Electronic Colloquium on Computational Complexity, Report No. 142 (2024)

The Orthogonal Vectors Conjecture and Non-Uniform Circuit Lower Bounds[*]

Ryan Williams[†]
MIT

Abstract

A line of work has shown how nontrivial uniform algorithms for analyzing circuits can be used to derive non-uniform circuit lower bounds. We show how the *non-existence* of nontrivial circuit-analysis algorithms can also imply non-uniform circuit lower bounds. Our connections yield new win-win circuit lower bounds, and suggest a potential approach to refuting the Orthogonal Vectors Conjecture in the $O(\log n)$-dimensional case, which would be sufficient for refuting the Strong Exponential Time Hypothesis (SETH). For example, we show that at least one of the following holds:

- There is an $\varepsilon > 0$ such that for infinitely many n, read-once 2-DNFs on n variables cannot be simulated by *non-uniform* $2^{\varepsilon n}$-size depth-two exact threshold circuits. It is already a notorious open problem to prove that the class E^{NP} does not have polynomial-size depth-two exact threshold circuits, so such a lower bound would be a significant advance in low-depth circuit complexity. In fact, a stronger lower bound holds in this case: the $2^n \times 2^n$ Disjointness Matrix (well-studied in communication complexity) cannot be expressed by a linear combination of $2^{o(n)}$ structured matrices that we call "equality matrices".
- For every $c \geq 1$ and every $\varepsilon > 0$, Orthogonal Vectors on n vectors in $c \log n$ dimensions can be solved in $n^{1+\varepsilon}$ *uniform* deterministic time. This case would provide a strong refutation of the Orthogonal Vectors conjecture, and of SETH: for example, CNF-SAT on n variables and $O(n)$ clauses could be solved in $2^{n/2+o(n)}$ time. Moreover, this case would imply non-uniform circuit lower bounds for E^{NP}, against Valiant series-parallel circuits.

Inspired by this connection, we give evidence from SAT/SMT solvers that the first item (in particular, the Disjointness lower bound) may be false in its full generality. In particular, we present a systematic approach to solving Orthogonal Vectors via constant-sized decompositions of the Disjointness Matrix, which already yields interesting new algorithms. For example, using a linear combination of 6 equality matrices that express $2^6 \times 2^6$ Disjointness, we derive an $\tilde{O}(n \cdot 6^{d/6}) \leq \tilde{O}(n \cdot 1.35^d)$ time and $n \cdot \mathrm{poly}(\log n, d)$ space algorithm for Orthogonal Vectors on n vectors in d dimensions. We show similar results for counting pairs of orthogonal vectors.

[*] This is the full version of a paper that will appear in FOCS 2024.

[†] Work was supported in part by NSF CCF-2127597 and a Frank Quick Faculty Research Innovation Fellowship.

ISSN 1433-8092

1 Introduction

The Orthogonal Vectors problem is a simple task about finding disjoint vectors in a given collection:

ORTHOGONAL VECTORS (**OV**)
Given: n vectors $v_1, \ldots, v_n \in \{0,1\}^d$
Decide: Are there i, j such that $\langle v_i, v_j \rangle = 0$?

The obvious algorithm for **OV** takes $O(n^2 \cdot d)$ time. Folklore $O(n \cdot 2^d \cdot d)$-time and $\tilde{O}(n + 2^d)$-time algorithms are also known (for a reference, see [CST17]). For larger dimensions, truly subquadratic-time algorithms have also been developed: the best known result in this direction is that for all constants $c \geq 1$, **OV** with $d = c \log n$ dimensions can be solved in $n^{2-1/O(\log c)}$ time [AWY15, CW21]. Note that the running time degrades to n^2 as c increases; the conjecture that there is no universal $\varepsilon > 0$ such that **OV** can be solved in $n^{2-\varepsilon}$ time (for all c) is known as the Orthogonal Vectors Conjecture, one of the major hypotheses in fine-grained complexity (see [Vas18] for a survey).

Conjecture 1 (Orthogonal Vectors Conjecture (OVC) [AVW14, BI15, ABV15])**.** *For every constant $\varepsilon > 0$, there is a constant $c \geq 1$ such that* **OV** *cannot be solved in $n^{2-\varepsilon}$ time on instances with $d = c \log n$.*

In other words, OVC states that **OV** requires $n^{2-o(1)}$ time on instances of super-logarithmic dimension.

Why might OVC be true? It is known that faster algorithms for **OV** in $c \log n$ dimensions imply faster algorithms for several other apparently harder problems on collections of vectors, such as finding a pair with minimum inner product [CW19]. Another major piece of evidence for OVC is that the Strong Exponential Time Hypothesis (on the time complexity of SAT) implies that OVC is true [Wil04, WY14]:

Hypothesis 1 (Strong Exponential Time Hypothesis (SETH) [IP01, CIP09])**.** *For every constant $\delta > 0$, there is a constant $k \geq 3$ such that k-SAT cannot be solved in $2^{(1-\delta)n}$ time.*

For this reason, and the fact that **OV** is simple to work with, the OVC has been the engine under the hood of *many* conditional lower bounds in fine-grained complexity (such as [RV13, AVW14, Bri14, ABV15, BI15, BK15, BM16, BI16, ABH$^+$16, BBK$^+$16, GIKW17, BRSV17, WY14, AVW16, CDHL16, APRS16, ED16, IR16, CGR16, KPS17]).

Circuit Lower Bounds from Falsifying OVC. SETH and its relatives also play a major role in a line of work on proving *non-uniform* circuit lower bounds (where we are allowed a separate algorithm for each input length) from *uniform* (deterministic) Circuit SAT algorithms (cf. [Wil10, Wil11, JMV15, Wil18c, AC19, Che19]). For this reason, it is known that if SETH is false for deterministic algorithms, the algorithm's existence would resolve some *non-uniform* circuit lower bound problems which have remained open for decades. The strongest known circuit lower bound consequences (for refuting OV) follow from results of [Val77, CDL$^+$12, JMV15, ABDN18]:

Theorem 1 ([Val77, CDL$^+$12, JMV15, ABDN18])**.** *If OVC is false, then E^{NP} does not have $O(n)$-size Valiant series parallel circuits. If a weaker version of OVC is false, where the dimension of vectors is $2^{O(\log n)^{\varepsilon}}$ for some arbitrarily small $\varepsilon > 0$, then E^{NP} does not have $O(n)$-wire $O(\log n)$-depth circuits of constant fan-in. If a still weaker version of OVC is false, where the dimension of vectors is n^{ε} for some arbitrarily small $\varepsilon > 0$, then E^{NP} does not have $O(n)$-wire $O(\log n)$-depth circuits composed of threshold gates of arbitrarily large fan-in.*

Therefore, if OVC is false and **OV** could indeed be solved in $n^{1.99}$ time, the resulting algorithm would be powerful enough to prove breakthrough *non-uniform* circuit lower bounds.

Could the OVC itself imply non-uniform lower bounds? Could the assumption that **OV** has no subquadratic-time algorithms *also* imply interesting circuit complexity lower bounds? Any such result would immediately imply a "win-win" circuit lower bound: either lower bounds hold because OVC is false, or they hold because OVC is true.

A priori, it looks unlikely that one might obtain circuit lower bounds from OVC. While OVC is clearly a lower bound statement, it is a *uniform* lower bound, an impossibility claim about *algorithms*. We would have to show that lower bounds on *algorithms* imply lower bounds on *non-uniform algorithms*, i.e., circuits. It is well-known that (for example) lower bounds on circuits of size T imply lower bounds on algorithms (specifically, multitape Turing machines) running in time $T^{1-o(1)}$ [PF79]. But a generic connection in the opposite direction (where time lower bounds for algorithms imply circuit size lower bounds) is provably impossible: small non-uniform circuits can decide some *undecidable* problems, and there is **no** algorithm whatsoever for such problems.

In this paper, we exploit the structure of the Orthogonal Vectors problem, as well as structure within low-depth circuit complexity, to show that the OVC indeed implies *non-uniform circuit lower bounds* of particular varieties which are longstanding open problems. Investigating deeper into this connection, we exploit the contrapositive: we are able to show *upper bounds* on such circuits in particular settings, which in turn lead to new combinatorial algorithms for solving the Orthogonal Vectors problem.

The Setup. In a win-win argument proving a claim ϕ, one finds a proposition ψ and shows both $\psi \Rightarrow \phi$ and $\neg\psi \Rightarrow \phi$ are true. The most general way of expressing our proposition ψ is in terms of certain representations of Disjointness matrices. For $d \geq 1$, the Disjointness matrix $\textsc{Disj}_d$ is a $2^d \times 2^d$ Boolean matrix, with rows and columns indexed by all vectors in $\{0,1\}^d$; $\textsc{Disj}_d(i,j) = 1$ if and only if the i-th vector is orthogonal to the j-th vector (they share no ones, so they are "disjoint"). Observe that **OV** is equivalent to the following task:[1]

> *Given a submatrix of* $\textsc{Disj}_d$ *specified by* n *rows* L *and* n *columns* R*, determine if the submatrix of entries* $L \times R$ *contains a* 1.

We consider two generic ways of expressing $\textsc{Disj}_d$ as sums of other matrices. For simplicity, we will consider only one type of $\textsc{Disj}_d$ matrix representation here, but our results can be generalized much further (see section 7 at the end of the paper). We say that a 0-1 $m \times n$ matrix A is an *equality matrix* if there exist defining vectors $u \in \mathbb{N}^m$ and $v \in \mathbb{N}^n$ such that for all i, j, $A[i,j] = 1$ if and only if $u[i] = v[j]$.[2] Without loss of generality, we may assume all entries in u and v are in the range $\{1, \ldots, m+n\}$.[3] The *equality rank* of a matrix A is defined to be the smallest number r of equality matrices $M_1, \ldots, M_r$ such that there are constants $\alpha_1, \ldots, \alpha_r$ satisfying $A = \sum_i \alpha_i M_i$. For a Boolean matrix A, we define the *weak equality rank of* A to be the smallest number of equality matrices $M_1, \ldots, M_r$ such that there are constants $\alpha_1, \ldots, \alpha_r$ satisfying the conditions:

- $A[i,j] = 0$ implies that $\sum_i \alpha_k \cdot M_k[i,j] = 0$.
- $A[i,j] = 1$ implies that $\sum_i \alpha_k \cdot M_k[i,j] \neq 0$.

These rank generalizations are of interest to researchers in circuit complexity, due to their connections to low-depth circuit lower bounds [Wil18b, Wil18c, HHH23]. An *exact threshold function* $f : \{0,1\}^n \to \{0,1\}$ is defined by weights $\alpha_1, \ldots, \alpha_n, t \in \mathbb{R}$, so that for all $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, $f(x) = 1$ if and only if $\sum_i \alpha_i x_i = t$. An $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit of size $s(n)$ is a depth-two circuit where the bottom layer consists of $s(n)$ exact threshold gates over variables $x_1, \ldots, x_n$, and the top layer is a single exact threshold gate which takes the outputs of the $s(n)$ bottom layer as inputs. A $\mathbf{SUM} \circ \mathbf{ETHR}$ circuit of size $s(n)$ is similar, except the top layer is simply a linear combination (over the rationals) of the s bottom layer gates. For over 30 years, it has been a notorious open problem to find efficient functions exhibiting super-polynomial lower bounds on $\mathbf{SUM} \circ \mathbf{ETHR}$ circuits [ROS94], and an open problem since 2010 to find super-polynomial lower bounds for $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits [HP10, Wil18c]. (See Section 2 for more discussion on the history of these low-depth threshold circuits.) It is only known that $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits require $\Omega(n^{1.5-o(1)})$ size for some functions in P [KW16], and that for every k there are functions in NP without n^k-size $\mathbf{SUM} \circ \mathbf{ETHR}$ circuits [Wil18c]. These circuit classes are naturally connected to equality-rank decompositions, in the following way:

Theorem 2 ([Wil18b, Wil18c, HHH23]). *Let* $f : \{0,1\}^{2n} \to \{0,1\}$ *and let* M_f *be a* $2^n \times 2^n$ *matrix indexed by* n*-bit strings such that for all* x, y, $M_f(x,y) = f(xy)$*. Let* $\neg M_f(x,y) = 1 - M_f(x,y)$.

[1] **OV** is fine-grained equivalent to the following variant: given "red" vectors $u_1, \ldots, u_n$ and "blue" vectors $v_1, \ldots, v_n$, determine if there is a red-blue orthogonal pair.

[2] Such matrices have been given other names in the literature, such as equivalence graphs [Alo86], fat matchings [PR94, Juk06], the adjacency matrices of P4-free bipartite graphs [BBM+21], and blocky matrices [HHH23, PSS23, AY24]. We find the name "equality matrix" more natural.

[3] We could sort the entries of the defining vectors, and replace each entry by its rank in sorted order.

