Conditional Complexity Hardness: Monotone Circuit Size, Matrix Rigidity, and Tensor Rank

Nikolai Chukhin
Neapolis University Pafos, Cyprus
JetBrains Research, Pafos, Cyprus

Alexander S. Kulikov
JetBrains Research, Pafos, Cyprus

Ivan Mihajlin
JetBrains Research, Pafos, Cyprus

Arina Smirnova
Neapolis University Pafos, Cyprus

Abstract

Proving complexity lower bounds remains a challenging task: currently, we only know how to prove conditional uniform (algorithm) lower bounds and nonuniform (circuit) lower bounds in restricted circuit models. About a decade ago, Williams (STOC 2010) showed how to derive nonuniform lower bounds from uniform upper bounds: roughly, by designing a fast algorithm for checking satisfiability of circuits, one gets a lower bound for this circuit class. Since then, a number of results of this kind have been proved. For example, Jahanjou et al. (ICALP 2015) and Carmosino et al. (ITCS 2016) proved that if NSETH fails, then E^{NP} has series-parallel circuit size $\omega(n)$.

One can also derive nonuniform lower bounds from nondeterministic uniform lower bounds. Perhaps the most well-known example is the Karp–Lipton theorem (STOC 1980): if $\Sigma_2 \neq \Pi_2$, then $\mathsf{NP} \not\subset \mathsf{P/poly}$. Some recent examples include the following. Nederlof (STOC 2020) proved a lower bound on the matrix multiplication tensor rank under an assumption that TSP cannot be solved faster than in 2^n time. Belova et al. (SODA 2024) proved that there exists an explicit polynomial family of arithmetic circuit size $\Omega(n^\delta)$, for any $\delta > 0$, assuming that MAX-3-SAT cannot be solved faster than in 2^n nondeterministic time. Williams (FOCS 2024) proved an exponential lower bound for $\mathsf{ETHR} \circ \mathsf{ETHR}$ circuits under the *Orthogonal Vectors* conjecture. Whereas all the lower bounds above are proved under strong assumptions that might eventually be refuted, the revealed connections are of great interest and may still give further insights: one may be able to weaken the used assumptions or to construct generators from other fine-grained reductions.

In this paper, we continue developing this line of research and show how uniform nondeterministic lower bounds can be used to construct generators of various types of combinatorial objects that are notoriously hard to analyze: Boolean functions of high circuit size, matrices of high rigidity, and tensors of high rank. Specifically, we prove the following.

- If, for some ε and k, k-SAT cannot be solved in input-oblivious co-nondeterministic time $O(2^{(1/2+\varepsilon)n})$, then there exists a monotone Boolean function family in coNP of monotone circuit size $2^{\Omega(n/\log n)}$. Combining this with the result above, we get win-win circuit lower bounds: either E^{NP} requires series-parallel circuits of size $\omega(n)$ or coNP requires monotone circuits of size $2^{\Omega(n/\log n)}$.
- If, for all $\varepsilon > 0$, MAX-3-SAT cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, then there exist small families of matrices with rigidity exceeding the best known constructions as well as small families of three-dimensional tensors of rank $n^{1+\Delta}$, for some $\Delta > 0$.

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

Keywords and phrases computational complexity, circuit complexity, lower bounds, conditional lower bounds, monotone circuits, matrix rigidity, tensor rank, arithmetic circuits, fine-grained complexity

Digital Object Identifier 10.4230/LIPIcs.STACS.2026.28

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2025/038 [28]

43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026).
Editors: Meena Mahajan, Florin Manea, Annabelle McIver, and Nguyễn Kim Thắng
Article No. 28; pp. 28:1–28:21

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements We are grateful to the anonymous reviewers for their many helpful comments and additional references, which helped us not only improve the writing and correct errors, but also prove stronger versions of some of our results.

1 Complexity Lower Bounds

Finding the minimum time required to solve a given computational problem is a central question in computational complexity. Answering such a question for a particular problem involves proving a complexity lower bound, that is, showing that no fast algorithm can solve this problem. While the Time Hierarchy Theorem [50, 53] guarantees that there are problems in P that cannot be solved in time $O(n^k)$, for any $k > 1$, we have no superlinear lower bounds for specific problems. For example, for SAT, one of the most important NP-complete problems, we have no algorithms working significantly faster than a brute force approach and at the same time have no methods of excluding a possibility that it can be solved in linear time.

Conditional Lower Bounds

As unconditional complexity lower bounds remain elusive, the classical complexity theory allows one to prove *conditional* lower bounds of the following form: if a problem A cannot be solved in polynomial-time, then B also cannot be solved in polynomial-time. Such results are proved via reductions that are essentially algorithms: one shows how to transform an instance of A into an instance of B. Nowadays, hundreds of such reductions between various NP-hard problems are known. For instance, if SAT cannot be solved in polynomial-time, then the *Hamiltonian Cycle* problem also cannot be solved in polynomial-time.

In a recently emerged area of *fine-grained complexity*, one aims to construct tighter reductions between problems showing that even a tiny improvement of an algorithm for one of them automatically leads to improved algorithms for the other one. For example, as proved by Williams [100], if SAT cannot be solved in time $O(2^{(1-\varepsilon)n})$, for any $\varepsilon > 0$, then the *Orthogonal Vectors* problem cannot be solved in time $O(n^{2-\varepsilon})$, for any $\varepsilon > 0$. Again, many reductions of this form have been developed in recent years. We refer the reader to a recent survey by Vassilevska Williams [97].

