}$ be using König's theorem on this graph. To deal with the general case of $\mathsf{k\text{-}Int}_{n,d}$, we first show a sunflower lemma on the hypergraph, and then use the sunflower structure to construct a t-limit. We show a version of the sunflower lemma on our hypergraph that is *very slightly* less demanding than the standard sunflower lemma [8]. We note that this does not improve the asymptotic complexity of our final bound.

We remark that the technique of [13] described above is impervious to the fan-in of the middle OR gates. Further, the proof strategy does not use the fact that the bottom gates are AND gates. As long as the fan-in of the gates at the bottom layer is restricted to t, the lower bound follows by definition of t-limit. For instance, even if we allowed the gates at the bottom to compute linear threshold functions over at most t variables, computing $\mathsf{k\text{-}Int}_{n,d}$ will still require size $\Omega((n/t)^k)$.

Secondary result. The exponential lower bound of [13] for $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuits computing the iterated intersection function $S_{n,d}$ for $d \in \sqrt{n}$ is of particular interest to us. The function $S_{n,d}$ bears a close resemblance to $\mathsf{2\text{-}Int}_{n,d}$. While $S_{n,d}$ is the *iterated* intersection, $\mathsf{2\text{-}Int}_{n,d}$ can be seen as "all-pairs" intersection.

We show a reduction (via projections) from $S_{n,d/n}$ to $\mathsf{2\text{-}Int}_{n,d}$. The blow-up in the dimension of vectors is rather hurtful, and we can conclude non-trivial lower bounds only for $d \in \omega(n)$.

2 Preliminaries

We often interpret a d-dimensional vector $u \in \{0,1\}^d$ as the characteristic vector of a subset of $[d]$.

▶ **Definition 4** ($\mathsf{k\text{-}OV}_{n,d}$). *For tuples $A_1, A_2, \ldots, A_k \subseteq \{0,1\}^d$ where $\forall i \in [k]$, $|A_i| = n$.*

$$\mathsf{k\text{-}OV}_{n,d}(A_1, A_2, \ldots, A_k) = 1 \iff \exists a_1 \in A_1, \exists a_2 \in A_2, \cdots, \exists a_k \in A_k, \text{ such that } a_1 \cap a_2 \cap \cdots \cap a_k = \emptyset$$

For notational convenience, we work with the negation of $\mathsf{k\text{-}OV}_{n,d}$ throughout the paper. We use $\mathsf{k\text{-}Int}_{n,d}$ to denote the negation of $\mathsf{k\text{-}OV}_{n,d}$, and is defined as follows:

▶ **Definition 5** (k-Int$_{n,d}$)**.** *For tuples $A_1, A_2, \dots, A_k \subseteq \{0,1\}^d$ where $\forall i \in [k]$, $|A_i| = n$.*

$$\mathsf{k\text{-}Int}_{n,d}(A_1, A_2, \dots, A_k) = 1 \iff \forall a_1 \in A_1, \forall a_2 \in A_2, \cdots, \forall a_k \in A_k, \text{ we have } a_1 \cap a_2 \cap \cdots \cap a_k \neq \emptyset$$

An input to the function k-Int$_{n,d}$ has nk vectors, each of dimension d. Hence nkd many input bits in total.

For any $x, y \in \{0,1\}^d$, we write $x \leq y$ if $\forall i, x_i \leq y_i$.

▶ **Definition 6** (Monotone function)**.** *We say that a Boolean function f is monotone if $\forall x, y \in \{0,1\}^d$ such that $x \leq y$, we have $f(x) \leq f(y)$.*

For monotone functions such as k-Int$_{n,d}$, we can define *maximal* 0*-inputs*:

▶ **Definition 7.** *(Maximal* 0*-input) Let f be a monotone Boolean function. An input x is a* maximal 0-input *for f if $f(x) = 0$ and for all strings y such that $x \leq y$, $f(y) = 1$.*

Throughout this article, we will use the term "*maxterm*" and "maximal 0-inputs" interchangeably. This deviates from the standard definition of *maxterm*, but is very convenient in our context.

For a vector $u \in \{0,1\}^d$, and a set of indices $S \subseteq [d]$, we denote the restriction of u to the indices in S as $u|_S$.

▶ **Definition 8** (t-limit)**.** *A vector $y \in \{0,1\}^m$ is said to be a* t-limit *for a set $B \subseteq \{0,1\}^m$ if and only if $\forall S \subseteq [m]$ with $|S| = t$, $\exists x \in B$ such that $y \neq x$ but $y|_S = x|_S$. Further, $y \in \{0,1\}^m$ is said to be an* upper t-limit *if $y \geq x$.*

We will always assume that the depth-3 circuits we consider are layered. i.e., inputs are read directly by only the gates at the bottom layer, and every layer reads outputs from the layer below it. This assumption does not affect asymptotic complexity. We say a depth-3 circuit C has *bottom positive fan-in* (*bottom negation fan-in*) t if for every gate in the bottom layer, at most t of its inputs are positive literals (negated literals respectively).