- *If f has an* $\mathbf{ETHR} \circ \mathbf{ETHR}$ *circuit of size s, then $\neg M_f$ has weak equality rank at most $s+1$.*
- *If f has a* $\mathbf{SUM} \circ \mathbf{ETHR}$ *circuit of size s, then M_f has equality rank at most s.*

(See the Preliminaries in Section 2 for a proof.) We say a function $f : \mathbb{N} \to \mathbb{N}$ is *subexponential* if for all $\varepsilon > 0$ and all sufficiently large $d \in \mathbb{N}$, $f(d) \leq 2^{\varepsilon d}$. Our connection to solving **OV** shows that, if the Boolean Inner Product function ($\bigvee_{i=1}^{d}(x_i \wedge y_i)$) has subexponential-size non-uniform $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits, then the OV Conjecture is false: in fact there is a *nearly-linear time* algorithm for **OV**. More generally, our main technical result is:

Theorem 3 (Subexponential Weak Equality Rank Refutes OVC)**.** *Suppose there is a subexponential function $f(d)$ such that for all d, $\textsc{Disj}_d$ has weak equality rank at most $f(d)$. Then for every $c \geq 1$ and every $\varepsilon > 0$,* **OV** *on n vectors in $c \log n$ dimensions can be solved in $n^{1+\varepsilon}$ deterministic time. (As a consequence, k-SAT can be solved in $2^{n/2+o(n)}$ deterministic time, for all constants k.)*

The consequence of Theorem 3 is a deterministic k-SAT algorithm which would be nearly as fast as the best-known *quantum* algorithm (based on Grover search [Gro96, ACL+20, BPS21]). Note that the *complement* of $\textsc{Disj}_d$, i.e., the $2^d \times 2^d$ $\textsc{Intersection}$ matrix (i.e. the matrix of Boolean Inner Product) has weak equality rank d.[4] Theorem 3's hypothesis is that $\textsc{Disj}_d$ itself has weak equality rank $2^{o(d)}$. This would follow for example, if $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits were efficiently closed under complement.

The most striking aspect of Theorem 3 is that the hypothesis is a **non-uniform upper bound**. We do not place any computational bounds on how difficult it might be to produce the rank decomposition for $\textsc{Disj}_d$. However, we are still able to obtain a *uniform* algorithm for **OV**, from the hypothesis. The contrapositive of Theorem 3 states that a *uniform* time lower bound on **OV** implies an exponential *non-uniform* size lower bound on $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits computing the Boolean Inner Product, by Theorem 2. Furthermore, a strengthening of Theorem 3 to equality rank would imply faster algorithms for *counting* the number of orthogonal pairs.

Theorem 4 (Subexponential Equality Rank Counts OV Pairs)**.** *Suppose that there is a subexponential function $f(d)$ such that for all d, $\textsc{Disj}_d$ has equality rank at most $f(d)$. Then for every $c \geq 1$ and every $\varepsilon > 0$, the* **number** *of orthogonal pairs among n vectors in $\{0,1\}^{c \log n}$ can be counted in $n^{1+\varepsilon}$ deterministic time. (As a consequence, $\#k$-SAT can be solved in $2^{n/2+o(n)}$ deterministic time, for all constants k.)*

In fact, even if $\textsc{Disj}_d$ has $2^{o(d)}$ equality rank over *some* finite field, we would still obtain a randomized nearly-linear time algorithm for **OV** in $c \log n$ dimensions for all constants $c \geq 1$.

Theorems 3 and 4 provide an intriguing path to win-win lower bounds. (In fact, the hypotheses required are even weaker: we just need low equality rank *rigidity*; see Section 5.1.) On the one hand, if the Disjointness matrix does not have subexponential weak equality rank, then Theorem 2 implies that Boolean Inner Product does not have subexponential-size $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits: this would be a significant advance in depth-two circuit complexity. On the other hand, if the Disjointness matrix *does* have subexponential weak equality rank, then Theorem 3 implies that the Orthogonal Vector Conjecture is false, implying *another* kind of circuit lower bound via SAT algorithms!

Corollary 1 (Win-Win Circuit Lower Bounds)**.** *At least one of the following non-uniform circuit lower bounds is true:*

- E^{NP} *does not have $O(n)$-size Valiant series parallel circuits.*
  (Lower bounds of this type have been open for decades; see [Val77] and [AB09], Chapter 14, Frontier 3.)
- *There is an $\varepsilon > 0$ such that Boolean Inner Product on n-bit vectors does not have $2^{\varepsilon n}$-size* $\mathbf{ETHR} \circ \mathbf{ETHR}$ *circuits.*

Applying Theorem 4, we can prove more non-uniform lower bounds from assuming the number of OV pairs cannot be counted efficiently, for example:

Theorem 5. *The Orthogonal Vectors Conjecture implies that the inner product of n-bit vectors modulo 2 cannot be expressed by* $\mathbf{SUM} \circ \mathbf{ETHR}$ *circuits of size $2^{o(n)}$.*

Theorem 5 directly implies another win-win circuit lower bound, with "Inner Product Mod 2" replacing "Boolean Inner Product" in the second bullet of Corollary 1, and "$\mathbf{SUM} \circ \mathbf{ETHR}$" replacing "$\mathbf{ETHR} \circ \mathbf{ETHR}$".

[4] To compute $\bigvee_{i=1}^{d}(x_i \wedge y_i)$, each of the d equality matrices in the rank decomposition can handle an $x_i \wedge y_i$ term.

New Algorithms for OV, and the Prospect of Refuting OVC(?!). The connections given by Theorem 2 and Theorem 4 are general enough to provide templates for designing faster algorithms for detecting and counting orthogonal pairs: for a *fixed constant* k, we can search for succinct equality-rank decompositions of the $2^k \times 2^k$ Disjointness matrix $\textsc{Disj}_k$, and use these constant-sized decompositions to construct **OV** algorithms for all dimensions $d \gg k$. That is, we can "bootstrap" from a constant-sized decomposition on constant-sized $\textsc{Disj}$ matrices, to get good decompositions for *all* $\textsc{Disj}$ matrices. This situation is similar to that of matrix multiplication, where one finds a good algorithm for constant-sized matrices, and applies it recursively to design an algorithm for all $n \times n$ matrices (a la Strassen [Str69]).

Encoding the search for efficient equality-rank decompositions in the SAT solvers Minisat [SE05], and CaDiCaL and Kissat [BFFH20], as well as the SMT solver Z3 [dMB08], we discovered surprisingly small rank decompositions for $\textsc{Disj}_k$ for small constant k. Applying our algorithmic results, we obtain new combinatorial **OV** algorithms.

Theorem 6 (Section 6)**.** *The* weak equality rank *of* $2^6 \times 2^6$ *Disjointness is* 6. *Applying Theorems 8 and 9 directly, there is a randomized algorithm for* **OV** *that runs in* $\tilde{O}(n \cdot 6^{d/6}) \le \tilde{O}(n \cdot 1.35^d)$ *time and* $n \cdot \text{poly}(\log n, d)$ *space.*

Theorem 7 (Section 6)**.** *The* equality rank *of* $2^5 \times 2^5$ *Disjointness is (at most)* 5. *Applying Theorem 8 and Theorem 14 directly, there is a deterministic algorithm for counting* **OV** *pairs (a.k.a.* $\#$**OV***) that runs in* $\tilde{O}(n \cdot 5^{d/5}) \le \tilde{O}(n \cdot 1.38^d)$ *time and* $n \cdot \text{poly}(\log n, d)$ *space.*

Unlike the $n^{2-1/O(\log c)}$-time algorithm for **OV** with $c \log n$ dimensions [AWY15, CW21] which relies on fast rectangular matrix multiplication, the above algorithms admit very simple implementations. Our algorithm is faster than all folklore algorithms for $d \in [1.76 \log(n), 2.3 \log(n)]$.[5]

Our computer searches found that for every $k \in \{1, \dots, 6\}$, the equality rank and weak equality rank of $\textsc{Disj}_k$ are never more than k. Recall Theorems 3 and 4 say that if these rank measures are subexponential in k, then the Orthogonal Vectors Conjecture is false. These findings lead us to believe that, if these rank measures are actually exponential, they must be rather low exponentials. Indeed, work of Jukna [Juk06] implies that $\textsc{Disj}_d$ requires an *OR* of $\Omega(1.08^d)$ equality matrices (see Appendix A for a proof). However, an "OR of equality matrices" seems significantly weaker than a linear combination of equality matrices which is nonzero on the ones of the matrix (the notion of weak equality rank). At any rate, an exponential lower bound on weak equality rank would imply exponential-size lower bounds against $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits for a simple function, by Theorem 2!

To summarize, our results from SAT/SMT solvers indicate that linear combinations of equality matrices are surprisingly powerful, in that they can represent $\textsc{Disj}$ in unexpectedly succinct ways; we believe that Theorems 6 and 7 are just the beginning of a new approach to solve the OV problem.

Fine-Grained Complexity Need Not Despair. It is worth pointing out that, even if $\textsc{Disj}_d$ did turn out to have weak equality rank $2^{o(d)}$, that would only refute the "standard" (and strongest) form of the Orthogonal Vectors Conjecture (Conjecture 1), which is the case where the vector dimension is $O(\log n)$ (Theorem 3). While refuting Conjecture 1 would be enough to refute SETH, many fine-grained lower bounds on problems in P in the literature only require a weaker form of Conjecture 1 to be true: that OV requires $n^{2-o(1)}$ time in the case where the vector dimension is n^{ε} for small $\varepsilon > 0$ (see for example, [GIKW17, ABDN18]). This weaker "moderate dimension" Orthogonal Vectors Conjecture could still be refuted by the approach in this paper, if $\textsc{Disj}_d$ has weak equality rank $\text{poly}(d)$ with a rank decomposition that is itself uniformly computable in $\text{poly}(d)$ time (that is, the rank decomposition is "explicit" in the sense of Definition 1). However, such a small decomposition seems to be a less likely prospect at this time.

2 Preliminaries

We assume basic familarity with computational complexity theory [AB09] and circuit complexity [Vol99]. We will be particularly interested in depth-two circuits of certain kinds.

[5] Recall the folklore algorithms take time $O(n^2 d)$, $O(n \cdot 2^d \cdot d)$ time, and $\tilde{O}(n + 2^d)$ time. Note the latter algorithm requires space complexity $\Omega(2^d)$, larger than ours for $d > \log(n)$. When we limit the space complexity to $n \cdot \text{poly}(\log n)$, our algorithm is faster for $d \in [\log(n), 2.3 \log(n)]$.

Sums of Thresholds. We let $\mathbf{SUM} \circ \mathbf{THR}$ be linear combinations (over the rationals) of linear threshold functions (LTFs). Such circuits are also known in the machine learning literature as *depth-two neural networks with sign activation functions*. By results of [HP10], every s-size $\mathbf{SUM} \circ \mathbf{THR}$ can be expressed as an $(s \cdot \text{poly}(n))$-size $\mathbf{SUM} \circ \mathbf{ETHR}$, i.e., a linear combination of *exact threshold functions*, and every s-size $\mathbf{SUM} \circ \mathbf{ETHR}$ can be expressed as a $2s$-size $\mathbf{SUM} \circ \mathbf{THR}$, so the two representations are essentially equivalent.

Thirty years ago, Roychowdhury, Orlitsky, and Siu [ROS94] noted that no interesting size lower bounds were known for computing Boolean functions with $\mathbf{SUM} \circ \mathbf{THR}$ circuits, beyond the few that are/were known for $\mathbf{THR} \circ \mathbf{THR}$ [HMP+93, ROS94, KW16, CSS16, Tam16, ACW16]. (By "computing" a function $f : \{0,1\}^n \to \{0,1\}$, we mean that we want the linear combination of LTFs to evaluate to the 0-1 value $f(x)$ on all $x \in \{0,1\}^n$.) The problem was raised again more recently by Hansen and Podolskii [HP10]. It remains largely open to find an efficiently computable f that does not have poly(n)-size $\mathbf{SUM} \circ \mathbf{THR}$ circuits; the best-known result in this direction shows that for every k, there is a function in NP which does not have n^k-size $\mathbf{SUM} \circ \mathbf{THR}$ circuits [Wil18c]. Our results (Theorem 4) show that poly(n)-size *non-uniform* $\mathbf{SUM} \circ \mathbf{THR}$ circuits for very simple functions such as CNFs and DNFs would already refute the Orthogonal Vectors Conjecture in a strong way.

The problem of proving lower bounds on the (presumably more expressive) class of $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits was first explicitly raised in a work of Hansen and Poldoskii [HP10]. In particular, they asked whether $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits are efficiently closed under complement: whether the negation of an $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit of size s can be computed by an $\mathbf{ETHR} \circ \mathbf{ETHR}$ of size (say) $2^{o(s)}$. Our work (combining Theorem 2 and Theorem 3) implies that if the answer is yes, then the Orthogonal Vectors Conjecture would be false, even if the negation circuit is non-uniform.

For completeness, we show here how "small" depth-two exact threshold circuits for a function f imply "small" equality rank for the matrix of the function f. The proof follows readily from observations in prior work [Wil18b, Wil18c, HHH23].

Reminder of Theorem 2. *Let $f : \{0,1\}^{2n} \to \{0,1\}$ and let M_f be a $2^n \times 2^n$ matrix indexed by n-bit strings such that for all $x, y \in \{0,1\}^n$, $M_f(x,y) = f(xy)$. Let $\neg M_f(x,y) = 1 - M_f(x,y)$.*

- *If f has an $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit of size s, then $\neg M_f$ has weak equality rank at most $s + 1$.*
- *If f has a $\mathbf{SUM} \circ \mathbf{ETHR}$ circuit of size s, then M_f has equality rank at most s.*

Proof. We prove the first item; the second has an analogous proof. Let $f : \{0,1\}^{2n} \to \{0,1\}$ have an $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit of size s, and let $g_1, \ldots, g_s$ be the bottom-layer gates of the circuit, each taking $2n$ variables.