Circuit Lower Bounds

One of the reasons why proving complexity lower bounds is challenging is that an algorithm (viewed as a Turing machine or a RAM machine) is a relatively complex object: it has a memory, may contain loops, function calls (that may in turn be recursive). A related computational model of *Boolean circuits* has a much simpler structure (a straight-line program) and at the same time is powerful enough to model algorithms: if a problem can be solved by algorithms in time $T(n)$, then it can also be solved by circuits of size $O(T(n) \log T(n))$ [76]. It turns out that proving circuit lower bounds is also challenging: while it is not difficult to show that almost all Boolean functions can be computed by circuits of exponential size only (this was proved by Shannon [88] back in 1949), for no function from NP, we can currently exclude the possibility that it can be computed by circuits of linear size [66, 34]. Strong lower bounds are only known for restricted models such as monotone circuits, constant-depth circuits, and formulas. Various such unconditional lower bounds can be found in the book by Jukna [61].

An important difference between algorithms and circuits is that algorithms represent a *uniform* model of computation (an algorithm is a program that needs to process instances of all possible lengths), whereas circuits are *nonuniform*: when saying that a problem can be solved by circuits, one usually means that there is an infinite collection of circuits, one circuit for every possible input length, and different circuits in this collection can, in principle, implement different programs. This makes the circuit model strictly more powerful than algorithms: on the one hand, every problem solved by algorithms can be solved by circuits of roughly the same size; on the other hand, it is not difficult to come up with a problem of small circuit size that cannot be solved by algorithms.

Connections Between Lower and Upper Bounds

Intuitively, it seems that proving complexity upper bounds should be easier than proving lower bounds. This intuition is well supported by a much higher number of results on algorithms compared to the number of results on lower bounds. Indeed, to prove an upper bound on the complexity of a problem, one designs an algorithm for the problem and analyzes it. Whereas to prove a complexity lower bound, one needs to reason about a wide range of fast algorithms (or small circuits) and to argue that none of them is able to solve the problem at hand. Perhaps surprisingly, the tasks of proving lower and upper complexity bounds are connected to each other. A classical example is Karp–Lipton theorem [62] stating that if $\mathsf{P} = \mathsf{NP}$, then EXP requires circuits of size $\Omega(2^n/n)$. More recently, Williams [102] established a deep connection between an upper bound for *Circuit SAT* and circuit lower bounds. Extending his results, Jahanjou, Miles and Viola [59] proved that if NSETH is false (meaning that UNSAT can be solved fast with nondeterminism), then E^{NP} requires series-parallel Boolean circuits of size $\omega(n)$. Such results show how to derive nonuniform lower bounds (that is, circuit lower bounds) from uniform *upper* bounds (algorithm upper bounds).

Even though one can simulate an algorithm using circuits with slight overhead, the converse is not true as there are undecidable languages of low circuit complexity. Recently, [12] showed results analogous to those presented herein, particularly on deriving a nonuniform lower bound from a non-randomized uniform lower bound. Specifically, they proved that if MAX-k-SAT cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, for any $\varepsilon > 0$, then, for any $\delta > 0$, there exists an explicit polynomial family that cannot be computed by arithmetic circuits of size $O(n^\delta)$. Also, Williams [103] proved that if the *Orthogonal Vectors* conjecture (OVC) holds, then *Boolean Inner Product* on n-bit vectors cannot be computed by $\mathsf{ETHR} \circ \mathsf{ETHR}$ circuits of size $2^{\varepsilon n}$, for some $\varepsilon > 0$. Combined with the result above (since OVC is weaker than NSETH), it immediately leads to win-win circuit lower bounds: if NSETH fails, we have a lower bound for series-parallel circuits, otherwise we have a strong lower bound for $\mathsf{ETHR} \circ \mathsf{ETHR}$ circuits.

1.1 Our Contribution

In this paper, we derive a number of nonuniform lower bounds from uniform nondeterministic lower bounds. Our lower bounds apply to various objects that are notoriously hard to analyze: Boolean functions of high monotone circuit size, high rigidity matrices, and high rank tensors. For circuits, we get a win-win situation similar to the one by Williams.

Our first result shows how to get $2^{\Omega(n/\log n)}$ monotone circuit lower bounds (improving best known bounds of the form $2^{\Omega(\sqrt{n})}$) under an assumption that SAT requires co-nondeterministic time $O(2^{(1/2+\varepsilon)n})$ if the verifier is given a proof that depends only on the length of the input.

▶ **Theorem 1.** *If, for some $\varepsilon > 0$ and $k \in \mathbb{Z}_{\geq 3}$, k-SAT cannot be solved in input-oblivious co-nondeterministic time $O(2^{(1/2+\varepsilon)n})$, then there exists a monotone Boolean function family in* coNP *of monotone circuit size $2^{\Omega(n/\log n)}$.*

As the previous theorem uses a weaker assumption than NSETH, we get the following corollary.