We denote the permutation group on k distinct elements with S_k. Let $\mathcal{P} = (P_1, \dots, P_k)$ be an ordered partition of $[d]$ into k parts. For any permutation $\sigma \in \mathsf{S}_k$, we use $\mathcal{P}_\sigma$ to denote the ordered partition obtained by permuting the parts of $\mathcal{P}$ using σ. i.e., $\mathcal{P}_\sigma \triangleq (P_{\sigma(1)}, \dots, P_{\sigma(k)})$

3 AND ∘ OR ∘ AND circuits

To describe the lower bound for k-Int$_{n,d}$ against AND ∘ OR ∘ AND circuits, we first identify a special set of maxterms (maximal 0-inputs) of k-Int$_{n,d}$. We do this by explicitly constructing such inputs.

3.1 Maxterms of k-Int$_{n,d}$

Fix any integer $k > 1$ and $d \in \mathbb{N}$. For any choice of $n_1, \dots, n_k \in [n]$, and any ordered partition $\mathcal{P} = (P_1, \dots, P_k)$ of $[d]$ into k parts, we will construct an input $N = (A_1 \dots, A_k)$ where $A_i \subseteq \{0,1\}^d$ with $|A_i| = n$ such that N is a maxterm for k-Int$_{n,d}$. Throughout, we will denote the j'th vector in A_i by a_i^j.

The input $N = (A_1 \dots, A_k) \in \{0,1\}^{nkd}$ is constructed as follows:

- Set every vector other than $a_1^{n_1}, \dots, a_k^{n_k}$ to all 1s.
- In each $a_i^{n_i}$, set the indices contained in P_i to 0s. Set every other position to 1. Formally, for all $i \in [k]$, set $a_i^{n_i}|_{P_i} \leftarrow 0^{|P_i|}$ and $a_i^{n_i}|_{[d]\setminus P_i} \leftarrow \vec{1}$.

We shall call $((n_1, \ldots, n_k), \mathcal{P})$ the *support* of N, and denote it by $\sup(N)$.

To see that N is indeed a maxterm of $\mathsf{k\text{-}Int}_{n,d}$, observe that since $\mathcal{P}$ is a partition of $[d]$, for every position $\ell \in [d]$, there is a *unique* $i \in [k]$ such that $\ell \in P_i$. Therefore, by construction of N, $a_i^{n_i}[\ell] = 0$. So for every position ℓ, there is some vector among $a_1^{n_1}, \ldots, a_k^{n_k}$ that is 0 in position ℓ, and hence $a_1^{n_1} \cap \cdots \cap a_k^{n_k} = \emptyset$. Moreover, due to i being unique for each such ℓ, we also have $a_j^{n_j}[\ell] = 1$ for all $j \neq i$. So changing $a_i^{n_i}[\ell]$ from 0 to 1 results in the vectors intersecting at ℓ. Combining this with the fact that every vector in N other than $a_1^{n_1}, \ldots, a_k^{n_k}$ is the all-1s vector, we conclude that N is indeed a maximal 0-input.

We will be particularly interested in a subset of such maxterms of $\mathsf{k\text{-}Int}_{n,d}$ that are formed by the permutations of the parts of some fixed partition into non-empty parts. We define this formally as follows.

▶ **Definition 9.** *(Permutation-maxterms) Fix an ordered partition $\mathcal{P} = (P_1, \ldots, P_k)$ of $[d]$ into k non-empty parts. A* permutation-maxterm *with respect to $\mathcal{P}$ is any maxterm N constructed as above that has $\sup(N) = ((n_1, \ldots, n_k), \mathcal{P}_\sigma)$ for some $n_1 \ldots, n_k \in [n]$ and $\sigma \in \mathsf{S}_k$.*

We shall use $\mathcal{N}_{\mathcal{P}}^{n,k,d}$ to denote the set of all permutation-maxterms of $\mathsf{k\text{-}Int}_{n,d}$ with respect to some ordered partition $\mathcal{P}$ of $[d]$ into k non-empty parts. We drop the subscript, and superscripts if it is clear from context.

Note that for any partition $\mathcal{P}$ as in the definition above, $|\mathcal{N}_{\mathcal{P}}^{n,k,d}| = n^k k!$ as there are n^k many k-tuples $(n_1, \ldots, n_k)$ and $k!$ many permutations in S_k.

▶ Remark 10. We will always assume that $\mathcal{P}$ is an ordered partition of $[d]$ into k non-empty parts. In doing so, we can safely assume that every permutation-maxterm with respect to $\mathcal{P}$ has no $\vec{0}$ vectors.

3.2 Support Graph

We define a k-partite hypergraph to encode, and reason about, the relationship between permutation-maxterms of $\mathsf{k\text{-}Int}_{n,d}$. Here, by k-partite hypergraph we mean that every hyperedge must contain exactly one vertex from each part.

Fix $k \geq 2$ and $d \geq k$, and any ordered partition $\mathcal{P}$ of $[d]$ into k non-empty parts. For any subset $S \subseteq \mathcal{N}_{\mathcal{P}}^{n,k,d}$ of permutation-maxterms of $\mathsf{k\text{-}Int}_{n,d}(A_1, \ldots, A_k)$, we define the *support graph* of S as a k-partite hypergraph $\mathcal{G}_S = (V_1 \cup \cdots \cup V_k, E)$ as follows. As usual we will use a_i^j to denote the j'th vector in A_i. Corresponding to each vector $a_i^j \in A_i$, we include k vertices in V_i denoted $v_i^{j,1}, \ldots, v_i^{j,k}$. So for all $i \in [k]$, we have $|V_i| = nk$ and hence the graph $\mathcal{G}_S$ is on nk^2 many vertices.

