

1 Depth-3 Circuit Lower Bounds for k -OV

2 **Anonymous author**

3 Anonymous affiliation

4 **Anonymous author**

5 Anonymous affiliation

6 — Abstract —

7 The 2-Orthogonal Vectors (2-OV) problem is the following: given two tuples A and B of n Boolean
 8 vectors, each of dimension d , decide if there exist vectors $u \in A$, and $v \in B$, such that u and v
 9 are orthogonal. This problem, and its generalization k -OV defined analogously for k tuples, are
 10 central problems in the area of fine-grained complexity. One of the major conjectures in fine-grained
 11 complexity is that k -OV cannot be solved by a randomised algorithm in $n^{k-\epsilon} \text{poly}(d)$ time for any
 12 constant $\epsilon > 0$.

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

17 We show that for all $k \leq d$, any disjunction of t -CNFs computing k -OV requires size $\Omega((n/t)^k)$.
 18 In particular, when k is a constant, any disjunction of k -CNFs computing k -OV needs to use
 19 $\Omega(n^k)$ CNFs. This matches the brute-force construction, and for each fixed $k > 2$, this is the first
 20 unconditional $\Omega(n^k)$ lower bound against k -OV for a computation model that can compute it in size
 21 $O(n^k)$. Our results partially resolve a conjecture by Kane and Williams [16] (page 12, conjecture 10)
 22 about depth-3 AC^0 circuits computing 2-OV.

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

26 **2012 ACM Subject Classification** Theory of computation \rightarrow Circuit complexity; Theory of compu-
 27 tation \rightarrow Problems, reductions and completeness

28 **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 theorem¹ that there are languages in $\text{DTIME}(n^2)$ that are not in $\text{DTIME}(n^c)$ 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 [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 time² is the *2-Orthogonal Vectors problem*.

The 2-Orthogonal Vectors problem $2\text{-OV}_{n,d}$ 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. To define a generalization of this problem, we think of each vector from $\{0,1\}^d$ as a characteristic vector of a subset from $[d]$. Then a natural generalization of $2\text{-OV}_{n,d}$ is the problem $k\text{-OV}_{n,d}$ that takes as input k tuples $A_1, A_2, \dots, A_k \subseteq \{0,1\}^d$ of n vectors each, and the task is to decide if there exists vectors $a_1 \in A_1, a_2 \in A_2, \dots, a_k \in A_k$ such that $a_1 \cap a_2 \cap \dots \cap a_k = \emptyset$. The problems 2-OV and $k\text{-OV}$ have emerged as central problems in fine-grained complexity. An important hypothesis is that no deterministic, or randomized, algorithm computing $2\text{-OV}_{n,d}$ can run in time $O(n^{2-\epsilon} \text{poly}(d))$ 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 $2\text{-OV}_{n,d}$, and $k\text{-OV}_{n,d}$. 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 $k\text{-OV}$. The fact that better algorithms for so many problems would imply better algorithms for $k\text{-OV}$, is perhaps not surprising. Intuitively, the

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

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

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

71 k -OV function looks “canonical” in a certain sense, and has managed to hide itself inside
 72 several other problems that look quite different at the surface. These include seemingly
 73 unrelated problems such as Longest Common Subsequence [1], Edit Distance [2], Fréchet
 74 distance [4, 5], Regular Expressions Matching [3], to name a few. Their survey [23] is an
 75 excellent source for those looking for a thorough treatment of fine-grained complexity, and in
 76 particular, this line of research.

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

82 Note that if a uniform circuit family of bounded fan-in, and size $O(s(n, d))$ computes
 83 $k\text{-OV}_{n,d}$, then an algorithm that simply evaluates the circuit, computes $k\text{-OV}_{n,d}$ in time
 84 $\tilde{O}(s(n, d))$. So if the k -OV hypothesis is true, then we can expect any uniform circuit family
 85 computing $k\text{-OV}_{n,d}$ to have size $\Omega(n^k)$. This begs the question:

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

88 One class of Boolean circuits that has been extensively studied in terms of lower bounds
 89 is AC^0 (gates from $\{\wedge, \vee, \neg\}$, unbounded fan-in, $O(1)$ -depth). In fact we know exponential
 90 lower bounds against this class of circuits. So a good target would be to show that $k\text{-OV}_{n,d}$
 91 requires AC^0 circuits of size $\Omega(n^k \text{ poly}(d))$. We note that $k\text{-OV}_{n,d}$ can indeed be computed
 92 by depth-3 AC^0 circuits of size $n^k d$, as shown later in equation 2. Can we show matching
 93 lower bounds?

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

111 More recently, the status of depth-3 $\text{AC}^0[\oplus]$ circuits (gates computing xor are allowed in
 112 addition to the usual \wedge, \vee, \neg) got an update. The lower bound for computing majority using
 113 depth-3 $\text{AC}^0[\oplus]$ circuits was improved from $2^{\Omega(n^{1/4})}$ to $2^{\Omega(\sqrt{n})}$ by Oliveira, Santhanam and
 114 Srinivasan [18]. This closed the gap between upper and lower bounds up to a logarithmic
 115 factor in the exponent.

116 While techniques such as the switching lemma and the polynomial method work in
 117 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, Pudlák [13]. Our results in this paper can be seen as a non-trivial application of the techniques of Håstad, Jukna, Pudlák [13].

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

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

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

Hence, informally, their conjecture for $2\text{-OV}_{n,d}$, and by extension $k\text{-OV}_{n,d}$, 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 $k\text{-OV}$. 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 $k\text{-OV}_{n,d}$ with the gates on the bottom layer restricted to having small fan-in. Our main result is the following:

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

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

“Any disjunction of t -CNFs computing $k\text{-OV}_{n,d}$ 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 $k\text{-OV}_{n,d}$, 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 $k\text{-OV}_{n,d}$ as a disjunction of k -CNFs is $\Theta(n^k)$.

The proof technique used for Theorem 1 actually goes through for a more general class of depth-3 circuits where the bottom gates can have arbitrary fan-in as long as the number of negated literals among their inputs is at most t . We describe this in the next subsection. The more general theorem is the following. Let \mathcal{C}_t^- be the set of all unate functions (see Definition 7) that are negative unate on at most t variables.