Fix a gate g_i defined by weights $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, t$, so that $g_i(x_1, \ldots, x_n, y_1, \ldots, y_n) = 1$ if and only if $\sum_{j=1}^{n}(\alpha_j x_j + \beta_j y_j) = t$. We claim that the $2^n \times 2^n$ matrix M_{g_i} corresponding to g_i is an equality matrix. Letting $x, y \in \{0,1\}^n$, we define 2^n-length vectors $u^{(i)}$ and $v^{(i)}$ as:

$$u^{(i)}[x] = \sum_j \alpha_j x_j, \quad v^{(i)}[y] = \left(t - \sum_j \beta_j y_j\right).$$

For every x, y, we have $g_i(x,y) = 1$ if and only if $u^{(i)}[x] = v^{(i)}[y]$. Therefore the matrix M_{g_i} is an equality matrix.

Now let $\gamma_1, \ldots, \gamma_s, t'$ be the weights defining the output gate of the $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit for f, so that the output gate returns 1 if and only if $(\sum_i \gamma_i g_i) = t'$. Let J denote the $2^n \times 2^n$ all-ones matrix. Define the linear combination of $s + 1$ equality matrices

$$N = \sum_{i=1}^{s} \gamma_i \cdot M_{g_i} - t' \cdot J.$$

(Note J is an equality matrix: set the defining vectors u, v to have $u[i] = 1$ and $v[i] = 1$ for all i.) Finally, observe that for all x, y, $f(x,y) = 1$ implies that $N(x,y) = (\sum_i \gamma_i \cdot g_i(x,y)) - t' = 0$, and $f(x,y) = 0$ implies $N(x,y) \neq 0$. That is, the matrix N is a weak representation of $\neg M_f$: the two matrices share exactly the same positions of zeroes (and nonzeroes). □

2.1 More Related Work

Jukna [Juk06] shows that Disjointness (expressed as $\bigwedge_{i=1^n}(x_i \vee y_i)$ over $2n$ variables $x_1, \ldots, x_n, y_1, \ldots, y_n$) does not have OR ∘ AND ∘ MOD2 circuits of size 1.08^n. However his proof actually shows the $2^n \times 2^n$ Disjointness matrix cannot be represented by an OR of 1.08^n equality matrices, which implies OR ∘ **ETHR** lower bounds, by a similar argument as in Theorem 2. (We give a self-contained exposition of this result in Appendix A.) Diamond and Yehudayoff [DY22] also give an exponential lower bound for computing Disjointness with an OR of **ETHR** gates.

In the classic paper of Babai, Frankl, and Simon [BFS86] introducing communication complexity classes, they consider the problem of "reducing" Disjointness to Equality in a communication complexity sense. Their results imply that any decision tree for Disjointness with oracle calls to Equality at each node requires depth $\Omega(\sqrt{n})$. The $\Omega(n)$ bound on the randomized complexity of Disjointness of Kalyanasundaram and Schintger [KS92] implies an $\Omega(n/\log n)$ depth bound, which was recently sharpened to an asymptotically tight lower bound of $\Omega(n)$ by Chattopadhyay, Lovett, and Vinyals [CLV19].

Prior algorithms for **OV** in restricted settings consider the Disjointness matrix. Nederlof and Wegrzycki [NW21] show that certain subrectangles of the disjointness matrix (where the vectors have at most $d/4$ ones) can be efficiently expressed as the OR of a "low exponential" number of rank-one matrices. They use this (and other properties) to obtain a new **OV** algorithm for "sparse" vectors, which in turn implies a more space-efficient Subset Sum algorithm. It would be interesting if our new approach to designing **OV** algorithms (Section 6) via equality matrices could be useful here.

In terms of related work that also uses computer searches to obtain better complexity upper bounds, Amano [AM05, Ama10, Ama20] has used computer search to construct improved depth-two threshold circuits for the inner product modulo 2 function, as well as prove lower bounds by a linear programming representations.

3 From Weak Equality Rank to Algorithms for Orthogonal Vectors

In this section, we prove the main result of the paper:

Reminder of Theorem 3. *Suppose there is a subexponential function $f(d)$ such that for all d, $\textsc{Disj}_d$ has weak equality rank at most $f(d)$. Then for every $c \geq 1$ and every $\varepsilon > 0$,* **OV** *on n vectors in $c \log n$ dimensions can be solved in $n^{1+\varepsilon}$ deterministic time.*

The first step of the proof is to use the non-uniform hypothesis in Theorem 3 to obtain an *efficiently computable* rank decomposition for $\textsc{Disj}_d$ for sufficiently large d.

Definition 1. *Consider a rank decomposition $\sum_{k=1}^{r} \alpha_k M_k$ of $2^d \times 2^d$ equality matrices, where the α_k are rationals, and each equality matrix M_k is defined by the 2^d-length vectors $u^{(k)}$ and $v^{(k)}$. We say that the rank decomposition is* explicit *if there is a* poly(d)*-time algorithm that, given $(i, j, k) \in [2^d] \times [2^d] \times [r]$, the algorithm outputs $u^{(k)}[i]$, $v^{(k)}[j]$, and α_k.*

Our first theorem says that, given a small (constant-sized) equality-rank decomposition for $\textsc{Disj}_k$ for constant k, there is an "efficient" rank decomposition for $\textsc{Disj}_d$ for all large d: the number of matrices in the decomposition is nontrivial, and the decomposition is also *uniform*, i.e., efficiently computable. This is the key step which lets us start with a non-uniform rank decomposition (or non-uniform **ETHR** ∘ **ETHR** circuit family), and obtain a uniform rank decomposition.

Theorem 8 (Uniformization)**.** *Suppose for some fixed k, r, $\textsc{Disj}_k$ has (weak) equality rank r. Then for all $d \geq k$, $\textsc{Disj}_d$ has (weak) equality rank at most $O(r^{d/k})$ with an* explicit *rank decomposition.*

Proof. First we assume d is divisible by k, then we show how to remove this assumption. We also just prove the below for weak equality rank; the proof for equality rank is analogous.

By hypothesis, the $2^k \times 2^k$ matrix $\textsc{Disj}_k$ has weak equality rank r. Let $M_1, \ldots, M_r$ be equality matrices defined by the respective vector pairs $u^{(1)}, v^{(1)}, \ldots, u^{(r)}, v^{(r)}$, and let $\alpha_1, \ldots, \alpha_r \in \mathbb{Q}$ be weights, so that $\textsc{Disj}_k[x, y] = 0$ implies $\sum_i \alpha_i M_i[x, y] = 0$, and $\textsc{Disj}_k[x, y] = 1$ implies $\sum_i \alpha_i M_i[x, y] \neq 0$. Note that all entries in all $u^{(i)}$ and $v^{(i)}$

can be assumed to be integers in $\{1, \ldots, 2^{k+1}\}$. Without loss of generality, the numerators and denominators of all weights α_i are bounded by a function of the constants k and r.[6]

Now consider the Disjointness function on $2d$ bits. For $u, v \in \{0,1\}^d$, we have:

$$\textsc{Disj}_d(u,v) = \bigwedge_{i=1}^{n} (\neg u_i \vee \neg v_i).$$

Recall the Kronecker power $A^{\otimes t}$ of an $N \times N$ matrix A is an $N^t \times N^t$ matrix whose entries are indexed by $(i_1, \ldots, i_t), (j_1, \ldots, j_t) \in [N]^t$ and

$$A^{\otimes t}((i_1, \ldots, i_t), (j_1, \ldots, j_t)) = \prod_{k=1}^{t} A[i_k, j_k].$$

It is a well-known property of the Disjointness matrix (see for example [Alm21]) that

$$\textsc{Disj}_d = (\textsc{Disj}_k)^{\otimes d/k}.$$

Intuitively, $\textsc{Disj}_k$ computes disjointness on k-bit vectors, and their d/k-th tensor power computes disjointness of d-bit vectors by computing disjointness on k-bit parts (out of d/k total parts), then taking the product of the results. If any one of the parts is not disjoint (some entry is 0) then the product is 0. If all parts are disjoint (all entries are 1) then the product is 1.

Replacing $\textsc{Disj}_k$ with its assumed rank decomposition, let $M = (\sum_i \alpha_i M_i)^{\otimes d/k}$, where the $2^k \times 2^k$ equality matrix M_i is defined by the vectors $u^{(i)}$ and $v^{(i)}$. Since $\textsc{Disj}_k[i,j] = 0$ implies $\sum_k \alpha_k \cdot M_k[i,j] = 0$, and $\textsc{Disj}_k[i,j] = 1$ implies $\sum_k \alpha_k \cdot M_i[i,j] \neq 0$, we have that:

- $\textsc{Disj}_d[i,j] = 0$ implies that $M[i,j] = 0$.
- $\textsc{Disj}_d[i,j] = 1$ implies that $M[i,j] \neq 0$.

Therefore, the matrix M models $\textsc{Disj}_d$ precisely as we would like in the notion of weak equality rank. We claim that M can be written as a linear combination of $r^{d/k}$ equality matrices.

Consider an entry indexed by $(i_1, \ldots, i_{d/k}), (j_1, \ldots, j_{d/k}) \in [2^k]^{d/k}$ of the matrix M. By definition, this entry equals

$$\prod_{m=1}^{d/k} \left(\sum_{\ell=1}^{r} \alpha_\ell \cdot \left[u^{(\ell)}[i_m] = v^{(\ell)}[j_m] \right] \right),$$

where here we use the Iverson bracket notation $[P]$ to output 1 when P is true, and 0 otherwise. The idea now is to apply distributivity to the product of sums, to get a sum of $r^{d/k}$ equality matrices. In particular, indexing the equality matrices in our desired decomposition by $q = (q_1, \ldots, q_{d/k}) \in [r]^{d/k}$, the coefficient of the q-th matrix is

$$\prod_{m=1}^{d/k} \alpha_{q_m}.$$

For $(i_1, \ldots, i_{d/k}), (j_1, \ldots, j_{d/k}) \in [2^k]^{d/k}$, the $(i_1, \ldots, i_{d/k}), (j_1, \ldots, j_{d/k})$ entry in the $(q_1, \ldots, q_{d/k})$-th matrix is 1 if and only if

$$\prod_{m=1}^{d/k} \left[u^{(q_m)}[i_m] = v^{(q_m)}[j_m] \right] = 1.$$

[6] Fixing the matrix $\textsc{Disj}_k$ and the specific Boolean equality matrices M_i, the problem of finding suitable α_i's for a weak equality rank decomposition can be posed as a system of 2^{2k} linear inequalities in the $r < 2^{2k}$ variables $\alpha_1, \ldots, \alpha_r$, with $0/1$ coefficients. In particular, for each $x, y \in \{0,1\}^k$, we include the equation $\sum_i \alpha_i M_i[x,y] = 0$ in the system when $\textsc{Disj}_k[x,y] = 0$, and we include either $\sum_i \alpha_i M_i[x,y] \geq 1$ or $\sum_i \alpha_i M_i[x,y] \leq 1$ when $\textsc{Disj}_k[x,y] = 1$. (We are already assuming there is a valid solution of α_i's, so we already know which of the two inequalities to choose, to get a feasible solution. By scaling, we can choose 1 as a suitable cut-off for the threshold.) By standard arguments in linear programming (for example, Theorem 4.4 in [KV11]), the numerators and denominators of all α_i's may be bounded by functions of k and r. An analogous argument holds for equality rank decompositions (in which case systems of linear equations are solved).

Define $U^{(q)}(i_1, \dots, i_{d/k})$ to be the concatenation of $u^{(q_m)}(i_m)$ over all $m = 1, \dots, d/k$, and define $V^{(q)}(j_1, \dots, j_{d/k})$ to be the concatenation of $v^{(q_m)}(j_m)$ over all m. Then, the vectors $U^{(q)}$ and $V^{(q)}$ define the q-th matrix as an equality matrix. Therefore M can be written as a linear combination of $r^{d/k}$ equality matrices.

Finally, we show that the rank decomposition can be computed uniformly. Note that given any $i = (i_1, \dots, i_{d/k}), j = (j_1, \dots, j_{d/k}) \in [2^k]^{d/k}$ and $q = (q_1, \dots, q_{d/k}) \in [r]^{d/k}$, we can compute $U^{(q)}(i_1, \dots, i_{d/k})$, $V^{(q)}(j_1, \dots, j_{d/k})$, and the coefficient $\prod_{m=1}^{d/k} \alpha_{q_m}$ for the q-th matrix, all in poly(d) time, by hard-coding the constant-sized vectors $u^{(i)}, v^{(i)}$ as well as the constant-sized values α_i, in the algorithm. Observe that the bit complexity of each coefficient is only $O(d)$.

In the above, we assumed that d is divisible by k. If that is not the case, then we let $d' \geq d$ be the smallest integer that is at least d and is divisible by k, and perform the above with d' in place of d. This yields a weak equality-rank decomposition of $r^{d'/k} \leq r^{d/k+1} \leq O(r^{d/k})$ equality matrices. □

We remark that we do not require k to be constant, in order to get an explicit rank decomposition: for small but unbounded $k(d)$, we could still brute-force search for a weak equality-rank decomposition for DISJ$_{k(d)}$, using the fact that the coefficients of the decomposition are functions of $k(d)$. This observation will be useful in a later uniformization result (Theorem 12).