▶ **Corollary 2.** *If* NSETH *holds, then there exists a monotone Boolean function family in* coNP *with monotone circuit size $2^{\Omega(n/\log n)}$.*

Combining this with circuit lower bounds that follow from the negation of NSETH due to [59, 22] (see Theorem 11), leads to win-win circuit lower bounds.

▶ **Corollary 3.** *At least one of the following two circuit lower bounds holds:*
1. E^{NP} *requires series-parallel circuits of size $\omega(n)$;*
2. *There exists a monotone Boolean function family in* coNP *of monotone circuit size $2^{\Omega(n/\log n)}$.*

Each of these lower bounds is far from what is currently known: the best known circuit lower bound for E^{NP} is $3.1n - o(n)$ [66], whereas the best known monotone circuit lower bound for coNP is $2^{\Omega(\sqrt{n})}$ (see Section 1.2).

Another result in this direction demonstrates how to derive circuit lower bounds from NETH (which asserts that 3-SAT cannot be solved in co-nondeterministic time $2^{o(n)}$). Specifically, we establish lower bounds for the classes Q_t^n.

▶ **Definition 4.** *Let Q_t^n denote the set of all Boolean functions f over n variables such that for any $x^1, \ldots, x^t \in f^{-1}(0)$, there exists an $i \in [n]$ for which $x_i^1 = \ldots = x_i^t = 0$.*

These sets have been studied in secure multiparty computation [54, 35, 55] and from a complexity-theoretic perspective [29, 64]. It turns out that the class Q_t^n coincides precisely with the set of functions f for which there exists a circuit C, composed solely of $\mathsf{THR}_{l+1}^{lt+1}$ gates for arbitrary $l \geq 1$, such that $C \leq f$ [29, 64][1].

Our result indicates that Q_t^n contains functions that are exponentially hard to represent using $\mathsf{THR}_{l+1}^{lt+1}$ gates only while being easy to compute by regular circuits.

▶ **Corollary 5.** *At least one of the following two circuit lower bounds must hold:*
1. E^{NP} *requires circuits of size $\omega(n)$;*
2. *For any $t = \omega(1)$, there exists a function $f \in Q_t^n \cap \mathsf{P/poly}$ such that any circuit $C \leq f$, composed only of $\mathsf{THR}_{l+1}^{lt+1}$ gates for arbitrary $l \geq 1$ over variables, has size $2^{\Omega(n)}$. Moreover, f can be computed in linear time.*

Our second result shows how to construct small families of matrices with rigidity exceeding the best known constructions under an assumption that MAX-3-SAT requires co-nondeterministic time nearly 2^n.

▶ **Theorem 6.** *If, for every $\varepsilon > 0$,* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, then, for all $\delta > 0$, there is a generator $g\colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k}$ computable in time polynomial in k such that, for infinitely many k, there exist a seed s for which $g(s)$ has $k^{\frac{1}{2}-\delta}$-rigidity $k^{2-\delta}$.*

Our third result extends the second result by including high-rank tensors.

[1] By $C \leq f$, we mean that $C(x) \leq f(x)$ for all inputs x.

▶ **Theorem 7.** *If, for any $\varepsilon > 0$,* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, then, for all $\delta > 0$ and some $\Delta > 0$, there are two generators $g_1 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k\times k}$ and $g_2 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k\times k\times k}$ computable in time polynomial in k such that, for infinitely many k, at least one of the following is satisfied:*

- *$g_1(s)$ has $k^{1-\delta}$-rigidity $k^{2-\delta}$, for some s;*
- *$\operatorname{rank}(g_2(s))$ is at least $k^{1+\Delta}$, for some s.*

It is worth noting that [12] showed circuit lower bounds under the same assumption: if MAX-k-SAT cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$ for any $\varepsilon > 0$, then for any $\delta > 0$, there exists an explicit polynomial family that cannot be computed by arithmetic circuits of size $O(n^\delta)$.

The best known lower bound for the size of depth-three circuits computing an explicit Boolean function is $2^{\Omega(\sqrt{n})}$ [51, 75]. Proving a $2^{\omega(\sqrt{n})}$ lower bound for this restricted circuit model remains a challenging open problem and it is known that a lower bound as strong as $2^{\omega(n/\log\log n)}$ would give an $\omega(n)$ lower bound for unrestricted circuits via Valiant's reduction [96]. One way of proving better depth-three circuit lower bounds is via *canonical circuits* introduced by Goldreich and Wigderson [42]. They are closely related to rigid matrices: if T is an $n \times n$ matrix of r-rigidity r^3, then the corresponding bilinear function requires canonical circuits of size $2^{\Omega(r)}$ [42]. Goldreich and Tal [41] showed that a random Toeplitz matrix has r-rigidity $\frac{n^3}{r^2 \log n}$, which implies a $2^{\Omega(n^{3/5})}$ lower bound on canonical depth-three circuits for an explicit function. By substituting $n^{2/3-\delta}$-rigidity for some $\delta > 0$ in Theorem 7, one gets the following result.

▶ **Corollary 8.** *If, for every $\varepsilon > 0$,* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, then, for any $\delta > 0$, one can construct an explicit family of $2^{\log^{O(1)} n}$ functions such that, for infinitely many n, at least one of them is either bilinear and requires canonical circuits of size $2^{\Omega(n^{2/3-\delta})}$ or trilinear and requires arithmetic circuits of size $\Omega(n^{1.25})$.*

This conditionally improves the recent result of Goldreich [40], who presented an $O(1)$-linear function that requires canonical depth-*two* circuits of size $2^{\Omega(n^{1-\varepsilon})}$, for every $\varepsilon > 0$. Moreover, every bilinear function can be computed by canonical circuits of size $2^{O(n^{2/3})}$, so the lower bound is almost optimal and conditionally addresses Open Problem 6.5 from [41].