► **Theorem 2.** *For all $k \leq d$, any $OR \circ AND \circ \mathcal{C}_t^-$ circuit computing $k\text{-OV}_{n,d}$ requires top fan-in $\Omega\left(\left(\frac{n}{t}\right)^k\right)$.*

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

159 becomes easy to compute by AC^0 circuits with high probability under random restrictions
160 [16] (section 3).

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

163 ► **Theorem 3.** *For all $\ell \leq d$, any $\text{AND} \circ \text{OR} \circ \text{AND}$ circuit computing $2\text{-OV}_{n,d}$ requires size*
164 *$s \in \Omega(\min\{2^\ell, (\frac{d}{n\ell})^n\})$. In particular, for $\ell = d/2n$ and $d \in \Omega(n^2)$, $s \in \Omega(2^n)$.*

165 Since $2\text{-OV}_{n,d}$ reduces to $k\text{-OV}_{n,d}$ by projections trivially, the above theorem holds for
166 $k\text{-OV}_{n,d}$ as well.

167 Techniques.

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

$$173 \quad k\text{-Int}_{n,d}(A_1, \dots, A_k) = \bigwedge_{i_1, \dots, i_k \in [n]} \bigvee_{j \in [d]} (a_{i_1}[j] \wedge \dots \wedge a_{i_k}[j]) \quad (3)$$

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

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

182 The technique of *finite limits* by [13] is used to show that C_i cannot be correct on many
183 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
184 y such that for any set of t input positions, it looks identical to *some string* in $C_i^{-1}(0) \cap \mathcal{N}$.
185 Such a string y is called a *t-limit* for the set $C_i^{-1}(0) \cap \mathcal{N}$. Then if the bottom gates in C_i
186 can each see only t bits of the input, the string y fools all of them into evaluating to 0
187 *simultaneously*, and hence C_i will output 0 on y . This is a contradiction since $y \in C^{-1}(1)$ by
188 construction, but $C_i(y) = 0$ implies $C(y) = 0$. It is not hard to see that if the t -limit string
189 y has the additional property that $y \geq x$ for all $x \in C_i^{-1}(0) \cap \mathcal{N}$, and each bottom gate in
190 C_i has at most t positive literals among its inputs, the same argument goes through. We
191 call such a y an *upper t-limit* to the set $C_i^{-1}(0) \cap \mathcal{N}$ (as opposed to the term ‘*lower t-limit*’
192 used in [13] for the case when $y \leq x$). We shall also use the term “bottom positive fan-in” to
193 indicate how many of the input literals are allowed to be positive for each bottom gate.

194 We remark here that that all t -limit strings that we construct in this paper are also
195 upper t -limit strings. Hence all our lower bounds for $k\text{-Int}_{n,d}$ go through for the circuit
196 class $\text{AND} \circ \text{OR} \circ \mathcal{C}_t^+$ where \mathcal{C}_t^+ is the set of all unate functions that are positive unate on
197 at most t variables. Informally, this means that the bottom gates can compute any unate
198 functions, have unbounded fan-in, but at most t of the inputs can be positive literals. (The
199 dual statement for $k\text{-OV}_{n,d}$ is Theorem 2 stated in the previous section.) As an example,
200 lower bounds using this technique will also work against depth-3 circuits where the top and
201 middle layers are AND and OR respectively, and the bottom layer consists of homogeneous
202 linear threshold functions, each of which is defined by a vector of weights that has at most t
203 positive weights.

204 An important observation about the technique described above is that it is impervious
 205 to the fan-in of the middle OR gates. So we could use a suitable DNF for each bottom
 206 gate and convert an $\text{AND} \circ \text{OR} \circ \mathcal{C}_t^+$ circuit to an $\text{AND} \circ \text{OR} \circ \text{AND}$ circuit with bottom
 207 positive fan-in at most t and a possibly larger middle fan-in. Since the technique gives lower
 208 bounds to top fan-in regardless of middle fan-in, all lower bounds that we can derive against
 209 $\text{AND} \circ \text{OR} \circ \text{AND}$ circuits with bottom positive fan-in t using this technique, transfer to
 210 $\text{AND} \circ \text{OR} \circ \mathcal{C}_t^+$ without any change. Hence throughout this paper, we focus our attention to
 211 $\text{AND} \circ \text{OR} \circ \text{AND}$ circuits.

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

220 We show in Section 5 a general construction for $\text{k-Int}_{n,d}$ that achieves a trade-off between
 221 top fan-in and bottom fan-in. This shows that in general, for circuits with bottom fan-in t
 222 computing $\text{k-Int}_{n,d}$, our lower bound for the top fan-in is at least a factor of t^{k-1}/k away
 223 from the corresponding upper bound.

224 *Secondary result.* The exponential lower bound of [13] for $\text{OR} \circ \text{AND} \circ \text{OR}$ circuits
 225 computing the iterated intersection function $S_{n,d}$ for $d \in \sqrt{n}$ is of particular interest to us.
 226 The function $S_{n,d}$ bears a close resemblance to $2\text{-Int}_{n,d}$. While $S_{n,d}$ is the *iterated* intersection,
 227 $2\text{-Int}_{n,d}$ can be seen as “all-pairs” intersection.

228 We show a reduction (via projections) from $S_{n,d/n}$ to $2\text{-Int}_{n,d}$. The blow-up in the
 229 dimension of vectors is rather large, and we can conclude non-trivial lower bounds only for
 230 $d \in \omega(n)$.

231 2 Preliminaries

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

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

235 $\text{k-OV}_{n,d}(A_1, A_2, \dots, A_k) = 1 \iff \exists a_1 \in A_1, \exists a_2 \in A_2, \dots, \exists a_k \in A_k$, such that

$$236 \quad a_1 \cap a_2 \cap \dots \cap a_k = \emptyset$$

237
 238 For notational convenience, we work with the negation of $\text{k-OV}_{n,d}$ throughout the paper.
 239 We use $\text{k-Int}_{n,d}$ to denote the negation of $\text{k-OV}_{n,d}$, and is defined as follows:

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

241 $\text{k-Int}_{n,d}(A_1, A_2, \dots, A_k) = 1 \iff \forall a_1 \in A_1, \forall a_2 \in A_2, \dots, \forall a_k \in A_k$, we have

$$242 \quad a_1 \cap a_2 \cap \dots \cap a_k \neq \emptyset$$

243
 244 An input to the function $\text{k-Int}_{n,d}$ has nk vectors, each of dimension d . Hence $nk d$ many
 245 input bits in total.