Now we demonstrate how explicit rank decompositions of equality matrices can be algorithmically useful. Here, we state the results in more generality, to illustrate how powerful the paradigm can be. For a family $\mathcal{M} = \{M_d\}$ of matrices, where M_d is $2^d \times 2^d$, we define a general "satisfying pairs problem":

$\mathcal{M}$-Satisfying-Pairs
Input: integer $d \geq 1$ and two sets $L, R \subseteq [2^d]$.
Decide: if there are $i \in L$ and $j \in R$ such that $M_d[i, j] = 1$.

Observe that for the family of DISJ matrices, the DISJ-Satisfying-Pairs problem is exactly the Orthogonal Vectors problem. First, we show that any matrix family that has "small" weak equality rank has a fast randomized satisfying pairs algorithm. This version, where the running time is linear in the rank r, will be useful for our concrete algorithms for solving **OV**.

Theorem 9 (Randomized Algorithm for Satisfying Pairs)**.** *Suppose a family $\mathcal{M} = \{M_d\}$ of matrices, where M_d is $2^d \times 2^d$, has weak equality rank at most r with an explicit rank decomposition. Then the $\mathcal{M}$-Satisfying-Pairs problem with $|L| = |R| = n$ and dimension d can be solved in randomized $r \cdot n \cdot$ poly$(d, \log n)$ time and $n \cdot$ poly$(\log n, d)$ space.*

Proof. Let us index the rows and columns of M_d by d-bit vectors. Since M_d has weak equality rank at most r, let $E_1, \dots, E_r$ be $2^d \times 2^d$ equality matrices, where each E_k is defined in terms of the vectors $u^{(k)}, v^{(k)}$ of length 2^d, so that $E_k[i, j] = 1$ if and only if $u^{(k)}[i] = v^{(k)}[j]$. Since the decomposition is assumed to be explicit, each entry $u^{(k)}[i]$ and $v^{(k)}[i]$ can be computed in poly(d) time given i, k, and there are poly(d)-time computable coefficients $\alpha_1, \dots, \alpha_r$ such that:

- $M_d[i, j] = 0$ implies $\sum_{k=1}^{r} \alpha_k E_k[i, j] = 0$, and
- $M_d[i, j] = 1$ implies $\sum_{k=1}^{r} \alpha_k E_k[i, j] \neq 0$.

Now suppose we are given two sets of n Boolean vectors $U = \{a_1, \dots, a_n\} \subseteq \{0, 1\}^d$ and $V = \{b_1, \dots, b_n\} \subseteq \{0, 1\}^d$, and we wish to efficiently determine if there is an i, j such that $M_d[a_i, b_j] = 1$. (Here, U just corresponds to a relabeling of the set of integers L, and V corresponds to R.) By the above properties, we have:

(1) If there is a satisfying pair, then there is an i, j such that $M_d[a_i, b_j] = 1$, and $\sum_{k=1}^{r} \alpha_k E_k[a_i, b_j] \neq 0$.

(2) If there is no satisfying pair, then for all i, j we have $M_d[a_i, b_j] = 0$, then for all i, j, $\sum_{k=1}^{r} \alpha_k E_k[a_i, b_j] = 0$.

Consider the following degree-two polynomial in $2n$ variables $x_1, \ldots, x_n, y_1, \ldots, y_n$:

$$S(x_1, \ldots, x_n, y_1, \ldots, y_n) := \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\sum_{k=1}^{r} \alpha_k E_k[a_i, b_j] \right) \cdot x_i \cdot y_j.$$

By properties (1) and (2), there is no satisfying pair if and only if S is identically zero.

Let N be a sufficiently large (constant) positive integer. Choosing uniform and independent random values $s_i, t_i \in \{0, 1, \ldots, N\}$ for each variable x_i and y_i respectively, the DeMillo-Lipton-Schwartz-Zippel Lemma [DL78, Sch80, Zip79] implies that if there is a satisfying pair, then

$$S(s_1, \ldots, s_n, t_1, \ldots, t_n) \neq 0$$

with probability at least $1 - 2/N$.[7] (Otherwise, $S(s_1, \ldots, s_n, t_1, \ldots, t_n) = 0$ with probability 1.) Note that, since the polynomial S is only degree-two, N is a constant, and each α_k is at most poly(d) bits long, the magnitude of $S(s_1, \ldots, s_n, t_1, \ldots, t_n)$ is at most $n^2 \cdot r \cdot 2^{\text{poly}(d)}$, i.e., the value can be stored in $O(\log(nr)) + \text{poly}(d)$ bits.

We can efficiently evaluate the polynomial S on the points $s_1, \ldots, s_n, t_1, \ldots, t_n$, as follows. Rearranging the order of summation for S, we have:

$$S(s_1, \ldots, s_n, t_1, \ldots, t_n) = \sum_{k=1}^{r} \alpha_k \cdot \left(\sum_{i=1}^{n} \sum_{j=1}^{n} s_i \cdot t_j \cdot \left[u^{(k)}[a_i] = v^{(k)}[b_j] \right] \right).$$

Thus we can think of computing S with r calls to certain double sums over all i, j.

For $k \in [r]$, define the double-sum

$$T_k = \alpha_k \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} s_i \cdot t_j \cdot \left[u^{(k)}[a_i] = v^{(k)}[b_j] \right].$$

We show how to quickly compute each T_k. Since the rank decomposition is explicit, each entry $u^{(k)}[a_i]$ and $v^{(k)}[b_j]$ takes only poly(d) time to compute, and thus also has only poly(d) bit complexity. For a fixed k, we sort the list of all $2n$ relevant $u^{(k)}[a_i]$ and $v^{(k)}[b_j]$ in $n \log n \cdot \text{poly}(d)$ time, putting equal $u^{(k)}[a_i]$'s before equal $v^{(k)}[b_j]$'s in the sorted order. We also store the index i of each $u^{(k)}[a_i]$ and the index j of each $v^{(k)}[b_j]$ as a secondary key in our sorted order. To compute T_k, we can sweep through the sorted order in one pass. In particular, the sorted order gives us a sequence of equivalence classes of those $u^{(k)}[a_i]$'s and $v^{(k)}[b_j]$'s that are all equal. (One could also use a union-find data structure to build the equivalence classes, if one cares about polylog factors [TvL84].) For each equivalence class $\mathcal{C}$ of equal values, we add the quantity

$$\left(\sum_{i \,:\, u^{(k)}[a_i] \in \mathcal{C}} s_i \right) \cdot \left(\sum_{j \,:\, v^{(k)}[b_j] \in \mathcal{C}} t_j \right)$$

to a running sum, using the secondary keys i and j of each $u^{(k)}[a_i]$ and $v^{(k)}[b_j]$ to determine the appropriate s_i and t_j. Finally, we multiply the resulting sum by α_k. This evaluates T_k, since

$$\begin{aligned} T_k &= \alpha_k \cdot \sum_i \sum_j s_i \cdot t_j \cdot [u^{(k)}[a_i] = v^{(k)}[b_j]] \\ &= \alpha_k \cdot \sum_{\substack{\text{equivalence} \\ \text{classes } \mathcal{C}}} \sum_{i: u^{(k)}[a_i] \in \mathcal{C}} \sum_{j: v^{(k)}[b_j] \in \mathcal{C}} s_i \cdot t_j \\ &= \alpha_k \cdot \sum_{\substack{\text{equivalence} \\ \text{classes } \mathcal{C}}} \left(\sum_{i \,:\, u^{(k)}[a_i] \in \mathcal{C}} s_i \right) \cdot \left(\sum_{j \,:\, v^{(k)}[b_j] \in \mathcal{C}} t_j \right). \end{aligned}$$

[7] We could also draw s_i, t_i uniformly and independently from $\{0, 1\}$, and appeal to another folklore PIT theorem which says that the probability of nonzero evaluation (of a nonzero degree-d) polynomial is at least $1/2^d$.

Counting the cost of multiplying α_k with the sum, it takes at most $O(n \log n) \cdot \text{poly}(d) \cdot r$ time to compute T_k, for all $k \in [r]$. The space complexity is at most $n \cdot \text{poly}(\log n, d)$, as we only need to keep a counter for $k \in [r]$ to remember the current T_k we are computing, and $r \leq 2^d$. □

The randomized algorithm given in Theorem 9 relies on polynomial identity testing, which in general we do not know how to derandomize in nearly-linear time (even in the simple setting we use). However, by a slight modification of the above proof, we can obtain a *deterministic* algorithm with a quadratic dependence on the rank r, which will be useful for proving circuit lower bounds when the matrix family is Disj.

Theorem 10 (Deterministic Algorithm for Satisfying Pairs)**.** *Suppose a family $\mathcal{M} = \{M_d\}$ of matrices, where M_d is $2^d \times 2^d$, has weak equality rank at most r with an explicit rank decomposition. Then the $\mathcal{M}$-Satisfying-Pairs problem with $|L| = |R| = n$ and dimension d can be solved in deterministic $r^2 \cdot n \cdot \text{poly}(d, \log n)$ time.*

Proof. We proceed as in the proof of Theorem 9: given two sets of n Boolean vectors $U = \{a_1, \ldots, a_n\} \subseteq \{0,1\}^d$ and $V = \{b_1, \ldots, b_n\} \subseteq \{0,1\}^d$, we want to quickly determine if there are i, j satisfying $M_d[a_i, b_j] = 1$.

Instead of the polynomial S in Theorem 9, we use a sum-of-squares trick. (This will increase the dependence on the rank r in our algorithm, but will yield a deterministic procedure.) Consider the following expression S', borrowing notation from the proof of Theorem 9:

$$S' := \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\sum_{k=1}^{r} \alpha_k E_k[a_i, b_j] \right)^2 .$$

Observe that if there is no satisfying pair then $S' = 0$, and if there is a satisfying pair then $S' > 0$. Thus it suffices to evaluate S'. We have

$$\begin{aligned} \left(\sum_{k=1}^{r} \alpha_k E_k[a_i, b_j] \right)^2 &= \sum_{k,k'} \alpha_k \cdot \alpha_{k'} \cdot E_k[a_i, b_j] \cdot E_{k'}[a_i, b_j] \\ &= \sum_{k,k'} \alpha_k \cdot \alpha_{k'} \cdot \left[u^{(k)}[a_i] = v^{(k)}[b_j] \right] \cdot \left[u^{(k')}[a_i] = v^{(k')}[b_j] \right] . \end{aligned}$$

Let us define r^2 equality matrices indexed by $(k, k') \in [r]^2$, where the (k, k')-th equality matrix has defining vectors

$$U^{(k,k')}[i] = (u^{(k)}[i], u^{(k')}[i]), \;\; V^{(k,k')}[j] = (v^{(k)}[j], v^{(k')}[j])$$

for all $i, j \in [2^d]$. Furthermore, let us define $\beta_{(k,k')} = \alpha_k \cdot \alpha_{k'}$. Then

$$\left(\sum_{k=1}^{r} \alpha_k E_k[a_i, b_j] \right)^2 = \sum_{k,k'} \beta_{(k,k')} \cdot \left[U^{(k,k')}[a_i] = V^{(k,k')}[b_k] \right] .$$

Now we can write

$$S' = \sum_{k,k' \in [r]} \beta_{(k,k')} \cdot \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left[U^{(k,k')}[a_i] = V^{(k,k')}[b_j] \right] \right)$$

and the r^2 double-sums $T_{(k,k')} = \sum_{i=1}^{n} \sum_{j=1}^{n} \left[U^{(k,k')}[a_i] = V^{(k,k')}[b_j] \right]$ can each be evaluated in $n \cdot \text{poly}(d, \log n)$ deterministic time, by sorting the $U^{(k,k')}[a_i]$'s and $V^{(k,k')}[b_j]$'s, and passing through the sorted order, analogously as in the proof of Theorem 9. □

We can now conclude Theorem 3.

Reminder of Theorem 3. *Suppose there is a subexponential function $f(d)$ such that for all d, Disj$_d$ has weak equality rank at most $f(d)$. Then for every $c \geq 1$ and every $\varepsilon > 0$,* **OV** *on n vectors in $c \log n$ dimensions can be solved in $n^{1+\varepsilon}$ deterministic time.*

Proof. Fix $c \geq 1$, so that $d = c \log n$. Let $\varepsilon > 0$ be an arbitrarily small constant in the following, and let $\varepsilon' = \varepsilon/(2c)$. Choose k to be a sufficiently large constant, so that by assumption, $\textsc{Disj}_k$ has weak equality rank at most $2^{\varepsilon' k}$. Then by Theorem 8, for $d \geq k$, we can obtain an *explicit* weak equality-rank decomposition for $\textsc{Disj}_d$, which is a linear combination of $O((2^{\varepsilon' k})^{d/k}) \leq O(2^{\varepsilon' d}) \leq O(2^{\varepsilon d/(2c)}) \leq O(n^{\varepsilon/2})$ equality matrices. Applying Theorem 10 to the rank decomposition of $\textsc{Disj}_d$ matrices, it follows that **OV** on n vectors and dimension $d = c \log n$ can be solved in deterministic time

$$n \log n \cdot n^{\varepsilon} \cdot \text{poly}(\log n) \leq n^{1+\varepsilon} \cdot \text{poly}(\log n).$$

Since $\varepsilon > 0$ can be made arbitrarily small, the theorem follows. □

4 Win-Win Circuit Lower Bounds

Given the results of the previous section, we can prove our win-win circuit lower bound:

Reminder of Corollary 1. *At least one of the following non-uniform circuit lower bounds is true:*

- E^{NP} *does not have $O(n)$-size Valiant series parallel circuits.*
- *There is an $\varepsilon > 0$ such that Boolean Inner Product on n-bit vectors does not have $2^{\varepsilon n}$-size* $\mathbf{ETHR} \circ \mathbf{ETHR}$ *circuits.*

Proof. We consider two cases.