We define the set E of hyperedges as follows:

$$\left(v_1^{n_1,b_1}, \ldots, v_k^{n_k,b_k}\right) \in E \iff \exists \text{ maxterm } N \in S \text{ such that}$$
$$\sup(N) = ((n_1, \ldots, n_k), \mathcal{P}_\sigma) \text{ and } b_i = \sigma(i)\ \forall i \in [k]$$

▶ Remark 11. Note that the set of maxterms $S \subseteq \mathcal{N}_{\mathcal{P}}$ and the set of hyperedges in $\mathcal{G}_S$ are in bijection. More precisely, a maxterm N with $\sup(N) = ((n_1, \ldots, n_k), \mathcal{P}_\sigma)$ corresponds to the hyperedge $\left(v_1^{n_1,\sigma(1)}, \ldots, v_k^{n_k,\sigma(k)}\right)$ and vice-versa.

▶ **Definition 12** (Co-disjoint). *We call two vectors $u \in \{0,1\}^d$ and $v \in \{0,1\}^d$ as* co-disjoint *if and only if $\overline{u} \cap \overline{v} = \emptyset$. i.e., the set of positions where u is 0, and the set where v is 0 are disjoint.*

For two tuples of vectors $A = (a_1, \dots, a_n)$ and $B = (b_1, \dots, b_n)$ where $a_i, b_i \in \{0,1\}^d$, we say A and B are co-disjoint if for all $i \in [n]$, a_i and b_i are co-disjoint.

Maxterms $M = (M_1, \dots, M_k)$ and $N = (N_1, \dots, N_k)$, both from $\mathcal{N}_{\mathcal{P}}^{n,k,d}$, are said to be co-disjoint if and only if for all $i \in [k]$, M_i and N_i are co-disjoint.

Intuitively, the graph $\mathcal{G}_S$ records where the 0s in each of the maxterms in S appear. This gives us the following close connection between co-disjointness of vectors across maxterms, and disjointness of their hyperedges.

▶ **Lemma 13.** *Let $S \subseteq \mathcal{N}_{\mathcal{P}}^{n,k,d}$, and let $\mathcal{G}_S = (V_1 \cup \dots \cup V_k, E)$ be its support graph. Let $M = (M_1, \dots, M_k)$ and $N = (N_1, \dots, N_k)$ be two maxterms from S and let E_M, and E_N respectively, denote their corresponding hyperedges in $\mathcal{G}_S$. Then for each $i \in [k]$, we have the following two properties:*
1. *If E_M and E_N share a vertex in V_i, then $M_i = N_i$.*
2. *If E_M and E_N contain different vertices from V_i, then M_i and N_i are co-disjoint.*

Proof. Let $\sup(M) = (a_1, \dots, a_k, \mathcal{P}_\sigma)$ and $\sup(N) = (b_1, \dots, b_k, \mathcal{P}_\pi)$.
Proof of (1): If E_M and E_N share a vertex in V_i for some $i \in [k]$, then $v_i^{a_i,\sigma(i)} = v_i^{b_i,\pi(i)}$ and so we have $a_i = b_i$ and $\sigma(i) = \pi(i)$. Let $\ell = a_i = b_i$, and let $q = \sigma(i) = \pi(i)$. Then by construction of the maxterms M and N, all vectors in M_i other than m_i^ℓ are all 1s, and similarly all vectors in N_i other than n_i^ℓ are all 1s. The vector m_i^ℓ and n_i^ℓ both have 0s in indices from the part P_q, and 1s elsewhere. So $m_i^\ell = n_i^\ell$. Hence the tuple M_i and N_i are identical.

Proof of (2): If E_M and E_N have different vertices from V_i, then $v_i^{a_i,\sigma(i)} \neq v_i^{b_i,\pi(i)}$. So either $a_i \neq b_i$ or $\sigma(i) \neq \pi(i)$ (or both). The claim holds in both cases:
- If $a_i \neq b_i$, then recall that by construction, the only vector that has 0s in M_i is the vector $m_i^{a_i}$. Every other vector in M_i, and in particular $m_i^{b_i}$ is the all 1s vector by construction. So the tuples of vectors M_i and N_i cannot both be 0 in any vector in any position.
- Else $a_i = b_i$ and $\sigma(i) \neq \pi(i)$. By our construction of maxterms, the 0s in the vectors $m_i^{a_i}$ and $n_i^{b_i = a_i}$ are in the indices given by $P_{\sigma(i)}$ and $P_{\pi(i)}$ respectively. Since $\mathcal{P}$ is a partition, and $\sigma(i) \neq \pi(i)$, $P_{\sigma(i)} \cap P_{\pi(i)} = \emptyset$. Therefore there cannot be an index where both $m_i^{a_i}$ and $n_i^{b_i}$ are both 0.