246 For any $x, y \in \{0,1\}^d$, we write $x \leq y$ if $\forall i, x_i \leq y_i$. Similarly, we write $x \oplus y$ to denote
 247 the string obtained by a point-wise xor between x and y .

248 ▶ **Definition 6** (Monotone function). We say that a Boolean function f is monotone if
 249 $\forall x, y \in \{0, 1\}^d$ such that $x \leq y$, we have $f(x) \leq f(y)$.

250 The notion of monotone can be generalized to the notion of being *unate*:

251 ▶ **Definition 7** (Unate function). A Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is unate if there
 252 exists a monotone Boolean function $g : \{0, 1\}^n \rightarrow \{0, 1\}$ and a string $s \in \{0, 1\}^n$ such that
 253 for all inputs x , we have $f(x) = g(x \oplus s)$.

254 Further, a unate function is *positive unate* (*negative unate*) on a variable x_i if $s_i = 0$
 255 ($s_i = 1$ respectively).

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

257 ▶ **Definition 8.** (*Maximal 0-input*) Let f be a monotone Boolean function. An input x is a
 258 maximal 0-input for f if $f(x) = 0$ and for all strings y such that $x < y$, $f(y) = 1$.

259 Throughout this article, we will use the term “*maxterm*” and “maximal 0-inputs” inter-
 260 changeably. This deviates from the standard definition of *maxterm*, but is very convenient in
 261 our context.

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

264 ▶ **Definition 9** (*t-limit*). A vector $y \in \{0, 1\}^m$ is said to be a *t-limit* for a set $B \subseteq \{0, 1\}^m$
 265 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,
 266 $y \in \{0, 1\}^m$ is said to be an *upper t-limit* if $y \geq x$.

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

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

275 **3 AND ◦ OR ◦ AND circuits**

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

279 **3.1 Maxterms of $k\text{-Int}_{n,d}$**

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

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

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

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

289 To see that N is indeed a maxterm of $\mathbf{k}\text{-Int}_{n,d}$, observe that since \mathcal{P} is a partition of $[d]$, for
 290 every position $\ell \in [d]$, there is a *unique* $i \in [k]$ such that $\ell \in P_i$. Therefore, by construction
 291 of N , $a_i^{n_i}[\ell] = 0$. So for every position ℓ , there is some vector among $a_1^{n_1}, \dots, a_k^{n_k}$ that is
 292 0 in position ℓ , and hence $a_1^{n_1} \cap \dots \cap a_k^{n_k} = \emptyset$. Moreover, due to i being unique for each
 293 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
 294 vectors intersecting at ℓ . Combining this with the fact that every vector in N other than
 295 $a_1^{n_1}, \dots, a_k^{n_k}$ is the all-1s vector, we conclude that N is indeed a maximal 0-input.

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

299 **► Definition 10.** (*Permutation-maxterms*) Fix an ordered partition $\mathcal{P} = (P_1, \dots, P_k)$ of
 300 $[d]$ into k non-empty parts. A permutation-maxterm with respect to \mathcal{P} is any maxterm
 301 N constructed as above that has $\text{sup}(N) = ((n_1, \dots, n_k), \mathcal{P}_\sigma)$ for some $n_1, \dots, n_k \in [n]$ and
 302 $\sigma \in \mathcal{S}_k$.

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

306 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
 307 many k -tuples (n_1, \dots, n_k) and $k!$ many permutations in \mathcal{S}_k .

308 **► Remark 11.** The proofs in this paper do not depend on the exact permutation chosen.
 309 Any arbitrary ordered permutation of $[d]$ into k non-empty parts will work. For a further
 310 simplification, one could assume $k = d$, and fix the permutation $\mathcal{P} = (P_1, \dots, P_k)$ to be
 311 $P_i = \{i\}$ for all $i \in [d]$.

312 3.2 Support Graph

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

316 Fix $k \geq 2$ and $d \geq k$, and any ordered partition \mathcal{P} of $[d]$ into k non-empty parts. For any
 317 subset $S \subseteq \mathcal{N}_{\mathcal{P}}^{n,k,d}$ of permutation-maxterms of $\mathbf{k}\text{-Int}_{n,d}(A_1, \dots, A_k)$, we define the *support*
 318 *graph* of S as a k -partite hypergraph $\mathcal{G}_S = (V_1 \cup \dots \cup V_k, E)$ as follows. As usual we will
 319 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
 320 vertices in V_i denoted $v_i^{j,1}, \dots, v_i^{j,k}$. So for all $i \in [k]$, we have $|V_i| = nk$ and hence the graph
 321 \mathcal{G}_S is on nk^2 many vertices.

322 We define the set E of hyperedges as follows:

$$323 \quad \left(v_1^{n_1, b_1}, \dots, v_k^{n_k, b_k} \right) \in E \iff \exists \text{ maxterm } N \in S \text{ such that}$$

$$324 \quad \text{sup}(N) = ((n_1, \dots, n_k), \mathcal{P}_\sigma) \text{ and } b_i = \sigma(i) \forall i \in [k]$$

326 **► Remark 12.** Note that the set of maxterms $S \subseteq \mathcal{N}_{\mathcal{P}}$ and the set of hyperedges in \mathcal{G}_S are in
 327 bijection. More precisely, a maxterm N with $\text{sup}(N) = ((n_1, \dots, n_k), \mathcal{P}_\sigma)$ corresponds to the
 328 hyperedge $\left(v_1^{n_1, \sigma(1)}, \dots, v_k^{n_k, \sigma(k)} \right)$ and vice-versa.

329 ► **Definition 13** (Co-disjoint). We call two vectors $u \in \{0, 1\}^d$ and $v \in \{0, 1\}^d$ as co-disjoint
 330 if and only if $\bar{u} \cap \bar{v} = \emptyset$. i.e., the set of positions where u is 0, and the set where v is 0 are
 331 disjoint.

332 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
 333 say A and B are co-disjoint if for all $i \in [n]$, a_i and b_i are co-disjoint.

334 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
 335 co-disjoint if and only if for all $i \in [k]$, M_i and N_i are co-disjoint.