1. Suppose for every $\varepsilon > 0$ and all sufficiently large d, $\textsc{Disj}_d$ has weak equality rank at most $2^{\varepsilon d}$. Then by Theorem 3, for every $c \geq 1$ and $\varepsilon > 0$, **OV** on n vectors in $c \log n$ dimensions can be solved in $n^{1+\varepsilon}$ deterministic time. By the fine-grained reduction from CNF-SAT to **OV** [Wil04, WY14], this implies that CNF-SAT on cn clauses and n variables can be solved in $2^{n/2+\varepsilon n}$ time, for every $c \geq 1$ and $\varepsilon > 0$. By results of [CDL+12], this algorithm for CNF-SAT implies that the satisfiability problem for Valiant series-parallel circuits of cn size and n inputs can also be solved in $2^{n/2+\varepsilon n}$ time. Finally, by results of [JMV15], the obtained SAT algorithm implies that there are functions in E^{NP} that do not have Valiant series-parallel circuits of $O(n)$. That is, bullet 1 in the theorem statement holds.

2. Suppose there is an $\varepsilon > 0$ such that for infinitely many d, $\textsc{Disj}_d$ has weak equality rank at least $2^{\varepsilon d}$. By Theorem 2, it follows that there is an $\varepsilon > 0$ so that for infinitely many d, the Boolean Inner Product function $\bigvee_{i=1}^{d}(x_i \wedge y_i)$ does not have $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits of size $2^{\varepsilon d}$. That is, bullet 2 in the theorem statement holds. □

4.1 Stronger OV Algorithms From Polynomial-Size ETHR of ETHR circuits

The second bullet in Corollary 1 is a rather strong circuit lower bound, compared to the first bullet which is a fairly weak lower bound. We consider what happens when we relax the lower bound in the second bullet to merely be super-polynomial. This improves the resulting **OV** algorithm obtained in the first bullet, but (as far as we know) it does not yet yield a better E^{NP} lower bound. The main theorem we can prove along these lines is the following.

Theorem 11. *Suppose there is a polynomial p such that for all d, the Boolean Inner Product function $\bigvee_{i=1}^{d}(x_i \wedge y_i)$ has an* $\mathbf{ETHR} \circ \mathbf{ETHR}$ *circuit of size $p(d)$. Then there is an $\alpha > 0$ such that for all $\varepsilon > 0$,* **OV** *on n vectors in $(\log n)^{1+\alpha}$ dimensions can be solved in $n^{1+\varepsilon}$ deterministic time.*

For the purposes of this section, we say that a function $f : \{0,1\}^{\star} \to \{0,1\}$ has *uniform* $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits of size $s(m)$ if there is an algorithm that, given 1^m, prints a description of a $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit on m inputs that computes f restricted to m-bit inputs, and does so in time $s(m) \cdot \text{poly}(\log s(m))$ time.

Assuming a stronger non-uniform circuit upper bound on the Boolean Inner Product function, we can obtain a slightly stronger "uniformization" of those non-uniform circuits.

Theorem 12 (Uniformization)**.** *Suppose there is a c such that all k, the Boolean Inner Product function $\bigvee_{i=1}^{k}(x_i \wedge y_i)$ has an* $\mathbf{ETHR} \circ \mathbf{ETHR}$ *circuit of size at most ck^c. Then there is an $\alpha > 0$ such that for all $\varepsilon > 0$ and for all sufficiently large n and $d = (\log n)^{1+\alpha}$, the function $\bigvee_{i=1}^{d}(x_i \wedge y_i)$ on $2d$ inputs has* uniform $\mathbf{ETHR} \circ \mathbf{ETHR}$ *circuits of size at most $n^{o(1)}$.*

Proof. Given the hypothesis of the theorem, the idea is to simply brute-force search for an $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit for Boolean Inner Product on $2k$-bit input. There are multiple ways we might do this; here is one simple way. It is known that every exact threshold function on $2k$ variables can be represented with integer weights in $[-k^k, k^k]$ [BHPS10]. Thus for a given circuit, there are $k^{O(k^2)}$ choices for the weights of a $2k$-input exact threshold function on the bottom layer [BHPS10], and there are $s^{O(s^2)}$ choices for the weights of the exact threshold function on the output layer, yielding $s^{O(s^2)} \cdot k^{O(sk^2)}$ total ways to choose the weights of our $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit. For $s(k) = ck^c$, the bound is $k^{O(k^{2c})}$. Setting $k = (\log n)^{1/(3c)}$, we have that for all sufficiently large n, the total number of choices is $n^{o(1)}$. Given such a circuit, an analogous argument as in Theorem 8 shows that we can obtain a circuit for $\bigvee_{i=1}^{d}(x_i \wedge y_i)$ of size $O((ck^c)^{d/k})$ time $(n^{o(1)} + (ck^c)^{d/k}) \cdot \text{poly}(d)$ for $k = (\log n)^{1/(3c)}$. Let $\alpha = 1/(4c)$. For $d = (\log n)^{1+\alpha}$, the running time for producing the circuit is still $n^{o(1)} \cdot \text{poly}(d)$. □

Applying Theorem 12 to the proof of Theorem 10, and using the reduction from $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits to weak equality rank Theorem 2, we obtain an improved **OV** algorithm from polynomial-size $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuits.

Theorem 13. *Suppose there is a c such that all k, the Boolean Inner Product function $\bigvee_{i=1}^{k}(x_i \wedge y_i)$ has an $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit of size at most ck^c. Then there is an $\alpha > 0$ such that* **OV** *on n vectors in $(\log n)^{1+\alpha}$ dimensions can be solved in $n^{1+o(1)}$ deterministic time.*

We believe that this connection from circuits to **OV** algorithms should be further improvable. The primary bottleneck is in the new uniformization (Theorem 12), where we brute-force a small $\mathbf{ETHR} \circ \mathbf{ETHR}$ circuit for Boolean Inner Product. We could obtain a stronger **OV** algorithm if we had a more efficient way of generating such circuits (assuming that the circuits exist non-uniformly). There are other situations where more efficient uniformization is possible: for example, Santhanam and Williams [SW13] build on Allender and Koucky [AK10] to show that if Boolean formulas can be simulated by polynomial-size non-uniform $O(1)$-depth threshold circuits of unbounded fan-in, then for every $\varepsilon > 0$, there are algorithms running in $2^{O(n^{\varepsilon})}$ time which, given a Boolean formula, output an equivalent $O(1)$-depth threshold circuit.

5 From Equality Rank to Algorithms for Counting Orthogonal Pairs

In this section, we prove that equality rank upper bounds on Disj imply non-trivial algorithms for counting the number of orthogonal pairs.

Reminder of Theorem 4. *Suppose that there is a subexponential function $f(d)$ such that for all d, Disj$_d$ has equality rank at most $f(d)$. Then for every $c \geq 1$ and every $\varepsilon > 0$, the* **number** *of orthogonal pairs (#***OV***) among n vectors in $\{0,1\}^{c \log n}$ can be counted in $n^{1+\varepsilon}$ deterministic time. (As a consequence, #k-SAT can be solved in $2^{n/2+o(n)}$ deterministic time, for all constants k.)*

The proof is analogous to that of Theorem 3, and is relatively straightforward in comparison. We first recall the relevant "uniformization" result:

Reminder of Theorem 8. *Suppose for some fixed k, r, Disj$_k$ has equality rank r. Then for all $d \geq k$, Disj$_d$ has equality rank at most $O(r^{d/k})$ with an* explicit *rank decomposition.*

Next, for a family $\mathcal{M} = \{M_d\}$ of matrices, where M_d is $2^d \times 2^d$, we define a general "counting satisfying pairs problem":

#$\mathcal{M}$-Satisfying-Pairs
Input: integer $d \geq 1$ and two sets $L, R \subseteq [2^d]$.
Output: the number of pairs $(i, j) \in L \times R$ such that $M_d[i, j] = 1$.

For the family of Disj matrices, the #Disj-Satisfying-Pairs problem is exactly the #**OV** (Counting Orthogonal Vectors) problem. Analogously to Theorem 10, one can show that any matrix family that has "small" equality rank has a fast deterministic algorithm for counting satisfying pairs.

Theorem 14 (Counting Satisfying Pairs)**.** *Suppose a family $\mathcal{M} = \{M_d\}$ of matrices, where M_d is $2^d \times 2^d$, has equality rank at most r with an explicit rank decomposition. Then the $\#\mathcal{M}$-Satisfying-Pairs problem with $|L| = |R| = n$ and dimension d can be solved in $r \cdot n \cdot \operatorname{poly}(d, \log n)$ time and $n \cdot \operatorname{poly}(\log n, d)$ space.*

Proof. Index the rows and columns of M_d by d-bit vectors. Assuming M_d has equality rank at most r, let $E_1, \ldots, E_r$ be $2^d \times 2^d$ equality matrices, and for all $k = 1, \ldots, r$ let $u^{(k)}, v^{(k)}$ be vectors of length d defining E_k. By hypothesis, there are $\operatorname{poly}(d)$-time computable $\alpha_1, \ldots, \alpha_r$ such that for all $a, b \in \{0,1\}^d$,

$$M_d[a,b] = \sum_{k=1}^{r} \alpha_k \cdot E_k[a,b] = \sum_{k=1}^{r} \alpha_k \cdot \left[u^{(k)}[a_i] = v^{(k)}[b_j]\right].$$

Now suppose we have an instance of $\#\mathcal{M}$-Satisfying-Pairs, with vectors $L = \{a_1, \ldots, a_n\} \subseteq \{0,1\}^d$ and $R = \{b_1, \ldots, b_n\} \subseteq \{0,1\}^d$. We wish to count the number of $(i,j) \in [n]^2$ such that $M_d[a_i, b_j] = 1$. Observe that, since $M[a,b] \in \{0,1\}$ for all $a, b \in \{0,1\}^d$, the expression

$$S := \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\sum_{k=1}^{r} \alpha_k E_k[a_i, b_j] \right)$$

counts exactly the number of (i,j) such that $M_d[a_i, b_j] = 1$. Rearranging the order of summation,

$$S = \sum_{k=1}^{r} \alpha_k \cdot \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left[u^{(k)}[a_i] = v^{(k)}[b_j]\right] \right),$$

and we can compute each inner sum $T_k = \alpha_k \sum_{i=1}^{n} \sum_{j=1}^{n} \left[u^{(k)}[a_i] = v^{(k)}[b_j]\right]$ in $n \cdot \operatorname{poly}(\log n, d)$ time by sorting the values of $u^{(k)}[a_i]$ and $v^{(k)}[b_j]$ as in the proof of Theorems 9 and 10, and passing through the sorted order. □

Theorem 4 directly follows from Theorem 8 and Theorem 14, mimicking the proof of Theorem 3.

The counting reduction of Theorem 14 can be used to show that other families of matrices must also have high equality rank, or the OVC (Conjecture 1) is false. For example, the $2^d \times 2^d$ Walsh-Hadamard transform (the matrix of the Inner Product Mod 2 function) cannot have equality rank $2^{o(d)}$, under the Orthogonal Vectors Conjecture:

Reminder of Theorem 5. *The Orthogonal Vectors Conjecture (Conjecture 1) implies that the inner product of n-bit vectors modulo 2 cannot be expressed by* $\mathbf{SUM} \circ \mathbf{ETHR}$ *circuits of size $2^{o(n)}$.*

Proof. In fact we show a stronger lower bound follows from OVC: namely, the matrix of the inner product mod 2 function requires exponential equality rank.

Let $\mathcal{M} = \{M_d\}$ be the family of Boolean matrices, where M_d is $2^d \times 2^d$, the rows and columns are indexed by $x, y \in \{0,1\}^d$, and $M_d(x,y) = 1$ if and only if the inner product of x and y mod 2 equals 1. Suppose $\mathcal{M}$ has equality rank at most $2^{o(d)}$. (Note this would follow, if the inner product of d-bit vectors mod 2 had a $\mathbf{SUM} \circ \mathbf{ETHR}$ circuit of size $2^{o(d)}$, by Theorem 2.) Then by an analogous uniformization argument as in Theorem 8, the family $\mathcal{M}$ has an explicit rank decomposition of $2^{o(d)}$ size, where each entry of each matrix can be computed in $\operatorname{poly}(d)$ time. In particular, by re-scaling the linear combination of equality matrices so that they are $1/-1$ valued rather than $0/1$ valued, then the product of d/k copies of the linear combination of equality matrices for M_k will compute the XOR of d/k copies of M_k, rather than the AND of d/k copies (as was needed for $\textsc{Disj}_d$). More precisely, suppose we replace all 1-entries in M_d with -1, and replace all 0-entries with 1, yielding the Walsh-Hadamard matrix H_d. Then the (d/k)-th Kronecker power of H_k is precisely H_d, i.e.,

$$H_d = (H_k)^{\otimes d/k}.$$

(This is a standard property of the Walsh-Hadamard transform.) Hence we can "hard-code" (or brute-force) an equality rank decomposition of M_k for constant (or small) k, translate the decomposition into one for H_k by adding one more matrix to the decomposition, and use H_k to obtain a uniform rank decomposition of H_d of size $2^{o(d)}$, precisely as was done in Theorem 8. Finally, we can translate an equality rank decomposition for H_d into one for M_d, by adding one more matrix to the decomposition.