◀

The following lemma follows directly from Lemma 13:

▶ **Lemma 14.** *Let $S \subseteq \mathcal{N}_{\mathcal{P}}^{n,k,d}$ be a set of maxterms such that all hyperedges in $\mathcal{G}_S$ are pairwise vertex-disjoint. Then the maxterms in S are pairwise co-disjoint. (i.e., for all positions $\ell \in [nkd]$, there is at most one maxterm in S that has 0 in the ℓ'th position.)*

Proof. Let $M, N \in S$ be any two maxterms, and let the vertex set of $\mathcal{G}_S$ be $V = V_1 \cup \dots \cup V_k$. The hyperedges E_M and E_N, corresponding to M, and N respectively, are vertex-disjoint from the premise. So for each $i \in [k]$, E_M and E_N contain different vertices from V_i. Applying Lemma 13 to G_S, we obtain that M_i and N_i are co-disjoint for all $i \in [k]$. Hence there is no position where both M and N are 0 by definition of co-disjoint. ◀

3.3 Warm-up: 2-$\mathsf{Int}_{n,d}$

We give a self-contained proof of our lower bound for the case of 2-$\mathsf{Int}_{n,d}$ that demonstrates the strategy behind the proof for the general case.

▶ **Theorem 15.** *For all $d > 1$, any* AND $\circ$ OR $\circ$ AND *circuit with bottom fan-in t computing* $2\text{-}\mathsf{Int}_{n,d}$ *requires top fan-in at least $2n^2/t^2$.*

Proof. Let $C = C_1 \wedge C_2 \wedge \cdots \wedge C_s$ be an $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}_t$ circuit with bottom fan-in t computing $2\text{-}\mathsf{Int}_{n,d}$. Let $\mathcal{P} = (P_1, P_2)$ be any ordered partition of $[d]$ into two non-empty parts. Consider the permutation-maxterms $\mathcal{N} = \mathcal{N}_{\mathcal{P}}^{n,2,d}$ of $2\text{-}\mathsf{Int}_{n,d}$ as described in definition 9. Since $\mathcal{N}$ is a subset of the 0-inputs of $2\text{-}\mathsf{Int}_{n,d}$, the circuit C outputs 0 on every input in $\mathcal{N}$. By an averaging argument, there exists $i \in [s]$ such that C_i correctly outputs 0 on at least $1/s$ fraction of inputs in $\mathcal{N}$. We will show that $|C_i^{-1}(0) \cap \mathcal{N}| \leq t^2$. Then the theorem follows as:

$$\frac{2n^2}{s} = \frac{1}{s}\,|\mathcal{N}| \leq |C_i^{-1}(0) \cap \mathcal{N}| \leq t^2$$

In the following, we will show that $\forall S \subseteq \mathcal{N}$ with $|S| > t^2$, there is a t-limit $y \in C^{-1}(1)$ for S. This will imply that $|C_i^{-1}(0) \cap \mathcal{N}| \leq t^2$. To see why, let $C_i = g_1 \vee g_2 \cdots \vee g_\ell$ with each g_j having fan-in at most t. Suppose $S \subseteq C_i^{-1}(0)$ is a subset of vectors such that there is a string $y \in C^{-1}(1)$ that is a t-limit for S. Then, by definition of t-limit, for all $T \subseteq [nkd]$ with $|T| = t$, there exists $x \in S$ such that $x|_T = y|_T$. Now each of the gates g_j is a function of at most t variables, and we know that for all inputs $x \in S$, we have $g_j(x) = 0$ for all $j \in [\ell]$. Since y looks identical to some string in S when restricted to these t positions, all the g_j will output 0 on y too. This forces $C_i(y) = 0$ leading to a contradiction since $y \in C^{-1}(1)$.

Let $S \subseteq \mathcal{N}$ be any set with size $|S| > t^2$ and let $\mathcal{G}_S$ be its support graph. Note that since $k = 2$, $\mathcal{G}_S$ is a bipartite graph with simple edges rather than hyperedges, and every maxterm in S corresponds to an edge in $\mathcal{G}_S$ and vice versa. We claim at least one of the following is true for $\mathcal{G}_S$:

(i) There exists a matching of size $t + 1$ in $\mathcal{G}_S$

(ii) There exists a vertex of degree at least $t + 1$ in $\mathcal{G}_S$.

Indeed this is a consequence of König's theorem: suppose the size of a maximum matching is at most than t, then by König's Theorem, the minimum vertex-cover has size at most t. Since there are $|S|$ many edges in $\mathcal{G}_S$, there must be a vertex v in the vertex cover with degree at least $\frac{|S|}{t}$. Since $|S| > t^2$, it must be that $\deg(v) > t$ which satisfies (ii). In both the above cases, we construct a string $y \in C^{-1}(1)$ that is a t-limit for S.

- Case (i): Consider the set S' of maxterms corresponding to the edges in a maximum matching of $\mathcal{G}_S$. Then S' is a set of at least $t + 1$ pairwise co-disjoint maxterms. Then $y \triangleq \vec{1}$ is a t-limit for S'. To see why, consider any set of t positions. By Lemma 14, at each of these positions, at most one of maxterms can be 0. Since there are $t + 1$ such maxterms and only t positions, there must be a maxterm where the value at all the given positions is 1, thus looking identical to y.
- Case (ii): Let the vertex set of $\mathcal{G}_S$ be $V = V_1 \cup V_2$. Without loss of generality, let the vertex v with $\deg(v) > t$ be in V_1. Let E be the edges that have v as one endpoint, and let $M_E \subseteq S$ be the maxterms corresponding to the edges in E. Then by property (1) of Lemma 13, the first tuple of vectors in all these maxterms is the same. Let A_1 be the first tuple of vectors. We construct the input $y = (Y_1, Y_2)$ as follows: set $Y_1 \leftarrow A_1$, and set $Y_2 \leftarrow \vec{1}$.