1.2 Known Explicit Constructions

In this section, we review known constructions of combinatorial objects (functions of high monotone circuit size, matrices of high rigidity, and tensors of high rank).

Monotone Functions

For monotone NP-problems (like *Clique*, *Matching*, *Hamiltonian Cycle*), it is natural to ask what is their monotone circuit size. A celebrated result by Razborov [84] is a lower bound of $n^{\Omega(\log n)}$ on monotone circuit size obtained by the approximation method (which was recently improved to $2^{n^{\Omega(1)}}$ [23]). Subsequently, Andreev [9] proved a $2^{n^{1/8-o(1)}}$ lower bound for another explicit monotone function. Following the work of [7, 10, 60, 49], in 2020, Cavalar, Kumar, and Rossman [24] achieved the best-known lower bound of $2^{n^{1/2-o(1)}}$. Recently, another approach for proving monotone circuit lower bounds was developed using lower bounds from Resolution proofs and lifting theorem [39]. Specifically, if an unsatisfiable formula F is hard to refute in the resolution proof system, then a monotone function associated with F has large monotone circuit complexity. In this manner, following the work of [39, 44, 71], the lower bound of $2^{\Omega(\sqrt{n})}$ was also achieved. Just recently, Blasiok and Meierhöfer [20] established the lower bound $2^{\Omega(\sqrt{n})}$ for the *Clique* problem.

Proving a $2^{\omega(n^{1/2})}$ lower bound remains a challenging open problem (whereas a lower bound $2^{\Omega(n)}$ was recently proved by Pitassi and Robere [77] for monotone *formulas*). Our Corollary 2 establishes a stronger lower bound under an assumption that NSETH holds.

Matrix Rigidity

A matrix M over a field $\mathbb{F}$ has *r-rigidity* s if for any matrices R, S over a field $\mathbb{F}$ such that $M = R + S$ and $\text{rank}(R) \leq r$, S has at least s nonzero entries. That is, one needs to change at least s elements in M to change its rank down to at most r. The concept of rigidity was introduced by Valiant [96] and Grigoriev [45]. It has striking connections to areas such as computational complexity [70, 6, 2, 43], communication complexity [104], data structure lower bounds [32, 80], and error-correcting codes [30].

Valiant [96] proved that if a matrix M has εn-rigidity $n^{1+\delta}$ for some $\varepsilon, \delta > 0$, then the bilinear form of M cannot be computed by arithmetic circuits of size $O(n)$ and depth $O(\log n)$. Following Razborov [86], Wunderlich [104] proved that the existence of strongly-explicit matrices with $2^{(\log\log n)^{\omega(1)}}$-rigidity δn^2, for some $\delta > 0$, implies the existence of a language that does not belong to the communication complexity analog of PH. Although it is known [96] that for any r almost every $n \times n$ matrix has r-rigidity $\Omega(\frac{(n-r)^2}{\log n})$ over algebraically closed fields, obtaining explicit constructions of rigid matrices remains a long-standing open question. Many works have aimed at finding explicit or semi-explicit rigid matrices [37, 79, 89, 6, 31, 2, 33, 99, 15, 13]. Also, a recent line of work establishes a connection between the *Range Avoidance* problem and the construction of matrices with high rigidity [63, 48, 38, 25, 67].

Small explicit[2] families of rigid matrices can be used to prove arithmetic circuit lower bounds [96]. The best known polynomial-time constructible matrices have r-rigidity $\frac{n^2}{r}\log(n/r)$ for any r, which was proved by Shokrollahi, Spielman and Stemann [89]. Goldreich and Tal [41] proved that a random $n \times n$ Toeplitz matrix over $\mathbb{F}_2$ (i.e., a matrix of the form $A_{i,j} = a_{i-j}$ for random bits $a_{-(n-1)}, \ldots, a_{n-1}$) has r-rigidity $\frac{n^3}{r^2 \log n}$ for $r \geq \sqrt{n}$. However, the size of that family is exponential in n. Our Theorem 6 demonstrates that, under the assumption that MAX-3-SAT is hard, for any $\delta > 0$, for infinitely many n one can construct a $2^{\log^{O(1)} n}$-sized family of $n \times n$ matrices with at least one having $n^{1/2-\delta}$-rigidity $n^{2-\delta}$.