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

339 ► **Lemma 14.** 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
 340 $M = (M_1, \dots, M_k)$ and $N = (N_1, \dots, N_k)$ be two maxterms from S and let E_M , and E_N
 341 respectively, denote their corresponding hyperedges in \mathcal{G}_S . Then for each $i \in [k]$, we have the
 342 following two properties:

- 343 1. If E_M and E_N share a vertex in V_i , then $M_i = N_i$.
- 344 2. If E_M and E_N contain different vertices from V_i , then M_i and N_i are co-disjoint.

345 **Proof.** Let $\text{sup}(M) = (a_1, \dots, a_k, \mathcal{P}_\sigma)$ and $\text{sup}(N) = (b_1, \dots, b_k, \mathcal{P}_\pi)$.

346 Proof of (1): If E_M and E_N share a vertex in V_i for some $i \in [k]$, then $v_i^{a_i, \sigma(i)} = v_i^{b_i, \pi(i)}$
 347 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
 348 construction of the maxterms M and N , all vectors in M_i other than m_i^ℓ are all 1s, and
 349 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
 350 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
 351 identical.

352 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
 353 either $a_i \neq b_i$ or $\sigma(i) \neq \pi(i)$ (or both). The claim holds in both cases:

- 354 ■ If $a_i \neq b_i$, then recall that by construction, the only vector that has 0s in M_i is the vector
 355 $m_i^{a_i}$. Every other vector in M_i , and in particular $m_i^{b_i}$ is the all 1s vector by construction.
 356 So the tuples of vectors M_i and N_i cannot both be 0 in any vector in any position.
- 357 ■ Else $a_i = b_i$ and $\sigma(i) \neq \pi(i)$. By our construction of maxterms, the 0s in the vectors $m_i^{a_i}$
 358 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,
 359 and $\sigma(i) \neq \pi(i)$, $P_{\sigma(i)} \cap P_{\pi(i)} = \emptyset$. Therefore there cannot be an index where both $m_i^{a_i}$
 360 and $n_i^{b_i}$ are both 0.

361 ◀

362 The following lemma follows directly from Lemma 14:

363 ► **Lemma 15.** Let $S \subseteq \mathcal{N}_{\mathcal{P}}^{n,k,d}$ be a set of maxterms such that all hyperedges in \mathcal{G}_S are
 364 pairwise vertex-disjoint. Then the maxterms in S are pairwise co-disjoint. (i.e., for all
 365 positions $\ell \in [nk]$, there is at most one maxterm in S that has 0 in the ℓ 'th position.)

366 **Proof.** Let $M, N \in S$ be any two maxterms, and let the vertex set of \mathcal{G}_S be $V = V_1 \cup \dots \cup V_k$.
 367 The hyperedges E_M and E_N , corresponding to M , and N respectively, are vertex-disjoint
 368 from the premise. So for each $i \in [k]$, E_M and E_N contain different vertices from V_i . Applying
 369 Lemma 14 to \mathcal{G}_S , we obtain that M_i and N_i are co-disjoint for all $i \in [k]$. Hence there is no
 370 position where both M and N are 0 by definition of co-disjoint. ◀

371 **3.3 Warm-up: 2-Int _{n,d}**

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

374 ► **Theorem 16.** *For all $d > 1$, any AND ◦ OR ◦ AND circuit with bottom fan-in t computing
 375 2-Int _{n,d} requires top fan-in at least $2n^2/t^2$.*

376 **Proof.** Let $C = C_1 \wedge C_2 \wedge \dots \wedge C_s$ be an AND ◦ OR ◦ AND _{t} circuit with bottom fan-in t
 377 computing 2-Int _{n,d} . Let $\mathcal{P} = (P_1, P_2)$ be any ordered partition of $[d]$ into two non-empty parts.
 378 Consider the permutation-maxterms $\mathcal{N} = \mathcal{N}_{\mathcal{P}}^{n,2,d}$ of 2-Int _{n,d} as described in definition 10.
 379 Since \mathcal{N} is a subset of the 0-inputs of 2-Int _{n,d} , the circuit C outputs 0 on every input in \mathcal{N} .
 380 By an averaging argument, there exists $i \in [s]$ such that C_i correctly outputs 0 on at least
 381 $1/s$ fraction of inputs in \mathcal{N} . We will show that $|C_i^{-1}(0) \cap \mathcal{N}| \leq t^2$. Then the theorem follows
 382 as:

$$383 \quad \frac{2n^2}{s} = \frac{1}{s} |\mathcal{N}| \leq |C_i^{-1}(0) \cap \mathcal{N}| \leq t^2.$$

385 In the following, we will show that $\forall S \subseteq \mathcal{N}$ with $|S| > t^2$, there is a t -limit $y \in C^{-1}(1)$
 386 for S . This will imply that $|C_i^{-1}(0) \cap \mathcal{N}| \leq t^2$. To see why, let $C_i = g_1 \vee g_2 \dots \vee g_\ell$ with each
 387 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
 388 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
 389 $|T| = t$, there exists $x \in S$ such that $x|_T = y|_T$. Now each of the gates g_j is a function of at
 390 most t variables, and we know that for all inputs $x \in S$, we have $g_j(x) = 0$ for all $j \in [\ell]$.
 391 Since y looks identical to some string in S when restricted to these t positions, all the g_j will
 392 output 0 on y too. This forces $C_i(y) = 0$ leading to a contradiction since $y \in C^{-1}(1)$.

393 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
 394 $k = 2$, \mathcal{G}_S is a bipartite graph with simple edges rather than hyperedges, and every maxterm
 395 in S corresponds to an edge in \mathcal{G}_S and vice versa. We claim at least one of the following is
 396 true for \mathcal{G}_S :

- 397 (i) There exists a matching of size $t + 1$ in \mathcal{G}_S .
- 398 (ii) There exists a vertex of degree at least $t + 1$ in \mathcal{G}_S .

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

404 ■ Case (i): Consider the set S' of maxterms corresponding to the edges in a maximum
 405 matching of \mathcal{G}_S . Then S' is a set of at least $t + 1$ pairwise co-disjoint maxterms. Then
 406 $y \triangleq \vec{1}$ is a t -limit for S' . To see why, consider any set of t positions. By Lemma 15, at
 407 each of these positions, at most one of maxterms can be 0. Since there are $t + 1$ such
 408 maxterms and only t positions, there must be a maxterm where the value at all the given
 409 positions is 1, thus looking identical to y .