  Since the string y was obtained by taking first tuple of a *maxterm*, and setting every vector in the 2nd tuple to 1, it must be a 1-input.

  To see that y is a t-limit, take any subset of indices $T \subseteq [2nd]$ with $|T| = t$. We will show that one of the maxterms in M_E looks identical to y in these t positions. For every position from $[nd]$ (the 1st tuple of vectors), *every* maxterm in M_E is identical to y since

$Y_1 = A_1$. So assume that all indices in T are from the range $\{nd+1, \ldots, 2nd\}$. By construction, y is all-1s in this range of indices. Since edges in E have distinct endpoints in V_2, property (2) of Lemma 13 tells us that the second tuple of vectors in the maxterms in T are pairwise co-disjoint. This is similar to case (i): we have $|M_E| \geq t+1$ many maxterms such that for any position in T, at most one of them is 0, and there are only t positions in T. So by the pigeon-hole principle, there must be a maxterm in M_E that has 1 in all positions from T, thus looking identical to y in these positions.

◀

Since $2\text{-}\mathsf{OV}_{n,d}$ is the negation of $2\text{-}\mathsf{Int}_{n,d}$, the following is an immediate corollary of Theorem 15.

▶ **Corollary 16.** *For all $d > 1$, any $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuit with bottom fan-in t computing $2\text{-}\mathsf{OV}_{n,d}$ requires top fan-in at least $2n^2/t^2$.*

▶ Remark 17. It is easy to see that the t-limit string y constructed in the proof of Theorem 15 is in fact an *upper* t-limit. Therefore the lower bound shown for $2\text{-}\mathsf{Int}_{n,d}$ works against a slightly more general class of circuits — $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}$ circuits that have each bottom AND-gate seeing at most t positive literals. Analogously the lower bound for $2\text{-}\mathsf{OV}_{n,d}$ works against $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuits where each bottom gate has at most t negated inputs.

3.4 General case: $\mathsf{k\text{-}Int}_{n,d}$

We will need the following lemma on k-partite hypergraphs:

▶ **Lemma 18.** *Let G be a k-partite hypergraph with m many hyperedges. Then for all $t > 0$ at least one of the following holds:*

(i) *There are more than t vertex-disjoint hyperedges in G.*
(ii) *There is a vertex u such that $\deg(u) > \lfloor \frac{m}{kt} \rfloor$*

Proof. Let G be a k-partite hypergraph with m hyperedges. Let S be a largest set of vertex-disjoint hyperedges in G. If $|S| > t$, then the lemma is true. Suppose $|S| \leq t$. Let V_S be the set of vertices participating in the hyperedges in S. Since each hyperedge contains exactly k many vertices, $|V_S| \leq kt$. Also, since S is a largest such set, each of the remaining hyperedges must contain at least one vertex from V_S. Therefore, by an averaging argument, there is a vertex $u \in V_S$ that is part of at least $\frac{m-|S|}{|V_S|}$ many hyperedges outside S, and 1 hyperedge in S. Therefore, we have:

$$deg(u) \geq \frac{m-|S|}{|V_S|} + 1 \geq \frac{m-t}{kt} + 1 = \frac{m}{kt} - \frac{1}{k} + 1 > \left\lfloor \frac{m}{kt} \right\rfloor$$

◀

We use Lemma 18 to show that if we start with enough hyperedges, then there is a subset of them such that in each part, either all of them coincide, or they are all distinct.

▶ **Lemma 19.** *Let $k \geq 2$, and let $G = (V_1 \cup \cdots V_k, E)$ be a k-partite hypergraph with $|E| > \frac{k!t^k}{2}$. Then there exists $S \subseteq E$ with $|S| > t$ such that for each $i \in [k]$, exactly one of the following holds:*

1. *There exists a vertex $u \in V_i$ such that all hyperedges in S share the vertex u.*
2. *No two hyperedges in S share the same vertex in V_i.*

Proof. Induction on k. Base case $k = 2$ is a consequence of König's theorem: Since $k = 2$, G is just a bipartite graph. If there is a matching in G of size more than t, then let S be the edges in such a matching. Clearly the edges in S are vertex-disjoint and statement (2) holds. Else the maximum matching has size $\leq t$. Then König's theorem implies that the minimum vertex cover has size at most t. By an averaging argument, there must exist a vertex u such that $deg(u) > |E|/t = \frac{k!t^k}{2t} = \frac{2t^2}{2t} = t$. Define S to be the set of edges that share u. Without loss of generality, let $u \in V_1$. Then all edges in S must have distinct vertices in V_2. Therefore in V_1, they all coincide, and in V_2 they are all distinct.

Case $k > 2$: Apply Lemma 18 to G. If (i) holds, then we have a set S of more than t vertex-disjoint hyperedges. This means for all $i \in [k]$, statement (2) holds and we are done.

Suppose (ii) holds, then there is a vertex u such that $\deg(u) > \lfloor m/kt \rfloor = \frac{(k-1)!\,t^{k-1}}{2}$. Let S be the set of all hyperedges that contain vertex u. Then $|S| = \deg(u)$. Let $z \in [k]$ be such that $u \in V_z$.