Applying Theorem 14 on the family $\mathcal{M}$, we can thereby count the number of pairs of $(c \log n)$-dimensional vectors with *odd inner product* in $n^{1+o(1)}$ time, for any constant $c \geq 1$. Finally, this counting algorithm implies that the original $\#$**OV** problem can also be solved under the same parameters ($d = c \log n$ for any constant $c \geq 1$) in $n^{1+o(1)}$ time ([Wil18a], Corollary 6). $\square$

5.1 Extension to Low-Rank Rigidity Decompositions

To conclude this section, we briefly point out one more extension of our method which may prove interesting for future work. For the **OV** algorithm consequence of Theorem 3 and Theorem 4, we do not require an *exact* equality-rank decomposition for $\textsc{Disj}_d$. Indeed, it would suffice if for all $\varepsilon > 0$ and all sufficiently large d that there is a *uniformly-computable* linear combination of $2^{\varepsilon d}$ equality matrices, along with a $2^d \times 2^d$ *sparse matrix*, the sum of which equals $\textsc{Disj}_d$. In particular, the only requirement on the sparse matrix needed is that in each row, there are at most $2^{\varepsilon d}$ nonzeroes. That is, we only require that $\textsc{Disj}_d$ has low equality rank *rigidity*: that $2^{\varepsilon d}$ entries in each row of $\textsc{Disj}_d$ can be perturbed to reduce its equality rank down to $2^{\varepsilon d}$, and that given an $x \in \{0,1\}^d$, the entries in the x-th row can be computed in $2^{\varepsilon d + o(d)}$ time.

To see why this is true, recall that when we solve **OV**, we are specifying n rows and n columns of $\textsc{Disj}_d$, and we are trying to detect if any entry in the subrectangle specified by the rows and columns is 0. By our reduction to sorting (Theorem 4), given an equality-rank decomposition, we can compute the sum over all n rows and all n columns of all the entries in the matrix specified by that rank decomposition. If we can also efficiently compute the relevant $O(n \cdot 2^{\varepsilon d})$ nonzero entries of the sparse matrix as well, then we can add their contribution to the sum over all n rows and n columns, exactly calculating the sum of ones in the given subrectangle of $\textsc{Disj}_d$.

6 Explicit Decompositions of the Disjointness Matrix

We now present our new combinatorial algorithms for **OV**, proving the following theorems claimed in the introduction.

Reminder of Theorem 6. *The* weak equality rank *of* $2^6 \times 2^6$ *Disjointness is (at most)* 6. *Applying Theorems 8 and 9 directly, there is a randomized algorithm for* **OV** *that runs in* $\tilde{O}(n \cdot 6^{d/6}) \leq \tilde{O}(n \cdot 1.35^d)$ *time and* $n \cdot \text{poly}(\log n, d)$ *space.*

Reminder of Theorem 7. *The* equality rank *of* $2^5 \times 2^5$ *Disjointness is (at most)* 5. *Applying Theorem 8 and Theorem 14 directly, there is a deterministic algorithm for counting* **OV** *pairs (a.k.a.* $\#$**OV**) *that runs in* $\tilde{O}(n \cdot 5^{d/5}) \leq \tilde{O}(n \cdot 1.38^d)$ *time and* $n \cdot \text{poly}(\log n, d)$ *space.*

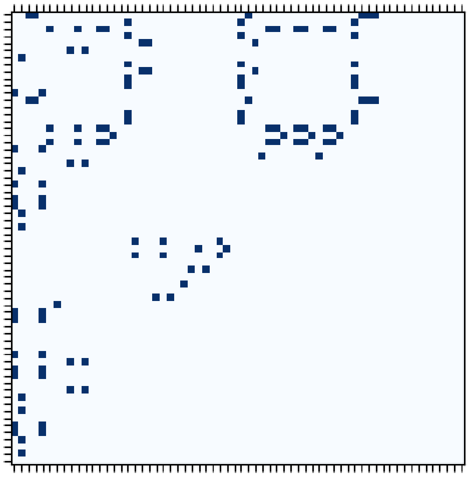
To establish these theorems, we simply provide the rank decompositions found using SAT and SMT solvers. The claimed randomized algorithms follow directly from the statements of Theorems 8,9, and 14.

6.1 Weak Equality-Rank Decomposition for Disjointness

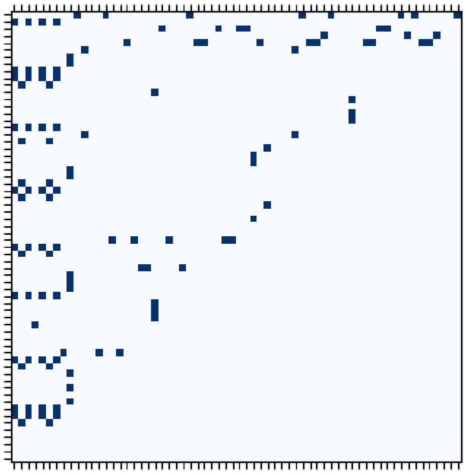
Our decomposition for the $2^6 \times 2^6$ Disjointness matrix is in fact an *OR* of 6 equality matrices, whose ones cover exactly the ones of the Disjointness matrix. Since Jukna [Juk06] proved that such an OR must have at least 1.08^d equality matrices in general (see Appendix A for a self-contained exposition), in general we cannot expect an OR of equality matrices to give a subexponential-size decomposition of Disjointness. We only searched for an OR of equality matrices because the task is significantly easier for SAT/SMT solvers (compared to the search for an arbitrary linear combination of equality matrices, which is nonzero in exactly those entries where Disjointness is 1).

For each of the six equality matrices, we give the pair of defining vectors for the matrix, as well as images of the matrices. (A light cell indicates 0, while a dark cell indicates 1.)

1.
```
u = [23, 41, 18, 41, 25, 52, 53, 41, 25, 41, 41, 68, 23, 105, 41, 41, 18, 17, 18, 68,
89, 52, 53, 22, 68, 116, 68, 68, 53, 87, 53, 99, 90, 97, 90, 47, 13, 65, 94, 54, 26,
20, 68, 68, 57, 87, 48, 64, 68, 52, 68, 68, 113, 52, 53, 4, 53, 79, 68, 68, 53, 87,
53, 4]
v = [68, 53, 23, 23, 68, 18, 20, 121, 52, 18, 52, 55, 18, 18, 17, 122, 41, 90, 25, 25,
26, 90, 26, 84, 94, 13, 97, 13, 43, 90, 97, 75, 41, 23, 25, 89, 18, 18, 17, 55, 18,
18, 17, 89, 18, 18, 17, 15, 41, 23, 23, 23, 9, 82, 28, 121, 0, 21, 73, 55, 50, 75, 30,
38]
```

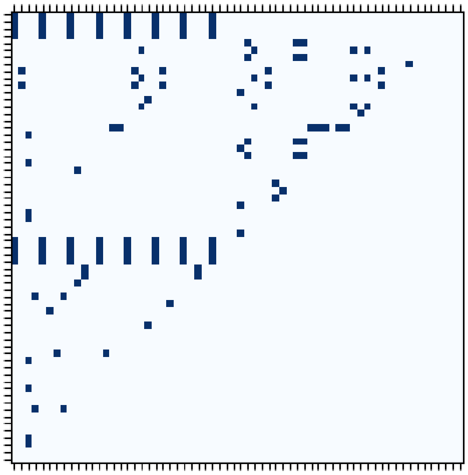


2.
   ```
   u = [111, 115, 30, 103, 67, 86, 70, 70, 115, 115, 127, 14, 71, 62, 71, 71, 115, 86,
   127, 113, 83, 83, 70, 70, 127, 115, 127, 113, 3, 83, 78, 47, 107, 115, 127, 39, 99,
   70, 70, 70, 115, 14, 14, 14, 17, 3, 94, 98, 123, 115, 127, 70, 27, 70, 94, 70, 115,
   115, 127, 31, 41, 19, 124, 48]
   v = [115, 127, 115, 17, 115, 127, 115, 123, 70, 111, 86, 49, 123, 111, 107, 123, 67,
   107, 99, 99, 14, 30, 107, 118, 99, 111, 67, 67, 126, 30, 107, 107, 30, 30, 83, 67, 113,
   87, 63, 102, 86, 111, 67, 67, 103, 111, 122, 108, 71, 95, 67, 67, 30, 30, 20, 111, 103,
   111, 67, 67, 103, 4, 51, 111]
   ```

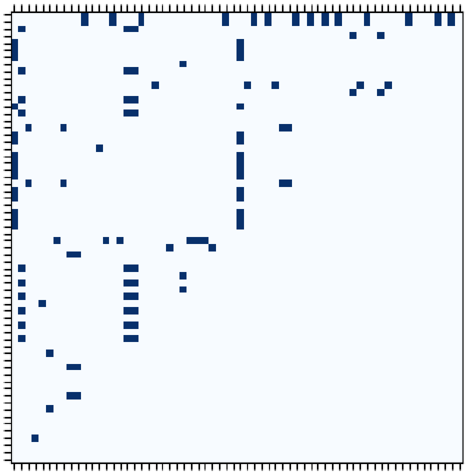


3.
   ```
   u = [112, 112, 112, 112, 50, 34, 50, 32, 59, 34, 59, 96, 58, 34, 64, 95, 53, 99, 50,
   96, 50, 99, 43, 103, 115, 51, 115, 96, 99, 99, 5, 96, 112, 112, 112, 112, 61, 61, 43,
   103, 119, 83, 91, 20, 58, 23, 27, 81, 78, 99, 110, 48, 48, 99, 14, 48, 119, 27, 48,
   49, 99, 99, 85, 109]
   ```

v = [112, 59, 99, 119, 112, 91, 78, 119, 112, 43, 61, 21, 112, 78, 53, 53, 112, 59, 34, 58, 112, 59, 83, 0, 112, 41, 61, 41, 112, 16, 93, 69, 96, 50, 34, 80, 59, 115, 51, 123, 50, 50, 53, 53, 53, 22, 53, 53, 34, 64, 34, 87, 59, 88, 123, 57, 32, 1, 6, 55, 111, 97, 56, 90]

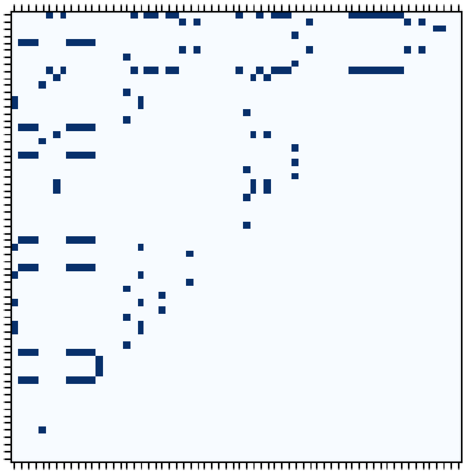


4. u = [68, 68, 112, 67, 65, 65, 65, 44, 112, 23, 48, 67, 112, 65, 112, 36, 1, 65, 65, 6, 65, 65, 65, 65, 1, 65, 65, 36, 65, 65, 65, 0, 49, 87, 35, 117, 112, 44, 112, 44, 112, 61, 112, 121, 112, 116, 112, 36, 57, 51, 35, 110, 10, 10, 35, 116, 57, 42, 121, 106, 26, 102, 2, 2]
   v = [65, 112, 1, 26, 61, 57, 49, 1, 35, 35, 68, 4, 6, 49, 68, 49, 112, 112, 68, 32, 48, 103, 87, 13, 44, 49, 49, 49, 87, 91, 68, 11, 65, 48, 68, 100, 68, 48, 1, 1, 68, 69, 68, 4, 68, 24, 68, 63, 67, 48, 68, 20, 67, 48, 17, 83, 68, 66, 115, 104, 68, 122, 68, 55]

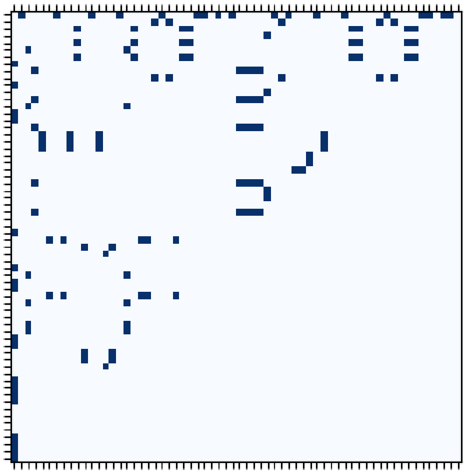


5. u = [86, 63, 123, 119, 110, 63, 103, 119, 86, 87, 127, 103, 90, 90, 54, 103, 110, 87, 127, 119, 110, 119, 54, 119, 87, 87, 54, 55, 94, 31, 54, 94, 110, 90, 71, 78, 110, 90, 71, 103, 91, 90, 91, 103, 90, 90, 62, 103, 110, 95, 95, 95, 110, 30, 30, 46, 92, 126, 26, 127, 78, 94, 46, 96]
   v = [90, 110, 110, 110, 127, 86, 87, 86, 110, 110, 110, 110, 95, 102, 15, 53, 103, 86,