410 ■ Case (ii): Let the vertex set of \mathcal{G}_S be $V = V_1 \cup V_2$. Without loss of generality, let the
 411 vertex v with $\deg(v) > t$ be in V_1 . Let E be the edges that have v as one endpoint, and
 412 let $M_E \subseteq S$ be the maxterms corresponding to the edges in E . Then by property (1) of
 413 Lemma 14, the first tuple of vectors in all these maxterms is the same. Let A_1 be the
 414 first tuple of vectors. We construct the input $y = (Y_1, Y_2)$ as follows: set $Y_1 \leftarrow A_1$, and
 415 set $Y_2 \leftarrow \vec{1}$.

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

418 To see that y is a t -limit, take any subset of indices $T \subseteq [2nd]$ with $|T| = t$. We will
419 show that one of the maxterms in M_E looks identical to y in these t positions. For every
420 position from $[nd]$ (the 1st tuple of vectors), *every* maxterm in M_E is identical to y since
421 $Y_1 = A_1$. So assume that all indices in T are from the range $\{nd + 1, \dots, 2nd\}$. By
422 construction, y is all-1s in this range of indices. Since edges in E have distinct endpoints
423 in V_2 , property (2) of Lemma 14 tells us that the second tuple of vectors in the maxterms
424 in T are pairwise co-disjoint. This is similar to case (i): we have $|M_E| \geq t + 1$ many
425 maxterms such that for any position in T , at most one of them is 0, and there are only t
426 positions in T . So by the pigeon-hole principle, there must be a maxterm in M_E that has
427 1 in all positions from T , thus looking identical to y in these positions.

428

429 Since $2\text{-OV}_{n,d}$ is the negation of $2\text{-Int}_{n,d}$, the following is an immediate corollary of
430 Theorem 16.

431 ► **Corollary 17.** *For all $d > 1$, any $\text{OR} \circ \text{AND} \circ \text{OR}$ circuit with bottom fan-in t computing
432 $2\text{-OV}_{n,d}$ requires top fan-in at least $2n^2/t^2$.*

433 ► **Remark 18.** It is easy to see that the t -limit string y constructed in the proof of Theorem
434 16 is in fact an *upper* t -limit. Therefore the lower bound shown for $2\text{-Int}_{n,d}$ works against a
435 slightly more general class of circuits — $\text{AND} \circ \text{OR} \circ \text{AND}$ circuits that have each bottom
436 AND -gate seeing at most t positive literals. Analogously the lower bound for $2\text{-OV}_{n,d}$ works
437 against $\text{OR} \circ \text{AND} \circ \text{OR}$ circuits where each bottom gate has at most t negated inputs.

438 3.4 General case: $k\text{-Int}_{n,d}$

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

440 ► **Lemma 19.** *Let G be a k -partite hypergraph with m many hyperedges. Then for all $t > 0$
441 at least one of the following holds:*

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

444 **Proof.** Let G be a k -partite hypergraph with m hyperedges. Let S be a largest set of
445 vertex-disjoint hyperedges in G . If $|S| > t$, then the lemma is true. Suppose $|S| \leq t$. Let V_S
446 be the set of vertices participating in the hyperedges in S . Since each hyperedge contains
447 exactly k many vertices, $|V_S| \leq kt$. Also, since S is a largest such set, each of the remaining
448 hyperedges must contain at least one vertex from V_S . Therefore, by an averaging argument,
449 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
450 hyperedge in S . Therefore, we have:

$$451 \deg(u) \geq \frac{m - |S|}{|V_S|} + 1 \geq \frac{m - t}{kt} + 1 = \frac{m}{kt} - \frac{1}{k} + 1 > \lfloor \frac{m}{kt} \rfloor$$

453

454 We use Lemma 19 to show that if we start with enough hyperedges, then there is a subset of
455 them such that in each part, either all of them coincide, or they are all distinct.

456 ► **Lemma 20.** *Let $k \geq 2$, and let $G = (V_1 \cup \dots \cup V_k, E)$ be a k -partite hypergraph with
 457 $|E| > \frac{k!t^k}{2}$. Then there exists $S \subseteq E$ with $|S| > t$ such that for each $i \in [k]$, exactly one of
 458 the following holds:*

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

461 **Proof.** Induction on k . Base case $k = 2$ is a consequence of König's theorem: Since $k = 2$,
 462 G is just a bipartite graph. If there is a matching in G of size more than t , then let S be the
 463 edges in such a matching. Clearly the edges in S are vertex-disjoint and statement (2) holds.
 464 Else the maximum matching has size $\leq t$. Then König's theorem implies that the minimum
 465 vertex cover has size at most t . By an averaging argument, there must exist a vertex u such
 466 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
 467 loss of generality, let $u \in V_1$. Then all edges in S must have distinct vertices in V_2 . Therefore
 468 in V_1 , they all coincide, and in V_2 they are all distinct.

469 Case $k > 2$: Apply Lemma 19 to G . If (i) holds, then we have a set S of more than t
 470 vertex-disjoint hyperedges. This means for all $i \in [k]$, statement (2) holds and we are done.

471 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
 472 S be the set of all hyperedges that contain vertex u . Then $|S| = \deg(u)$. Let $z \in [k]$ be such
 473 that $u \in V_z$.

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

$$476 \quad V' \triangleq V_1 \cup \dots \cup V_{z-1} \cup V_{z+1}, \dots \cup V_k$$

$$477 \quad E' \triangleq \{(v_1, \dots, v_{z-1}, v_{z+1}, \dots, v_k) \mid (v_1, \dots, v_{z-1}, u, v_{z+1}, v_k) \in S\}$$

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

480 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
 481 share the vertex u in V_z . So there must exist $j \neq z$ such that e_1 and e_2 use different vertices
 482 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
 483 and only if e_1 and e_2 share the same vertex in V_i .

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

490 ► **Remark 21.** The statement of Lemma 19 can be seen as a sunflower lemma. Take any
 491 vertex u in the graph G that participates in at least one hyperedge from S . Then exactly
 492 one of the following holds: (i) The vertex u participates in exactly one hyperedge in S , or
 493 (ii) The vertex u participates in all hyperedges in S . The standard sunflower lemma would
 494 require more than $k!t^k$ hyperedges, while our statement needs half of that.