This result is still far from the regime where circuit lower bounds can be derived via Valiant's result, but it strictly improves the polynomial-time construction [89] for any $r < \sqrt{n}$ and improves the result of Goldreich and Tal [41] by substantially reducing the family size while maintaining the same rigidity for $r \approx \sqrt{n}$. An open question remains as to whether explicit constructions of rigid matrices exist in the class P^{NP} [81]. The construction provided by Goldreich and Tal [41] lies in E^{NP}. Following the work of Alman and Chen [2], [13] established that there exists a constant $\delta > 0$ such that one can construct $n \times n$ matrices with $2^{\log n/\Omega(\log\log n)}$-rigidity δn^2 in FNP. Subsequently, Chen and Lyu [26] demonstrated a method for constructing highly rigid matrices, proving that there exists a constant $\delta > 0$ such that one can construct $n \times n$ matrices with $2^{\log^{1-\delta} n}$-rigidity $(1/2 - \exp(-\log^{2/3\cdot\delta} n)) \cdot n^2$ in P^{NP}. More recently, Alman and Liang [4] showed that the Walsh-Hadamard matrix H_n has $c_1 \log n$-rigidity $n^2\left(\frac{1}{2} - n^{-c_2}\right)$ for some constants $c_1, c_2 > 0$. Our construction, in the class $\mathsf{DTIME}[2^{\log^{O(1)} n}]^{\mathsf{NP}}$, produces matrices with $n^{\frac{1}{2}-\delta}$-rigidity $n^{2-\delta}$ for any $\delta > 0$, under the condition that MAX-3-SAT is hard.

[2] A matrix or family of matrices is called explicit if it is polynomial-time constructible.

Tensor Rank and Arithmetic Circuits

Proving arithmetic circuit lower bounds is another important challenge in complexity theory. An arithmetic circuit over a field $\mathbb{F}$ uses as inputs formal variables and field elements and computes in every gate either a sum or a product. As proved by Strassen [93, 94] and Baur and Strassen [11], computing $\sum_{i=1}^{n} x_i^n$ requires arithmetic circuits of size $\Omega(n \log n)$, provided n does not divide the characteristic of $\mathbb{F}$. Raz [82] further established that arithmetic circuits with bounded coefficients require $\Omega(n^2 \log n)$ gates to perform matrix multiplication over $\mathbb{R}$ or $\mathbb{C}$, following the work in [83]. However, no superlinear lower bounds are known for polynomials of constant degree. For constant-depth arithmetic circuits over fields of characteristic 2, exponential lower bounds are known [85, 91]. For other finite characteristics, exponential lower bounds are known only for depth 3 [46, 47]. For characteristic 0, Limaye, Srinivasan, and Tavenas [68] proved the first superpolynomial lower bounds for constant-depth circuits. We refer to [68, 8, 14, 36] and the references therein for recent advances in lower bounds for small-depth algebraic circuits.

Matrix multiplication is one of the fundamental problems whose arithmetic circuit size is of great interest. While many highly nontrivial algorithms for it are known (starting from Strassen [92]), we still do not have superlinear lower bounds on its arithmetic circuit complexity. Proving such lower bounds is closely related to the problem of determining the rank of tensors. A d-dimensional tensor is said to have rank q if it can be expressed as a sum of q rank-one tensors. Here, a rank-one d-dimensional tensor is a tensor of the form $u_1 \otimes \cdots \otimes u_d$, where $\otimes$ stands for a tensor product. By a multiplication tensor, we mean a tensor of size $n^2 \times n^2 \times n^2$ (formally defined in Section 3.2). Establishing an upper bound for the rank of the multiplication tensor provides a means of proving upper bounds for matrix multiplication via the laser method [95]. Moreover, proving a lower bound for the tensor rank would yield superlinear lower bounds for arithmetic circuits computing the polynomial defined by that tensor.

Therefore, proving lower bounds on the tensor rank provides a path to proving lower bounds for arithmetic circuits. For the rank of the matrix multiplication tensor, Bshouty [21] and Bläser [18] proved a lower bound $2.5n^2 - \Theta(n)$. Subsequently, Shpilka [90] improved the bound to $3n^2 - o(n^2)$ over $\mathbb{F}_2$. The bound $3n^2 - o(n^2)$ was later achieved by Landsberg [65] over arbitrary fields and further slightly improved by Massarenti and Raviolo [72, 73]. Alexeev, Forbes and Tsimerman [1] constructed explicit d-dimensional tensors with rank $2n^{\lfloor \frac{d}{2} \rfloor} + n - \Theta(d \log n)$, thus improving the lower bounds on high-dimensional tensors. Nevertheless, superlinear size lower bounds for constant-degree polynomials remain unknown. Additionally, Håstad [52] established that determining the rank of a d-dimensional tensor is NP-hard for any $d \geq 3$. Consequently, a major open problem is to construct an explicit family of d-dimensional tensors with rank at least $n^{\lfloor \frac{d}{2} \rfloor + \varepsilon}$ for some $\varepsilon > 0$ and $d \geq 3$.

Our Theorem 7 shows that, under an assumption that MAX-3-SAT cannot be solved fast co-nondeterministically, one gets an explicit $2^{\log^{O(1)} n}$-size family of $n \times n$-matrices and $n \times n \times n$-tensors, such that, for any $\delta > 0$ and some $\Delta > 0$, at least one of the matrices has $n^{1-\delta}$-rigidity $n^{2-\delta}$ or one of the tensors has rank $n^{1+\Delta}$. Furthermore, we establish a trade-off between matrix rigidity and tensor rank, see Theorem 21. Other results for proving lower bounds on tensor rank under certain assumptions are known. Nederlof [74] proved that, if for any $\varepsilon > 0$, the bipartite *Traveling Salesman* problem cannot be solved in time $2^{(1-\varepsilon)n}$, then the matrix multiplication tensor has superlinear rank. Additionally, Björklund and Kaski [17] recently proved that if, for any $\varepsilon > 0$, there exists a k such that the *k-Set Cover* problem cannot be solved in time $O(2^{(1-\varepsilon)n}|\mathcal{F}|)$, then there is an explicit tensor with superlinear rank, where $\mathcal{F}$ is a family of subsets of $[n]$, each of size at most k. Pratt [78] improved this result,

showing that under the same conjecture there exists an explicit tensor of shape $n \times n \times n$ and rank at least $n^{1.08}$. [16] showed that if for every $\varepsilon > 0$ *Chromatic Number* problem cannot be solved in time $2^{(1-\varepsilon)n}$, then there exists an explicit tensor of superlinear rank.