We construct a $(k-1)$-partite hypergraph $G' = (V', E')$ by removing V_z, and the z'th coordinate from each edge. More formally:

$$V' \triangleq V_1 \cup \cdots \cup V_{z-1} \cup V_{z+1}, \cdots \cup V_k$$
$$E' \triangleq \{(v_1, \ldots, v_{z-1}, v_{z+1}, \ldots, v_k) \mid (v_1, \ldots, v_{z-1}, u, v_{z+1}, v_k) \in S\}$$

(Informally, an edge $e' \in E'$ is just an edge $e \in S$ with its z'th coordinate removed.)

Note that $|E'| = |S|$. This is because $\forall e_1, e_2 \in S$ such that $e_1 \neq e_2$, the edges e_1 and e_2 share the vertex u in V_z. So there must exist $j \neq z$ such that e_1 and e_2 use different vertices in V_j. Hence $e'_1 \neq e'_2$. Further, observe that for any $i \neq z$, $e'_1, e'_2 \in E'$ share a vertex in V'_i if and only if e_1 and e_2 share the same vertex in V_i.

Now G' is a $(k-1)$-partite hypergraph with $|E'| = |S| > \frac{(k-1)!\,t^{k-1}}{2}$ many hyperedges. By induction on G', for each $i \neq z$, either all hyperedges in E' share a vertex in V'_i, or they use distinct vertices in V'_i. By a previous observation, this means for all $i \neq z$, all hyperedges in S share a vertex in V_i, or they use distinct vertices in V_i. We already know that all edges in S share the same vertex in V_z, namely u. Hence for all $i \in [k]$, the edges in S satisfy (1) or (2). ◀

▶ Remark 20. The statement of Lemma 18 can be seen as a sunflower lemma. Take any vertex u in the graph G that participates in at least one hyperedge from S. Then exactly one of the following holds: (i) The vertex u participates in exactly one hyperedge in S, or (ii) The vertex u participates in all hyperedges in S. The standard sunflower lemma would require more than $k!\,t^k$ hyperedges, while our statement needs half of that.

We now describe how to construct an upper t-limit in the general case.

▶ **Lemma 21.** *Let $\mathcal{M} \subseteq \mathcal{N}_{\mathcal{P}}^{n,k,d}$ be any set of permutation-maxterms of $\mathsf{k\text{-}Int}_{n,d}$ for any $k \geq 2$ and $d \geq k$. If $|\mathcal{M}| > \frac{k!\,t^k}{2}$, then there is a string $y \in \mathsf{k\text{-}Int}_{n,d}^{-1}(1)$ that is an upper t-limit for $\mathcal{M}$.*

Proof. Let $G_{\mathcal{M}} = (V, E)$ be the k-partite support graph of $\mathcal{M}$ (defined in section 3.2), and let $V = V_1 \cup \cdots \cup V_k$. By Lemma 19, there exists a set of hyperedges $S \subseteq E$ with $|S| \geq t + 1$ such that for each $i \in [k]$, either all edges in S share the same vertex in V_i, or no two edges share a vertex of V_i. Let M_S be the set of maxterms corresponding to S.

Let $B \subseteq [k]$ be the set of all indices $i \in [k]$ such that all edges in S share the same vertex in V_i. Then $\overline{B}$ contains indices of parts where the edges in S use distinct vertices. By property (1) of Lemma 13, this implies that for each $i \in B$, the i'th tuple of vectors in the maxterms in M_S are identical. For each $i \in B$, denote the i'th tuple of vectors in all these maxterms with A_i.

We construct the string $y = (Y_1, \ldots, Y_k)$ as follows:

$$\forall i \in B, \text{ set } Y_i \leftarrow A_i$$
$$\forall j \in \overline{B}, \text{ set } Y_j \leftarrow \vec{1}$$

y is a 1-input of $\mathsf{k\text{-}Int}_{n,d}$:

Observe that y can also be obtained by starting with any maxterm $N = (N_1, \ldots, N_k)$ from S, and setting to 1s all vectors in N_j for all $j \in \overline{B}$. Since N is a maxterm (maximal 0-input), the string y must be a 1-input. This also means that the string y is pointwise greater than or equal to any maxterm in S.

y is a t-limit:

Let $T \subseteq [nkd]$ with $|T| = t$ be a set of any t positions. For all $i \in B$, the string y is identical to *every* maxterm in M_S. So assume that T only has positions that fall into tuples indexed by $\overline{B}$. By property (2) of Lemma 13, the maxterms in M_S are pairwise co-disjoint on all such positions. i.e., for any position $\ell \in T$, at most one maxterm in M_S can be 0. So we have t positions, and $|M_S| = |S| \geq t + 1$ maxterms. By pigeon-hole principle, there exists a maxterm in M_S that is 1 on all these t positions, thus looking identical to y.

Since y is pointwise greater or equal to every maxterm in S, we conclude that indeed y is an upper t-limit to $\mathcal{M}$. ◀

▶ **Lemma 22.** *Let C be any $\mathsf{OR} \circ \mathsf{AND}$ circuit with bottom positive fan-in t computing a function f on n variables. Let y be any string that is an upper t-limit to $f^{-1}(0)$. Then $C(y) = 0$.*

Proof. Let g be any bottom AND-gate of C. Let $P \subseteq [n]$ ($Q \subseteq [n]$) be the variables whose positive literals (negated literals resp.) are input to g. Then $|P| \leq t$ by assumption.