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

496 ► **Lemma 22.** *Let $\mathcal{M} \subseteq \mathcal{N}_{\mathcal{P}}^{n,k,d}$ be any set of permutation-maxterms of k -Int $_{n,d}$ for any $k \geq 2$
 497 and $d \geq k$. If $|\mathcal{M}| > \frac{k!t^k}{2}$, then there is a string $y \in k$ -Int $_{n,d}^{-1}(1)$ that is an upper t -limit for
 498 \mathcal{M} .*

499 **Proof.** Let $G_{\mathcal{M}} = (V, E)$ be the k -partite support graph of \mathcal{M} (defined in section 3.2), and
 500 let $V = V_1 \cup \dots \cup V_k$. By Lemma 20, there exists a set of hyperedges $S \subseteq E$ with $|S| \geq t + 1$

501 such that for each $i \in [k]$, either all edges in S share the same vertex in V_i , or no two edges
 502 share a vertex of V_i . Let M_S be the set of maxterms corresponding to S .

503 Let $B \subseteq [k]$ be the set of all indices $i \in [k]$ such that all edges in S share the same vertex
 504 in V_i . Then \bar{B} contains indices of parts where the edges in S use distinct vertices. (Observe
 505 that \bar{B} is non-empty because otherwise all maxterms would share all vertices, and hence
 506 would be one and the same. But we know that $|S| \geq t + 1 > 1$, so this cannot happen.) By
 507 property (1) of Lemma 14, this implies that for each $i \in B$, the i 'th tuple of vectors in the
 508 maxterms in M_S are identical. For each $i \in B$, denote the i 'th tuple of vectors in all these
 509 maxterms as A_i .

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

511 $\forall i \in B$, set $Y_i \leftarrow A_i$

512 $\forall j \in \bar{B}$, set $Y_j \leftarrow \bar{1}$
 513

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

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

519 **y is a t -limit:**

520 Let $T \subseteq [nkd]$ with $|T| = t$ be a set of any t positions. For all $i \in B$, the string y is identical
 521 to *every* maxterm in M_S . So assume that T only has positions that fall into tuples indexed
 522 by \bar{B} . By property (2) of Lemma 14, the maxterms in M_S are pairwise co-disjoint on all
 523 such positions. i.e., for any position $\ell \in T$, at most one maxterm in M_S can be 0. So we
 524 have t positions, and $|M_S| = |S| \geq t + 1$ maxterms. By pigeon-hole principle, there exists a
 525 maxterm in M_S that is 1 on all these t positions, thus looking identical to y .

526 Since y is point-wise greater or equal to every maxterm in S , we conclude that indeed y
 527 is an upper t -limit to \mathcal{M} . \blacktriangleleft

528 **► Lemma 23.** *Let C be any $\text{OR} \circ \text{AND}$ circuit with bottom positive fan-in t computing a*
 529 *function f on n variables. Let y be any string that is an upper t -limit to $f^{-1}(0)$. Then*
 530 *$C(y) = 0$.*

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

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

536 The gate g sees negative literals of all variables from Q . Since y is an *upper* t -limit, we
 537 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
 538 $g(y) \leq g(x^{(T)}) = 0$ as $x^{(T)} \in g^{-1}(0)$. \blacktriangleleft

539 **► Theorem 24.** *For all k, d such that $k \leq d$, any $\text{AND} \circ \text{OR} \circ \text{AND}$ circuit with bottom*
 540 *positive fan-in t computing $k\text{-Int}_{n,d}$ requires top fan-in $\Omega\left(\left(\frac{n}{t}\right)^k\right)$.*

541 **Proof.** Let $C = C_1 \wedge \dots \wedge C_s$ be an $\text{AND} \circ \text{OR} \circ \text{AND}_t$ circuit with bottom positive fan-in
 542 t , computing $k\text{-Int}_{n,d}$. Consider the set $\mathcal{N} = \mathcal{N}_{\mathcal{P}}^{n,k,d}$ of all permutation-maxterms of $k\text{-Int}_{n,d}$
 543 with respect to any ordered permutation \mathcal{P} of $[d]$ into k non-empty parts (see Definition 10,

544 and Remark 11). Since C outputs 0 on all inputs from \mathcal{N} , there must be some $\text{OR} \circ \text{AND}_t$
 545 subcircuit C_i that correctly outputs 0 on at least $1/s$ fraction of inputs in \mathcal{N} . We will show
 546 that $|C_i^{-1}(0) \cap \mathcal{N}| \leq k! t^k / 2$, and the theorem follows since:

$$547 \quad \frac{k! n^k}{s} = \frac{1}{s} |\mathcal{N}| \leq |C_i^{-1}(0) \cap \mathcal{N}| \leq \frac{k! t^k}{2}$$

549 Let $\mathcal{M} = C_i^{-1}(0) \cap \mathcal{N}$. Suppose, for the sake of contradiction, $|\mathcal{M}| > k! t^k / 2$. Since
 550 $\mathcal{M} \subseteq \mathcal{N}$, we apply Lemma 22 to conclude that there exists a string $y \in \text{k-Int}_{n,d}^{-1}(1)$ that is
 551 an upper t -limit y for \mathcal{M} . Then by Lemma 23, it must be that $C(y) = 0$. But this is a
 552 contradiction since $y \in \text{k-Int}_{n,d}^{-1}(1)$. \blacktriangleleft

553 Since $\text{k-OV}_{n,d}$ is the negation of $\text{k-Int}_{n,d}$, the following is an immediate corollary of Theorem
 554 24.