1.3 Discussion and Open Problems

Many of the lower bounds mentioned above are proved under various strong assumptions (on the complexity of SAT, MAX-SAT, *Set Cover*, *Chromatic Number*, *Traveling Salesman*). They seem much stronger than merely $\mathsf{P} \neq \mathsf{NP}$ and might eventually be refuted. Still, the revealed reductions between problems are of great interest and may still yield further insights: one may be able to weaken the used assumptions or to construct generators from other fine-grained reductions. Moreover, as with the case of NSETH assumption, one may be able to derive interesting consequences from both an assumption and its negation leading to a win-win situation. Below, we state a few open problems in this direction.

If one were to partition 3-SAT in Theorem 12 into t parts, where $t = \omega(1)$, then creating an explicit function equivalent to t-OV would imply lower bounds for that function under NETH. This approach would yield a more advantageous win-win situation, as the assumption that NETH is false gives stronger lower bounds [27].

▶ **Open Problem 1.** *Prove that if* NETH *is true, then there exists an explicit function of monotone complexity* $2^{\omega(\sqrt{n})}$.

Moreover, the existence of small monotone circuits not only refutes NSETH but also establishes that UNSAT has input-oblivious proof size $2^{o(n)}$ that can be verified in time $2^{\frac{n}{2}+o(n)}$. Therefore, we believe that it may be possible to derive even stronger implications beyond merely refuting NSETH.

Another question arises regarding tensor rank. Is it possible to construct a small family of tensors with superlinear rank, assuming that MAX-3-SAT cannot be solved efficiently in co-nondeterministic time (this way, eliminating the dependence on rigidity from our results)?

▶ **Open Problem 2.** *Prove that, if, for any* $\varepsilon > 0$, MAX-*3*-SAT *cannot be solved in co-nondeterministic time* $O(2^{(1-\varepsilon)n})$*, then, for all* $\delta > 0$ *there exists a small family of tensors of size* $k \times k \times k$ *such that, for infinitely many* k*, at least one of them has rank at least* $k^{1+\delta}$.

However, we do not know of any consequences of solving MAX-3-SAT faster in the co-nondeterministic setting. This makes our assumption weak and raises the question of whether it can be refuted.

▶ **Open Problem 3.** *Prove that, for some* $\varepsilon > 0$, MAX-*3*-SAT *can be solved in co-nondeterministic time* $O(2^{(1-\varepsilon)n})$.

On the other hand, to get a win-win situation, it would be interesting to find nontrivial consequences of the existence of such an algorithm.

▶ **Open Problem 4.** *Derive new circuit lower bounds from the existence of an algorithm solving* MAX-*3*-SAT *in co-nondeterministic time* $O(2^{(1-\varepsilon)n})$*, for some* $\varepsilon > 0$.

Structure of the Paper

The paper is organized as follows. In Section 2, we give an overview of the main proof ideas. In Section 3, we introduce the notation used throughout the paper and provide the necessary background. In Section 4, we establish the win-win circuit lower bound. In Section 5,

we construct rigid matrices under an assumption that MAX-3-SAT cannot be solved fast co-nondeterministically. In Section 6, we construct either three-dimensional tensors with high rank or matrices with high rigidity under the same assumption. The proofs of certain statements are omitted due to the page limit. They can be found in the full version [28].

2 Proof Ideas

In this section, we give high-level ideas of the main results.

2.1 Monotone Circuit Lower Bound

To prove a lower bound $2^{\Omega(n/\log n)}$ for monotone circuit size of coNP under NSETH, we assume that all monotone functions from coNP have monotone circuit size $2^{o(n/\log n)}$ and show how this can be used to solve UNSAT nondeterministically in less than 2^n steps.

Given a k-CNF F, we construct an instance (A_F, B_F) of OV of size $2^{n/2}$ using Theorem 12. We show (see Lemma 15) that (A_F, B_F) is a yes-instance if and only if there exists a monotone Boolean function separating $(A_F, \overline{B_F})$. Hence, one can guess a small monotone circuit separating $(A_F, \overline{B_F})$ and verify that it is correct. Overall, the resulting nondeterministic algorithm proceeds as follows.

1. In time $O(n^2 2^{n/2})$, generate the sets A_F and B_F.
2. Guess a monotone circuit C_F of size $O(2^{(1-\varepsilon)n/2})$.
3. Verify that C_F separates $(A_F, \overline{B_F})$. To do this, check that $C_F(a) = 1$, for every $a \in A_F$. Then, check that $C_F(b) = 0$, for every $b \in \overline{B_F}$. The running time of this step is
$$O(2^{(1-\varepsilon)n/2} \cdot |A_F| + 2^{(1-\varepsilon)n/2} \cdot |\overline{B_F}|) = O(2^{(1-\varepsilon)n/2} \cdot 2^{n/2}) = O(2^{(1-\varepsilon/2)n}).$$
4. If C_F separates $(A_F, \overline{B_F})$, then $F \in$ UNSAT.