Since y is an upper t-limit to $g^{-1}(0)$, it must be that for every set T of t positions there exists a string $x^{(T)} \in g^{-1}(0)$ such that $y|_T = x^{(T)}|_T$. In particular, this holds for the set P. So in all positions from P, the gate g sees no difference between y and $x^{(T)}$.

The gate g sees negative literals of all variables from Q. Since y is an *upper* t-limit, we have $x^{(T)}|_Q \leq y|_Q$. Hence for all $i \in Q$ such that $\neg x_i = 0$, we also have $\neg y_i = 0$. Hence $g(y) \leq g(x^{(T)}) = 0$ as $x^{(T)} \in g^{-1}(0)$. ◀

▶ **Theorem 23.** *For all k, d such that $k \leq d$, any $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}$ circuit with bottom positive fan-in t computing $\mathsf{k\text{-}Int}_{n,d}$ requires top fan-in $\Omega\left(\left(\frac{n}{t}\right)^k\right)$.*

Proof. Let $C = C_1 \wedge \cdots \wedge C_s$ be an $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}_t$ circuit with bottom positive fan-in t, computing $\mathsf{k\text{-}Int}_{n,d}$. Consider the set $\mathcal{N} = \mathcal{N}_{\mathcal{P}}^{n,k,d}$ of all permutation-maxterms of $\mathsf{k\text{-}Int}_{n,d}$. Since C outputs 0 on all inputs from $\mathcal{N}$, there must be some $\mathsf{OR} \circ \mathsf{AND}_t$ subcircuit C_i that correctly outputs 0 on at least $1/s$ fraction of inputs in $\mathcal{N}$. We will show that $|C_i^{-1}(0) \cap \mathcal{N}| \leq k!\, t^k/2$, and the theorem follows since:

$$\frac{k!\, n^k}{s} = \frac{1}{s}|\mathcal{N}| \leq |C_i^{-1}(0) \cap \mathcal{N}| \leq \frac{k!\, t^k}{2}$$

Let $\mathcal{M} = C_i^{-1}(0) \cap \mathcal{N}$. Suppose, for the sake of contradiction, $|\mathcal{M}| > k!\, t^k/2$. Since $\mathcal{M} \subseteq \mathcal{N}$, we apply Lemma 21 to conclude that there exists a string $y \in \mathsf{k\text{-}Int}_{n,d}^{-1}(1)$ that is an upper t-limit y for $\mathcal{M}$. Then by Lemma 22, it must be that $C(y) = 0$. But this is a contradiction since $y \in \mathsf{k\text{-}Int}_{n,d}^{-1}(1)$. ◀

Since $\mathsf{k\text{-}OV}_{n,d}$ is the negation of $\mathsf{k\text{-}Int}_{n,d}$, the following is an immediate corollary of Theorem 23.

▶ **Theorem 1.** *For all $k \leq d$, any $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuit with bottom fan-in t computing $\mathsf{k\text{-}OV}_{n,d}$ requires top fan-in $\Omega\left((\frac{n}{t})^k\right)$.*

4 OR ∘ AND ∘ OR circuits

In this section, we show that any $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuit requires exponential size to compute $\mathsf{2\text{-}Int}_{n,d}$ for any $d \in \Omega(n^2)$. This result is a consequence of a known lower bound for the iterated intersection function defined as follows:

▶ **Definition 24** (Iterated Intersection). *Let $A, B \subseteq \{0,1\}^d$ be tuples of d dimensional vectors with $|A| = |B| = n$.*

$$S_{n,d}(A,B) = 1 \iff \forall i \in [n] \text{ we have } a_i \cap b_i \neq \emptyset$$

where a_i and b_i are the i'th vector in A and B respectively.

The function $S_{n,d}$ can also be defined using an $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}_2$ circuit of size nd:

$$S_{n,d}(A,B) = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{d} a_i[j] \wedge b_i[j]$$

The result by Håstad, Jukna, Pudlák in [13] shows the following lower bound for computing $S_{n,d}$ by $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuits:

▶ **Proposition 25** ([13]). *For all $k \leq nd$, any $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuit computing $S_{n,d}$ requires size $\min\{2^k, (d/k)^n\}$*

In particular, Proposition 25 shows that $S_{\sqrt{n},\sqrt{n}}$ requires $2^{\sqrt{n}}$ size $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuits. This can be used to show lower bounds for $\mathsf{2\text{-}Int}_{n,d}$:

▶ **Theorem 26.** *Let C be an $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}$ circuit computing $\mathsf{2\text{-}Int}_{n,d}$. Then for all $k \leq d$, size of C is at least $\min\{2^k, \left(\frac{d}{nk}\right)^n\}$.*

Proof. We show this by reducing $S_{n,\lfloor d/n \rfloor}$ to $\mathsf{2\text{-}Int}_{n,d}$ via projections. Let $d' = \lfloor d/n \rfloor$. Take any instance $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ with $a_i, b_i \in \{0,1\}^{d'}$ of $S_{n,d'}$. We create two sets of d-dimensional vectors $A' = (a'_1, \ldots, a'_n)$ and $B' = (b'_1, \ldots, b'_n)$ that serve as an instance of $\mathsf{2\text{-}Int}_{n,d}$ as follows — for all $i \in [n]$, define $a'_i = 1^{(i-1)d'}\, a_i\, 1^{(n-i)d'}$ and $b'_i = 0^{(i-1)d}\, b_i\, 0^{(n-i)d}$. Note that the dimension of each a_i and b_i is $nd' \leq d$.