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

557 **4** OR \circ AND \circ OR circuits

558 In this section, we show that any $\text{OR} \circ \text{AND} \circ \text{OR}$ circuit requires exponential size to compute
 559 $2\text{-Int}_{n,d}$ for any $d \in \Omega(n^2)$. This result is a consequence of a known lower bound for the
 560 iterated intersection function defined as follows:

561 **► Definition 25** (Iterated Intersection). *Let $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_n)$ be*
 562 *tuples of vectors from $\{0, 1\}^d$,*

$$563 \quad S_{n,d}(A, B) = 1 \iff \forall i \in [n] \text{ we have } a_i \cap b_i \neq \emptyset$$

564 Observe that $S_{n,d}(A, B)$ differs from $2\text{-Int}_{n,d}(A, B)$ in that the intersection between two
 565 vectors a_i and b_j when $i \neq j$ does not affect the value of $S_{n,d}$ at all. Recall the definition of
 566 $2\text{-Int}_{n,d}(A, B)$:

$$567 \quad 2\text{-Int}_{n,d}(A, B) = 1 \iff \forall i, j \in [n] \text{ we have } a_i \cap b_j \neq \emptyset$$

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

$$570 \quad S_{n,d}(A, B) = \bigwedge_{i=1}^n \bigvee_{j=1}^d a_i[j] \wedge b_i[j].$$

572 The result by Håstad, Jukna, Pudlák in [13] shows the following lower bound for computing
 573 $S_{n,d}$ by $\text{OR} \circ \text{AND} \circ \text{OR}$ circuits:

574 **► Proposition 26** ([13]). *For all $\ell \leq nd$, any $\text{OR} \circ \text{AND} \circ \text{OR}$ circuit computing $S_{n,d}$ requires*
 575 *size $\min\{2^\ell, (d/\ell)^n\}$.*

576 In particular, Proposition 26 shows that $S_{\sqrt{n}, \sqrt{n}}$ requires $2^{\Omega(\sqrt{n})}$ size $\text{OR} \circ \text{AND} \circ \text{OR}$
 577 circuits. This can be used to show lower bounds for $2\text{-Int}_{n,d}$:

578 **► Theorem 27.** *Let C be an $\text{OR} \circ \text{AND} \circ \text{OR}$ circuit computing $2\text{-Int}_{n,d}$. Then for all $\ell \leq d$,*
 579 *size of C is at least $\min\{2^\ell, \left(\frac{d}{n\ell}\right)^n\}$.*

580 **Proof.** We show this by reducing $S_{n, \lfloor d/n \rfloor}$ to $2\text{-Int}_{n,d}$ via projections. Let $d' = \lfloor d/n \rfloor$. Take
 581 any instance $A = (a_1, \dots, a_n)$ and $B = (b_1, \dots, b_n)$ with $a_i, b_i \in \{0, 1\}^{d'}$ of $S_{n,d'}$. We
 582 create two sets of d -dimensional vectors $A' = (a'_1, \dots, a'_n)$ and $B' = (b'_1, \dots, b'_n)$ that serve
 583 as an instance of $2\text{-Int}_{n,d}$ as follows — for all $i \in [n]$, define $a'_i = 1^{(i-1)d'} a_i 1^{(n-i)d'}$ and
 584 $b'_i = 0^{(i-1)d'} b_i 0^{(n-i)d'}$. Note that the dimension of each a_i and b_i is $nd' \leq d$.

585 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
 586 0-instance of $S_{n,d'}$, then (A', B') is a 0-instance of $2\text{-Int}_{n,d}$.

587 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,
 588 observe that if $b_j \neq \vec{0}$, then there is some position $p \in [(j-1)d+1, jd]$ such that $b'_j[p] = 1$.
 589 But by construction, the vector a'_i is 1 everywhere outside the interval $[(i-1)d+1, id]$. Since
 590 $i \neq j$, the vector a'_i must be 1 at position p .

591 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
 592 vectors. Thus for all $i, j \in [n]$, $a'_i \cap b'_j \neq \emptyset$.

593 The above reduction shows that C can be used to compute $S_{n, \lfloor d/n \rfloor}$. Applying Proposition
 594 26 to C tells us that C must have size at least $\min\{2^\ell, \left(\frac{d}{n\ell}\right)^n\}$ for all $\ell \leq d$. ◀

595 Our reduction in proof of Theorem 27 inflates the dimension of vectors by a factor
 596 of n making the obtained bound trivial when $d \in O(n)$. However, we can still conclude
 597 an exponential lower bound by substituting $\ell = d/2n$ that gives us a lower bound of
 598 $\min\{2^{d/2n}, 2^n\} \in 2^{\Omega(n)}$ when $d \in \Omega(n^2)$.

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

600 ▶ **Theorem 3.** For all $\ell \leq d$, any $\text{AND} \circ \text{OR} \circ \text{AND}$ circuit computing $2\text{-OV}_{n,d}$ requires size
 601 $s \in \Omega(\min\{2^\ell, \left(\frac{d}{n\ell}\right)^n\})$. In particular, for $\ell = d/2n$ and $d \in \Omega(n^2)$, $s \in \Omega(2^n)$.

602 5 A General Upper Bound

603 In this section, we describe a more general construction of a depth-3 circuit to compute
 604 $k\text{-Int}_{n,d}$ that allows a trade-off between the top fan-in and bottom fan-in. We recall the
 605 construction given by equation 3 here:

$$606 \quad k\text{-Int}_{n,d}(A_1, \dots, A_k) = \bigwedge_{i_1, \dots, i_k \in [n]} \bigvee_{j \in [d]} (a_{i_1}[j] \wedge \dots \wedge a_{i_k}[j]) \quad (3)$$

608 We now show that $k\text{-Int}_{n,d}$ can be computed by a monotone depth-3 $\text{AND} \circ \text{OR} \circ \text{AND}$
 609 circuit with top fan-in $\lceil \frac{n^k}{t} \rceil$ and bottom fan-in at most kt for any integer $1 \leq t \leq n^k$.

610 Let C be the circuit described in equation 3. Observe that each $\text{OR} \circ \text{AND}$ subcircuit
 611 of C is checking whether a particular choice $a_{i_1} \in A_1, a_{i_2} \in A_2, \dots, a_{i_k} \in A_k$ of vectors are
 612 intersecting or not. Since there are n^k many such choices, the top fan-in is n^k . Checking if a
 613 particular choice of k vectors intersects at some fixed coordinate uses an AND of fan-in k ,
 614 and hence the bottom fan-in is k .