This already shows how small monotone circuits could help to break NSETH, though it does not provide a single explicit function with this property. In the full proof in Section 4, we introduce such a function. We also use a weaker assumption (than NSETH).

2.2 Rigid Matrices and High Rank Tensors

To construct matrices of high rigidity under the assumption that MAX-3-SAT cannot be solved fast co-nondeterministically, we proceed as follows. Take a 3-CNF formula over n variables and an integer t and transform it, using Theorem 14, into a 4-partite 3-uniform hypergraph G with each part of size $k = n^{O(1)} 2^{\frac{n}{4}}$. The graph G contains a 4-clique if and only if one can satisfy t clauses of F. We show that checking whether G has a 4-clique is equivalent to evaluating a certain expression over three-dimensional tensors. We then show that if all slices of these tensors have low rigidity, then one can solve 4-*Clique* on G in co-nondeterministic time $O(k^{4-\varepsilon})$: to achieve this, one guesses a decomposition of a matrix into a sum of a low rank matrix and a matrix with few nonzero entries. The idea is that a low rank matrix can be guessed quickly as a decomposition into rank-one matrices (which are just products of two vectors), whereas the second matrix can be guessed quickly as one needs to guess the nonzero entries only. In turn, this allows one to solve MAX-3-SAT faster than 2^n co-nondeterministically. Thus, this reduction is a generator of rigid matrices: it takes a 3-CNF formula and outputs a matrix. The same idea can be used to generate either tensors with high rank or matrices with high rigidity.

3 Preliminaries

For a positive integer k, $[k] = \{1, 2, \ldots, k\}$, whereas for a predicate P, $[P] = 1$ if P is true and $[P] = 0$ otherwise (the Iverson bracket). For a set S and an integer k, by $\binom{S}{k}$ we denote the set of all subsets of S of size k.

The definitions of Boolean circuits and arithmetic circuits are omitted due to space restrictions. They can be found in the full version [28].

3.1 SAT, MAX-SAT, OV, and Clique

For a CNF formula F, by $n(F)$ and $m(F)$ we denote the number of variables and clauses of F, respectively. We write just n and m, if the corresponding CNF formula is clear from the context. In SAT (UNSAT), one is given a CNF formula and the goal is to check whether it is satisfiable (unsatisfiable, respectively). In k-SAT, the given formula is in k-CNF (that is, all clauses have at most k literals). In MAX-k-SAT, one is given a k-CNF and an integer t and is asked to check whether it is possible to satisfy exactly t clauses.

When designing an algorithm for SAT, one can assume that the input formula has a linear (in the number of variables) number of clauses. This is ensured by the following *Sparsification Lemma*. By (β, k)-SAT, we denote a special case of k-SAT where the input k-CNF formula has at most βn clauses.

▶ **Theorem 9** (Sparsification Lemma, [58]). *For any $k \in \mathbb{Z}_{\geq 3}$ and $\varepsilon > 0$, there exists $\alpha = \alpha(k, \varepsilon)$ and an algorithm that, given a k-CNF formula F over n variables, outputs $t \leq 2^{\varepsilon n}$ formulas $F_1, \ldots, F_t$ in k-CNF such that $n(F_i) \leq n$ and $m(F_i) \leq \alpha n$, for all $i \in [t]$, and $F \in$ SAT if and only if $\vee_{i \in [t]} F_i \in$ SAT. The running time of the algorithm is $O(n^{O(1)} 2^{\varepsilon n})$.*

▶ **Corollary 10.** *There exists a function $\beta \colon \mathbb{Z}_{\geq 3} \times \mathbb{R}_{>0} \to \mathbb{Z}_{>0}$ such that, if there exists $\varepsilon > 0$ for which $(\beta(k, \varepsilon), k)$-SAT can be solved in time $O(2^{n/2+\varepsilon n})$ for any $k \in \mathbb{Z}_{\geq 3}$, then k-SAT can be solved in time $O(2^{n/2+2\varepsilon n})$.*

The proof of Corollary 10 can be found in the full version [28].

The *Strong Exponential Time Hypothesis* (SETH), introduced in [58, 57], asserts that, for any $\varepsilon > 0$, there is k such that k-SAT cannot be solved in time $O(2^{(1-\varepsilon)n})$. The *Nondeterministic* SETH (NSETH), introduced in [22], extends SETH by asserting that SAT is difficult even for co-nondeterministic algorithms: for any $\varepsilon > 0$, there is k such that k-SAT cannot be solved in *co-nondeterministic* time $O(2^{(1-\varepsilon)n})$. Though both these statements are stronger than $\mathsf{P} \neq \mathsf{NP}$, they are known to be hard to refute: as proved by Jahanjou, Miles and Viola [59], if SETH is false, then there exists a Boolean function family in E^{NP} of series-parallel circuit size $\omega(n)$. [22] noted that it suffices to refute NSETH to get the same circuit lower bound.