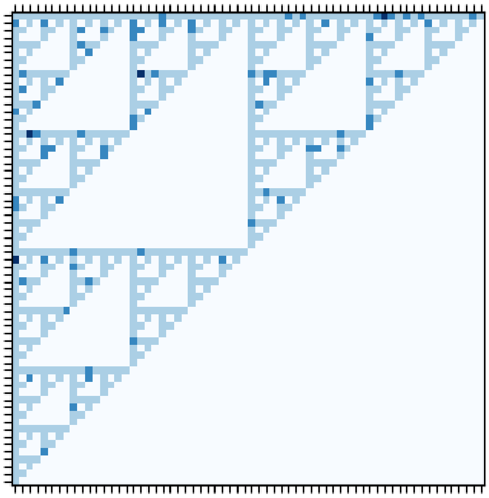
`90, 86, 86, 91, 86, 86, 63, 71, 63, 39, 34, 40, 70, 0, 86, 54, 87, 86, 87, 86, 86, 86, 119, 102, 63, 34, 38, 104, 79, 0, 86, 86, 86, 86, 86, 86, 86, 86, 63, 23, 63, 57, 123, 123, 109, 73]`



6. `u = [24, 61, 17, 49, 17, 45, 17, 36, 18, 61, 36, 49, 18, 45, 36, 36, 18, 37, 37, 37, 16, 16, 53, 21, 18, 49, 49, 109, 18, 113, 68, 36, 121, 77, 125, 41, 36, 45, 36, 36, 121, 45, 8, 73, 45, 45, 36, 36, 77, 77, 125, 93, 36, 36, 36, 36, 109, 97, 88, 96, 36, 36, 36, 36]`
   `v = [36, 24, 45, 18, 37, 121, 24, 121, 37, 17, 77, 24, 37, 125, 77, 24, 45, 17, 121, 121, 61, 24, 61, 121, 17, 17, 24, 24, 52, 24, 13, 24, 18, 18, 18, 18, 49, 24, 61, 24, 53, 53, 16, 24, 37, 5, 29, 24, 17, 17, 9, 51, 61, 24, 61, 51, 17, 17, 24, 24, 67, 24, 24, 74]`



We observe that the OR of these 6 matrices yields the $2^6 \times 2^6$ Disjointness matrix:

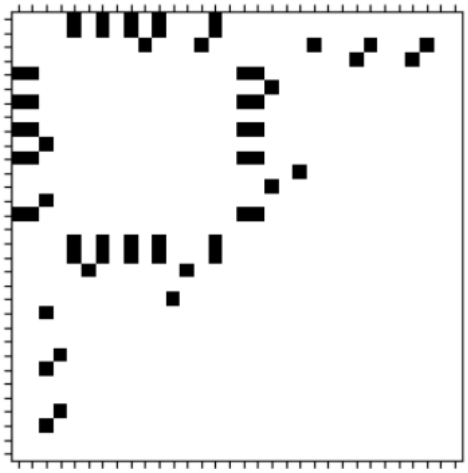


6.2 Equality-Rank Decomposition for Disjointness

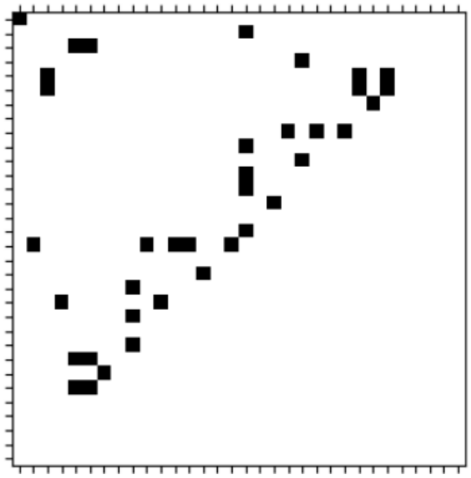
Our decomposition for the $2^5 \times 2^5$ Disjointness matrix is in fact an *disjoint-OR* of 5 equality matrices, whose ones form a *partition* the ones of the Disjointness matrix. As in the previous subsection, we searched for a disjoint-OR of equality matrices because the task is far easier for SAT/SMT solvers.

Here are the defining vectors, as well as explicit figures of the matrices. (A white cell indicates 0, while a black cell indicates 1.)

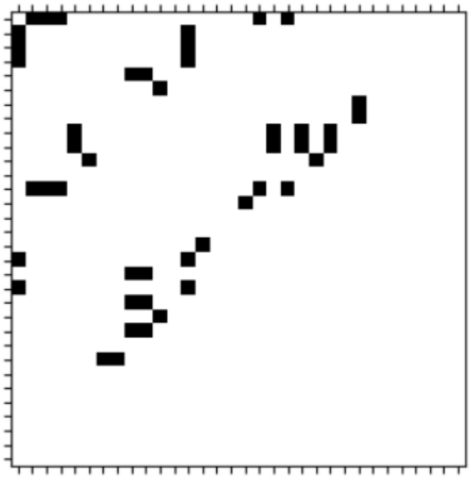
1. u = [1, 1, 3, 5, 16, 10, 16, 9, 16, 14, 16, 11, 10, 14, 16, 2, 1, 1, 7, 2, 6, 14, 4, 4, 12, 14, 4, 9, 12, 14, 2, 9]
   v = [16, 16, 14, 12, 1, 7, 1, 8, 1, 3, 1, 6, 7, 3, 1, 15, 16, 16, 10, 15, 11, 3, 13, 13, 5, 3, 13, 8, 5, 3, 15, 8]



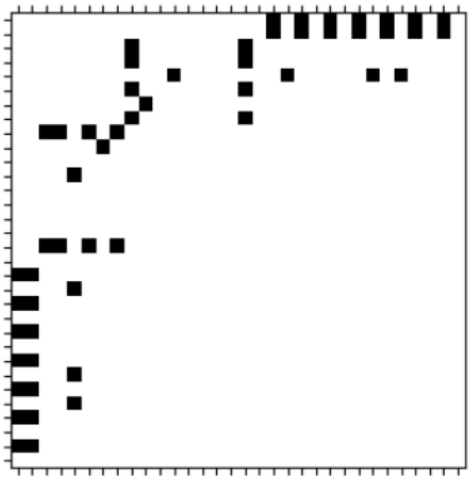
2. u = [1, 9, 6, 5, 11, 11, 3, 4, 7, 9, 5, 9, 9, 15, 4, 9, 8, 4, 2, 10, 12, 10, 4, 10, 6, 14, 6, 4, 4, 4, 4, 4]
   v = [1, 8, 11, 12, 6, 6, 14, 13, 10, 8, 12, 8, 8, 2, 13, 8, 9, 13, 15, 7, 5, 7, 13, 7, 11, 3, 11, 13, 13, 13, 13, 13]



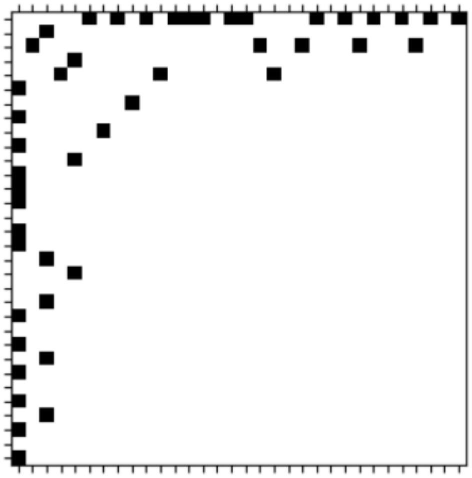
```
3. u = [1, 16, 16, 16, 3, 12, 7, 7, 14, 14, 5, 13, 1, 9, 13, 2, 8, 16, 3, 16, 3, 12, 3,
   11, 10, 2, 13, 11, 11, 13, 11, 13]
   v = [16, 1, 1, 1, 14, 5, 10, 10, 3, 3, 12, 4, 16, 8, 4, 15, 9, 1, 14, 1, 14, 5, 14,
   6, 7, 15, 4, 6, 6, 4, 6, 4]
```



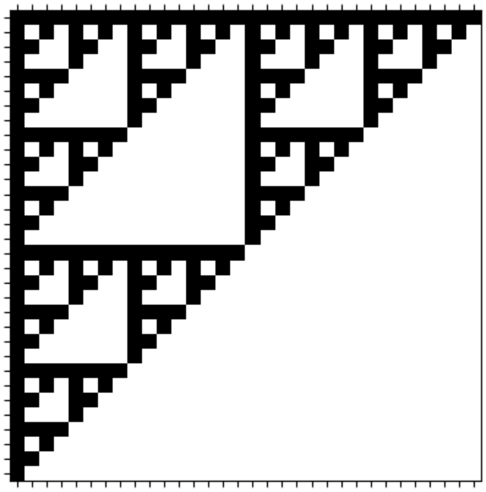
```
4. u = [1, 1, 3, 3, 15, 3, 7, 3, 14, 10, 9, 2, 11, 5, 11, 4, 14, 9, 16, 2, 16, 4, 16, 4,
   16, 2, 16, 2, 16, 11, 16, 9]
   v = [16, 16, 14, 14, 2, 14, 10, 14, 3, 7, 8, 15, 6, 12, 6, 13, 3, 8, 1, 15, 1, 13, 1,
   13, 1, 15, 1, 15, 1, 6, 1, 8]
```



5. `u = [1, 11, 6, 3, 14, 16, 9, 16, 8, 16, 3, 16, 16, 16, 12, 16, 16, 11, 3, 15, 11, 16, 13, 16, 11, 16, 12, 16, 11, 16, 12, 16]`
   `v = [16, 6, 11, 14, 3, 1, 8, 1, 9, 1, 14, 1, 1, 1, 5, 1, 1, 6, 14, 2, 6, 1, 4, 1, 6, 1, 5, 1, 6, 1, 5, 1]`



Observe that these matrices are *disjoint* in that for every pair of matrices in the above, no cell (i, j) is black (i.e., 1) in both matrices. The OR of these 5 matrices yields the $2^5 \times 2^5$ Disjointness matrix:



7 Discussion

We have shown how non-uniform circuit lower bounds are implied by uniform conjectures such as the Orthogonal Vectors Conjecture. We have also seen how the notions of "weak equality rank" and "equality rank" have close connections to algorithms for Orthogonal Vectors. Let us conclude with a discussion on other rank notions that could also lead to faster OV algorithms.

A Little More Skepticism of the OV Conjecture. One may view our algorithms for solving OV as reductions from OV to a c^d-size collection of $\#2SUM$ instances, each of which can be solved in $n \cdot \text{poly}(\log n)$ time. One could choose other target problems to reduce to, which would result in different algorithmic approaches (and different win-win lower bounds). We studied other rank notions beyond those described in this paper, but we have chosen not to include formal results on them, because they do not seem to enjoy a "uniformization" lemma. Here are two:

1. One could study "less-than-or-equal-to" (LEQ) matrices, where the defining vectors u, v are like that of equality matrices, but we put a 1 in the i, j if and only if $u[i] \leq v[j]$ (rather than $u[i] = v[j]$). It is not hard to show that a sum of LEQ matrices can be efficiently expressed as a sum of equality matrices, so if there was a $2^{o(d)}$-size *uniformly computable* sum of LEQ matrices computing $2^d \times 2^d$ Disjointness, we would still refute the OV Conjecture. Our SAT/SMT searches found interesting decompositions of constant-size Disjointness matrices into sums of LEQ matrices, but as far as we can tell, the corresponding uniformization lemma for such matrices is too weak to yield better OV algorithms.

2. We also considered sums of what we call "ReLU matrices": these matrices have defining vectors u, v over the integers, where the (i, j) component of the matrix is defined to be $\max\{0, u[i] + v[j]\}$. It is not hard to show that sums of such ReLU matrices can simulate the truth tables of depth-two neural networks with a ReLU activation function, similarly to how weak equality rank captures the truth tables of depth-two exact threshold circuits. As in the case of LEQ matrices, we can prove that an *explicit* (uniformly computable) subexponential-size sums of ReLU matrices computing Disjointness would also refute the OV Conjecture.

In summary, the OV Conjecture implies a variety of *uniform* circuit lower bounds, but this is perhaps to be expected (a uniform conjecture ought to imply uniform circuit lower bounds of some kind). We hope that future work will better clarify the reach of the OV Conjecture with respect to circuit complexity.

Acknowledgments. Much of the code that produced the results of Section 6.1 and 6.2 was written (and run) while I was visiting the Simons Institute in Fall 2023 for the program on Logic and Algorithms in Database Theory and AI; I thank the Simons Institute for their hospitality. I am grateful to Lianna Hambardzumyan, Rahul Ilango, Ce Jin, and

Morgan Shirley for useful discussions on this work. I am also grateful to Shyan Akmal and the FOCS referees for very useful comments which improved the writing.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[ABDN18] Amir Abboud, Karl Bringmann, Holger Dell, and Jesper Nederlof. More consequences of falsifying SETH and the orthogonal vectors conjecture. In *STOC*, pages 253–266. ACM, 2018.

[ABH$^+$16] Amir Abboud, Arturs Backurs, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Or Zamir. Subtree isomorphism revisited. In *SODA*, pages 1256–1271, 2016.

[ABV15] Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *FOCS*, pages 59–78, 2015.

[AC19] Josh Alman and Lijie Chen. Efficient construction of rigid matrices using an NP oracle. In *FOCS*, pages 1034–1055. IEEE Computer Society, 2019.