615 We can generalise this to a circuit where each $\text{OR} \circ \text{AND}$ subcircuit checks whether t
 616 many such choices of vectors intersect. Each choice can be written as a k -tuple of vectors
 617 $(a_{i_1}, \dots, a_{i_k})$. For convenience, let's assume that t divides n^k . Let $T = \{T_1, T_2, \dots, T_{n^k/t}\}$ be
 618 a partition of the set of n^k possible k -tuples of vectors into n^k/t parts with each T_l containing
 619 exactly t many k -tuples. For the vectors in any particular k -tuple in T_l to have non-empty
 620 intersection, there must exist a position $i \in [d]$ where all the k vectors in the k -tuple are 1.
 621 Hence to check if each of the k -tuples of vectors in T_l have non-zero intersection, it suffices to

622 check if there exist t positions $i_1, i_2, \dots, i_t \in [d]$ such that the j 'th k -tuple of vectors intersect
 623 in i_j .

624 Let $A_l^j[i]$ be the AND of the bits in the i^{th} position of the vectors in the j^{th} tuple in
 625 T_l . This is an AND gate with fan-in k because there are k many vectors in each tuple. We
 626 construct the following circuit where the ℓ 'th OR \circ AND subcircuit checks if each k -tuple of
 627 vectors in T_ℓ have non-zero intersection:

$$628 \quad G_t = \bigwedge_{l \in \{1, \dots, \frac{n^k}{t}\}} \bigvee_{i_1, i_2, \dots, i_t \in [d]} (A_l^1[i_1] \wedge A_l^2[i_2] \wedge \dots \wedge A_l^t[i_t])$$

629 Observe that G_t has top fan-in as n^k/t , middle fan-in as d^t , and bottom fan-in kt as desired.

630 Acknowledgements

631 The authors would like to thank the anonymous referees of a previous version for their
 632 valuable comments that helped improve the presentation of this paper in several respects.

633 — References —

- 634 1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for
 635 LCS and other sequence similarity measures. In *IEEE 56th Annual Symposium on Foundations
 636 of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 59–78.
 637 IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.14.
- 638 2 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic
 639 time (unless SETH is false). In *Proceedings of the Forty-Seventh Annual ACM on Symposium
 640 on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 51–58.
 641 ACM, 2015. doi:10.1145/2746539.2746612.
- 642 3 Arturs Backurs and Piotr Indyk. Which regular expression patterns are hard to match? In
 643 *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October
 644 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 457–466. IEEE Computer
 645 Society, 2016. doi:10.1109/FOCS.2016.56.
- 646 4 Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly sub-
 647 quadratic algorithms unless SETH fails. In *55th IEEE Annual Symposium on Foundations of
 648 Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 661–670.
 649 IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.76.
- 650 5 Karl Bringmann and Wolfgang Mulzer. Approximability of the discrete fréchet distance. *J.
 651 Comput. Geom.*, 7(2):46–76, 2016. doi:10.20382/jocg.v7i2a4.
- 652 6 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. On the exact complexity of evalu-
 653 ating quantified k -cnf. In *Parameterized and Exact Computation - 5th International Symposium,
 654 IPEC 2010, Chennai, India, December 13-15, 2010. Proceedings*, volume 6478 of *Lecture Notes
 655 in Computer Science*, pages 50–59. Springer, 2010. doi:10.1007/978-3-642-17493-3_7.
- 656 7 Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. In *Proceedings
 657 of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San
 658 Diego, California, USA, January 6-9, 2019*, pages 21–40. SIAM, 2019. doi:10.1137/1.
 659 9781611975482.2.
- 660 8 Paul Erdős and Richard Rado. Intersection theorems for systems of sets. *Journal of the
 661 London Mathematical Society*, 1(1):85–90, 1960.
- 662 9 Oded Goldreich and Avishay Tal. On constant-depth canonical boolean circuits for computing
 663 multilinear functions. In *Computational Complexity and Property Testing - On the Interplay
 664 Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*,
 665 pages 306–325. Springer, 2020. doi:10.1007/978-3-030-43662-9_17.

- 666 10 Oded Goldreich and Avi Wigderson. On the size of depth-three boolean circuits for computing
667 multilinear functions. In *Computational Complexity and Property Testing - On the Interplay*
668 *Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*,
669 pages 41–86. Springer, 2020. doi:10.1007/978-3-030-43662-9_6.
- 670 11 András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold
671 circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993. doi:10.1016/
672 0022-0000(93)90001-D.
- 673 12 Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the*
674 *18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California,*
675 *USA*, pages 6–20. ACM, 1986. doi:10.1145/12130.12132.
- 676 13 Johan Håstad, Stasys Jukna, and Pavel Pudlák. Top-down lower bounds for depth-three
677 circuits. *Computational Complexity*, 5(2):99–112, 1995. doi:10.1007/BF01268140.
- 678 14 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst.*
679 *Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.
- 680 15 Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms*
681 *and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-24508-4.
- 682 16 Daniel M. Kane and Richard Ryan Williams. The orthogonal vectors conjecture for branching
683 programs and formulas. In *10th Innovations in Theoretical Computer Science Conference, ITCS*
684 *2019, January 10-12, 2019, San Diego, California, USA*, volume 124 of *LIPIcs*, pages 48:1–48:15.
685 Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ITCS.2019.48.
- 686 17 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-
687 logarithmic depth. *SIAM J. Discret. Math.*, 3(2):255–265, 1990. doi:10.1137/0403021.
- 688 18 Igor Carboni Oliveira, Rahul Santhanam, and Srikanth Srinivasan. Parity helps to compute
689 majority. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New*
690 *Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 23:1–23:17. Schloss Dagstuhl - Leibniz-
691 Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CCC.2019.23.
- 692 19 Alexander A. Razborov. On the method of approximations. In *Proceedings of the 21st Annual*
693 *ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages
694 167–176. ACM, 1989. doi:10.1145/73007.73023.
- 695 20 Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit
696 complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing,*
697 *1987, New York, New York, USA*, pages 77–82. ACM, 1987. doi:10.1145/28395.28404.
- 698 21 Leslie G. Valiant. Exponential lower bounds for restricted monotone circuits. In *Proceedings*
699 *of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston,*
700 *Massachusetts, USA*, pages 110–117. ACM, 1983. doi:10.1145/800061.808739.
- 701 22 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications.
702 *Theoretical Computer Science*, 348(2-3):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.
- 703 23 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity.
704 In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages
705 3447–3487. World Scientific, 2018.
- 706 24 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix
707 and triangle problems. In *51th Annual IEEE Symposium on Foundations of Computer Science,*
708 *FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 645–654. IEEE Computer
709 Society, 2010. doi:10.1109/FOCS.2010.67.