Observe that a_i and b_i are disjoint if and only if a'_i and b'_i are disjoint. So if (A, B) was a 0-instance of $S_{n,d'}$, then (A', B') is a 0-instance of $\mathsf{2\text{-}Int}_{n,d}$.

Further, if $b_j \neq \vec{0}$ for some $j \in [n]$, then for all $i \neq j$, we have $a'_i \cap b'_j \neq \emptyset$. To see this, observe that if $b_j \neq \vec{0}$, then there is some position $p \in [(j-1)d + 1, jd]$ such that $b'_j[p] = 1$. But by construction, the vector a'_i is 1 everywhere outside the interval $[(i-1)d + 1, id]$. Since $i \neq j$, the vector a'_i must be 1 at position p.

If (A, B) was a 1-instance of $S_{n,d'}$, then all a_i intersect b_i. This means all b_i are non-zero vectors. Thus for all $i, j \in [n]$, $a'_i \cap b'_j \neq \emptyset$.

The above reduction shows that C can be used to compute $S_{n,\lfloor d/n \rfloor}$. Applying Proposition 25 to C tells us that C must have size at least $\min\{2^k, \left(\frac{d}{nk}\right)^n\}$ for all $k \leq d$. ◀

Our reduction in proof of Theorem 26 inflates the dimension of vectors by a factor of n making the obtained bound trivial when $d \in O(n)$. However, we can still conclude an exponential lower bound by substituting $k = d/2n$ that gives us a lower bound of $\min\{2^{d/2n}, 2^n\} \in 2^{\Omega(n)}$ when $d \in \Omega(n^2)$.

Since 2-$\mathsf{OV}_{n,d}$ is the negation of 2-$\mathsf{Int}_{n,d}$, the following is an immediate corollary.

▶ **Theorem 3.** *For all $k \leq d$, any* AND ∘ OR ∘ AND *circuit computing* 2-$\mathsf{OV}_{n,d}$ *requires size $s \in \Omega(\min\{2^k, \left(\frac{d}{nk}\right)^n\})$. In particular, for $k = d/2n$ and $d \in \Omega(n^2)$, $s \in \Omega(2^n)$.*

References

1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 59–78. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.14.

2 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 51–58. ACM, 2015. doi:10.1145/2746539.2746612.

3 Arturs Backurs and Piotr Indyk. Which regular expression patterns are hard to match? In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 457–466. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.56.

4 Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 661–670. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.76.

5 Karl Bringmann and Wolfgang Mulzer. Approximability of the discrete fréchet distance. *J. Comput. Geom.*, 7(2):46–76, 2016. doi:10.20382/jocg.v7i2a4.

6 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. On the exact complexity of evaluating quantified k-cnf. In *Parameterized and Exact Computation - 5th International Symposium, IPEC 2010, Chennai, India, December 13-15, 2010. Proceedings*, volume 6478 of *Lecture Notes in Computer Science*, pages 50–59. Springer, 2010. doi:10.1007/978-3-642-17493-3_7.

7 Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 21–40. SIAM, 2019. doi:10.1137/1.9781611975482.2.

8 Paul Erdös and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, 1(1):85–90, 1960.

9 Oded Goldreich and Avishay Tal. On constant-depth canonical boolean circuits for computing multilinear functions. In *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*, pages 306–325. Springer, 2020. doi:10.1007/978-3-030-43662-9_17.

10 Oded Goldreich and Avi Wigderson. On the size of depth-three boolean circuits for computing multilinear functions. In *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*, pages 41–86. Springer, 2020. doi:10.1007/978-3-030-43662-9_6.

11 András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993. doi:10.1016/0022-0000(93)90001-D.

12 Johan Håstad. Almost optimal lower bounds for small depth circuits. In *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.

13 Johan Håstad, Stasys Jukna, and Pavel Pudlák. Top-down lower bounds for depth-three circuits. *Computational Complexity*, 5(2):99–112, 1995. doi:10.1007/BF01268140.

14 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

15 Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-24508-4.

16 Daniel M. Kane and Richard Ryan Williams. The orthogonal vectors conjecture for branching programs and formulas. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, volume 124 of *LIPIcs*, pages 48:1–48:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ITCS.2019.48.

17 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discret. Math.*, 3(2):255–265, 1990. doi:10.1137/0403021.

18 Igor Carboni Oliveira, Rahul Santhanam, and Srikanth Srinivasan. Parity helps to compute majority. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 23:1–23:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CCC.2019.23.

19 Alexander A. Razborov. On the method of approximations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 167–176. ACM, 1989. doi:10.1145/73007.73023.

20 Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *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.

21 Leslie G. Valiant. Exponential lower bounds for restricted monotone circuits. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 110–117. ACM, 1983. doi:10.1145/800061.808739.

22 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2-3):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.

23 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3447–3487. World Scientific, 2018.

24 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 645–654. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.67.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il