[ACL$^+$20] Scott Aaronson, Nai-Hui Chia, Han-Hsuan Lin, Chunhao Wang, and Ruizhe Zhang. On the quantum complexity of closest pair and related problems. In *CCC*, volume 169 of *LIPIcs*, pages 16:1–16:43. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[ACW16] Josh Alman, Timothy M. Chan, and R. Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *FOCS*, pages 467–476, 2016.

[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3), 2010.

[Alm21] Josh Alman. Kronecker products, low-depth circuits, and matrix rigidity. In *STOC*, pages 772–785. ACM, 2021.

[Alo86] Noga Alon. Covering graphs by the minimum number of equivalence relations. *Comb.*, 6(3):201–206, 1986.

[AM05] Kazuyuki Amano and Akira Maruoka. On the complexity of depth-2 circuits with threshold gates. In *MFCS*, pages 107–118, 2005.

[Ama10] Kazuyuki Amano. Researching the complexity of boolean functions with computers. *Bull. EATCS*, 101:64–91, 2010.

[Ama20] Kazuyuki Amano. On the size of depth-two threshold circuits for the inner product mod 2 function. In *Language and Automata Theory and Applications - LATA*, volume 12038 of *Lecture Notes in Computer Science*, pages 235–247. Springer, 2020.

[APRS16] Thomas Dybdahl Ahle, Rasmus Pagh, Ilya P. Razenshteyn, and Francesco Silvestri. On the complexity of inner product similarity join. In *PODS*, pages 151–164, 2016.

[AVW14] Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *ICALP*, pages 39–51, 2014.

[AVW16] Amir Abboud, Virginia Vassilevska Williams, and Joshua Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *SODA*, pages 377–391, 2016.

[AWY15] Amir Abboud, Richard Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *SODA*, pages 218–230, 2015.

[AY24] Daniel Avraham and Amir Yehudayoff. On blocky ranks of matrices. *Comput. Complex.*, 33(1):2, 2024.

[BBK$^+$16] Kevin Buchin, Maike Buchin, Maximilian Konzack, Wolfgang Mulzer, and André Schulz. Fine-grained analysis of problems on curves. In *EuroCG, Lugano, Switzerland*, 2016.

[BBM$^+$21] Alexander R. Block, Simina Brânzei, Hemanta K. Maji, Himanshi K. Mehta, Tamalika Mukherjee, and Hai H. Nguyen. P_4-free partition and cover numbers & applications. In *2nd Conference on Information-Theoretic Cryptography, ITC*, volume 199 of *LIPIcs*, pages 16:1–16:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[BFFH20] Armin Biere, Katalin Fazekas, Mathias Fleury, and Maximillian Heisinger. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. In *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, volume B-2020-1 of *Department of Computer Science Report Series B*, pages 51–53. University of Helsinki, 2020.

[BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory (preliminary version). In *FOCS*, pages 337–347. IEEE Computer Society, 1986.

[BHPS10] László Babai, Kristoffer Arnsfelt Hansen, Vladimir V. Podolskii, and Xiaoming Sun. Weights of exact threshold functions. In *MFCS*, volume 6281 of *Lecture Notes in Computer Science*, pages 66–77. Springer, 2010.

[BI15] Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *STOC*, pages 51–58, 2015.

[BI16] Arturs Backurs and Piotr Indyk. Which regular expression patterns are hard to match? In *FOCS*, pages 457–466, 2016.

[BK15] Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *FOCS*, pages 79–97, 2015.

[BM16] Karl Bringmann and Wolfgang Mulzer. Approximability of the discrete Fréchet distance. *JoCG*, 7(2):46–76, 2016.

[BPS21] Harry Buhrman, Subhasree Patro, and Florian Speelman. A framework of quantum strong exponential-time hypotheses. In *STACS*, volume 187 of *LIPIcs*, pages 19:1–19:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[Bri14] Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In *FOCS*, pages 661–670, 2014.

[BRSV17] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Average-case fine-grained hardness. *IACR Cryptology ePrint Archive*, 2017:202, 2017.

[CDHL16] Krishnendu Chatterjee, Wolfgang Dvorák, Monika Henzinger, and Veronika Loitzenbauer. Model and objective separation with conditional lower bounds: Disjunction is harder than conjunction. In *LICS*, pages 197–206, 2016.

[CDL$^+$12] Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. In *CCC*, pages 74–84, 2012.

[CGR16] Massimo Cairo, Roberto Grossi, and Romeo Rizzi. New bounds for approximating extremal distances in undirected graphs. In *SODA*, pages 363–376, 2016.

[Che19] Lijie Chen. Non-deterministic quasi-polynomial time is average-case hard for ACC circuits. In *FOCS*, pages 1281–1304, 2019.

[CIP09] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Parameterized and Exact Computation*, pages 75–85. Springer, 2009.

[CLV19] Arkadev Chattopadhyay, Shachar Lovett, and Marc Vinyals. Equality alone does not simulate randomness. In *CCC*, volume 137 of *LIPIcs*, pages 14:1–14:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[CSS16] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-case lower bounds and satisfiability algorithms for small threshold circuits. In *CCC*, pages 1:1–1:35, 2016.

[CST17] Pairwise comparison of bit vectors. https://cstheory.stackexchange.com/questions/37361/pairwise-comparison-of-bit-vectors, January 20, 2017.

[CW19] Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. In *SODA*, pages 21–40. SIAM, 2019.

[CW21] Timothy M. Chan and R. Ryan Williams. Deterministic apsp, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. *ACM Trans. Algorithms*, 17(1):2:1–2:14, 2021.

[DL78] Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):192–195, 1978.

[dMB08] Leonardo Mendonça de Moura and Nikolaj S. Bjørner. Z3: an efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

[DY22] Benjamin E. Diamond and Amir Yehudayoff. Explicit exponential lower bounds for exact hyperplane covers. *Discret. Math.*, 345(11):113114, 2022.

[ED16] Jacob Evald and Søren Dahlgaard. Tight hardness results for distance and centrality problems in constant degree graphs. *CoRR*, abs/1609.08403, 2016.

[GIKW17] Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and R. Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. In *SODA*, pages 2162–2181, 2017.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219. ACM, 1996.

[HHH23] Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. Dimension-free bounds and structural results in communication complexity. *Israel Journal of Mathematics*, 253(2):555–616, 2023.

[HMP$^+$93] 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.

[HP10] Kristoffer Arnsfelt Hansen and Vladimir V. Podolskii. Exact threshold circuits. In *CCC*, pages 270–279. IEEE Computer Society, 2010.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[IR16] Costas S. Iliopoulos and Jakub Radoszewski. Truly subquadratic-time extension queries and periodicity detection in strings with uncertainties. In *27th Annual Symposium on Combinatorial Pattern Matching, CPM 2016, June 27-29, 2016, Tel Aviv, Israel*, pages 8:1–8:12, 2016.

[JMV15] Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local reductions. In *ICALP*, pages 749–760, 2015.

[Juk06] Stasys Jukna. On graph complexity. *Comb. Probab. Comput.*, 15(6):855–876, 2006.

[KPS17] Marvin Künnemann, Ramamohan Paturi, and Stefan Schneider. On the fine-grained complexity of one-dimensional dynamic programming. In *ICALP*, volume 80 of *LIPIcs*, pages 21:1–21:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[KS92] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM J. Discr. Math.*, 5(4):545–557, 1992.

[KV11] Bernhard H Korte and Jens Vygen. *Combinatorial optimization*, volume Algorithms and Combinatorics 21. Springer, 2011.

[KW16] Daniel M. Kane and Ryan Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *STOC*, pages 633–643. ACM, 2016.

[NW21] Jesper Nederlof and Karol Wegrzycki. Improving schroeppel and shamir's algorithm for subset sum via orthogonal vectors. In *STOC*, pages 1670–1683. ACM, 2021.

[PF79] Nicholas Pippenger and Michael J. Fischer. Relations among complexity measures. *J. ACM*, 26(2):361–381, 1979.

[PR94] Pavel Pudlák and Vojtech Rödl. Some combinatorial-algebraic problems from complexity theory. *Discret. Math.*, 136(1-3):253–279, 1994.

[PSS23] Toniann Pitassi, Morgan Shirley, and Adi Shraibman. The strength of equality oracles in communication. In *ITCS*, volume 251 of *LIPIcs*, pages 89:1–89:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[ROS94] Vwani P. Roychowdhury, Alon Orlitsky, and Kai-Yeung Siu. Lower bounds on threshold and related circuits via communication complexity. *IEEE Transactions on Information Theory*, 40(2):467–474, 1994.

[RV13] Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *STOC*, pages 515–524, 2013.

[Sch80] Jacob Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *JACM*, 27(4):701–717, 1980.

[SE05] Niklas Söorensson and Niklas Een. Minisat v1.13 – a SAT solver with conflict-clause minimization. *Proc. Theory and Applications of Satisfiability Testing*, 2005.

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969.

[SW13] Rahul Santhanam and Ryan Williams. On medium-uniformity and circuit lower bounds. In *CCC*, pages 15–23, 2013.

[Tam16] Suguru Tamaki. A satisfiability algorithm for depth two circuits with a sub-quadratic number of symmetric and threshold gates. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:100, 2016.

[TvL84] Robert Endre Tarjan and Jan van Leeuwen. Worst-case analysis of set union algorithms. *J. ACM*, 31(2):245–281, 1984.

[Val77] L. G. Valiant. Graph-theoretic arguments in low-level complexity. In *MFCS*, volume 53 of *LNCS*, pages 162–176, Tatranská Lomnica, Czechoslovakia, September 1977. Springer.

[Vas18] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the International Congress of Mathematicians (ICM)*, pages 3447–3487. World Scientific, 2018.

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999.

[Wil18a] R. Ryan Williams. Counting solutions to polynomial systems via reductions. In *1st Symposium on Simplicity in Algorithms, SOSA*, volume 61 of *OASIcs*, pages 6:1–6:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[Wil18b] R. Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. *Theory Comput.*, 14(1):1–25, 2018.

[Wil18c] Richard Ryan Williams. Limits on representing boolean functions by linear combinations of simple functions: Thresholds, ReLUs, and low-degree polynomials. In *CCC*, volume 102 of *LIPIcs*, pages 6:1–6:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[Wil11] Ryan Williams. Nonuniform ACC circuit lower bounds. *JACM*, 61(1):2, 2014. See also CCC'11.

[Wil04] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. See also ICALP'04.

[Wil10] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal on Computing*, 42(3):1218–1244, 2013. See also STOC'10.

[WY14] Ryan Williams and Huacheng Yu. Finding orthogonal vectors in discrete structures. In *SODA*, pages 1867–1877, 2014.

[Zip79] R. E. Zippel. Probabilistic algorithms for sparse polynomials. In *International Symposium on Symbolic and Algebraic Manipulation*, pages 216–226, 1979.

A Simplified version of Jukna's argument

Here we give a self-contained proof that no OR of a "small" number of equality matrices can represent the Disjointness matrix.

Theorem 15 ([Juk06]). *Any OR of t equality matrices computing* $\textsc{Disj}_d$ *requires* $t \geq \left(\frac{3}{2^{3/2}}\right)^d \geq 1.08^d$.

Proof. WLOG, let d be even. First, we observe that the $2^d \times 2^d$ Disjointness matrix does not contain a 1-rectangle of size $2^{d/2+1} \times 2^{d/2+1}$. That is, for any collection $\mathcal{D}$ of $2^{d/2}+1$ subsets of $[d]$, and any collection $\mathcal{C}$ of $2^{d/2}+1$ subsets of $[d]$, there must be a set $C \in \mathcal{C}$ and $D \in \mathcal{D}$ which intersect. Contrapositively, in order for every pair of sets in $\mathcal{C}$ and $\mathcal{D}$ to be disjoint, it must be that all sets in $\mathcal{C}$ are subsets of some $S \subseteq [d]$ and all sets in $\mathcal{D}$ are subsets of some $T \subseteq [d]$ such that $S \cap T = \varnothing$. A collection $\mathcal{C}$ of subsets of S must have cardinality at most $2^{|S|}$, and similarly $\mathcal{D}$ must have at most $2^{|T|}$ subsets. If $S \cap T = \varnothing$ then we must have $|S| + |D| \leq d$. Therefore if $\mathcal{C}$ and $\mathcal{D}$ both have $2^{d/2}+1$ subsets, then such S and T cannot exist, so some pair of sets in $\mathcal{C}$ and $\mathcal{D}$ are not disjoint.

Since Disjointness does not contain a 1-rectangle of size $(2^{d/2}+1) \times (2^{d/2}+1)$, any equality matrix whose 1-entries are contained in the 1-entries of Disjointness must also not contain such a rectangle. Therefore, in each equality matrix $M_1, \ldots, M_t$ in our OR of equality matrices for Disjointness, each number appearing in the defining vectors for M_i can only appear either at most $2^{d/2}$ times in the left defining vector, or at most $2^{d/2}$ times in the right defining vector. Thus, the maximum number of 1-entries of Disjointness that could possibly be covered by an equality matrix is at most $(2^d/2^{d/2}) \cdot (2^{d/2})^2 = 2^{3d/2}$: this would be achieved by having $2^d/2^{d/2}$ separate numbers, each of which appear $2^{d/2}$ times in both defining vectors.

There are 3^d ones in the Disjointness matrix we have to cover. So our OR must contain at least $3^d/2^{3d/2}$ equality matrices to cover them all. □