▶ **Theorem 11** ([59, 22]). *If NSETH is false, then there exists a Boolean function family in E^{NP} of series-parallel circuit size $\omega(n)$.*

SETH-based conditional lower bounds are known for a wide range of problems and input parameters. One of such problems is *Orthogonal Vectors* (OV): given two sets $A, B \subseteq \{0,1\}^d$ of size n, check whether there exists $a \in A$ and $b \in B$ such that $a \cdot b = \sum_{i \in [d]} a_i b_i = 0$. It is straightforward to see that OV can be solved in time $O(n^2 d)$. Williams [100] proved that under SETH, there is no algorithm solving OV in time $O(n^{2-\varepsilon} d^{O(1)})$, for any $\varepsilon > 0$. This follows from the following reduction.

▶ **Theorem 12** ([100]). *There exists an algorithm that, given a CNF formula F with n variables and m clauses, outputs two sets $A_F, B_F \subseteq \{0,1\}^m$ such that $|A_F| = |B_F| = 2^{n/2}$ and $F \in \mathsf{SAT}$ if and only if $(A_F, B_F) \in \mathsf{OV}$. The running time of the algorithm is $O(mn \cdot 2^{n/2})$.*

In a similar manner, one can reduce a CNF formula to the t-OV problem involving a greater number of sets. Consider the t-OV problem, where one is given t sets $A^1, \ldots, A^t \subseteq \{0,1\}^d$, each of size n, and the objective is to determine whether there exist elements $a^1 \in A^1, \ldots, a^t \in A^t$ such that $\sum_{i \in [d]} a^1_i \cdot \ldots \cdot a^t_i = 0$.

▶ **Lemma 13.** *There exists an algorithm which, given a CNF formula F with n variables and m clauses, along with an integer t, constructs t sets $A^1_F, \ldots, A^t_F \subseteq \{0,1\}^m$ such that $|A^1_F| = \ldots = |A^t_F| = 2^{n/t}$ and $F \in \mathsf{SAT}$ if and only if $(A^1_F, \ldots, A^t_F) \in t$-OV. The running time of the algorithm is $O(mn \cdot 2^{n/t})$.*

The proof follows the approach presented in [100]. An instance $(A^1, \ldots, A^t) \notin t$-OV if and only if for all vectors $a^1 \in A^1, \ldots, a^t \in A^t$, the vectors share a common coordinate with value one. This can equivalently be formulated by stating that the union $Q = A^1 \cup \ldots \cup A^t$ possesses the property that for every $a^1, \ldots, a^t \in Q$, the vectors share a common one. Consequently, the instance $(A^1, \ldots, A^t) \notin t$-OV if and only if there exists a circuit C, composed of $\mathsf{THR}^{lt+1}_{l+1}$ gates and variables, such that $C(\overline{x}) = 0$ for every $x \in A^1 \cup \ldots \cup A^t$.

A similar reduction allows to solve MAX-3-SAT by finding a 4-clique in a 3-uniform hypergraph. A subset of l nodes in a k-hypergraph is called an *l-clique*, if any k of them form an edge in the graph.

▶ **Theorem 14** ([101, 69]). *There exists an algorithm that, given a 3-CNF formula F with n variables and an integer t, outputs a 4-partite 3-uniform hypergraph G with parts of size $k = n^{O(1)}2^{n/4}$ such that G has a 4-clique if and only if it is possible to satisfy exactly t clauses of F.*

The proof is given in the full version [28].

3.2 Rigidity and Tensor Rank

For a field $\mathbb{F}$, by $\mathbb{F}^{a \times b}$ we denote the set of all matrices of size $a \times b$ over $\mathbb{F}$. Similarly, by $\mathbb{F}^{a \times b \times c}$ we denote the set of all three-dimensional tensors of shape $a \times b \times c$ over $\mathbb{F}$. For two tensors A and B (of arbitrary shape), by $A \otimes B$ we denote a tensor product of A and B. For a matrix $M \in \mathbb{F}^{a \times b}$, by $|M|$ we denote the number of nonzero entries of M.

For a matrix $M \in \mathbb{F}^{a \times b}$, we say that it has r-rigidity s if it is necessary to change at least s entries of M to reduce its rank to r. That is, for each decomposition $M = R + S$ such that $\mathrm{rank}(R) \leq r$, it holds that $|S| \geq s$.

The rank of a three-dimensional tensor is a natural extension of the matrix rank. For a tensor $\mathcal{A} \in \mathbb{F}^{n \times n \times n}$, we define its rank, $\mathrm{rank}(\mathcal{A})$, as the smallest integer r such that there exist r tuples of vectors $a_l, b_l, c_l \in \mathbb{F}^n$ for which

$$\mathcal{A} = \sum_{l \in [r]} a_l \otimes b_l \otimes c_l,$$

or equivalently,

$$\mathcal{A}[i,j,k] = \sum_{l \in [r]} a_l[i] b_l[j] c_l[k],$$

for all $i, j, k \in [n]$.

We denote by ω the smallest real number such that any two $n \times n$ matrices can be multiplied in time $O(n^{\omega+\varepsilon})$ for any $\varepsilon > 0$ using only field operations[3].

Consider the three-dimensional tensor $\mathcal{A}_n \in \mathbb{F}^{n^2 \times n^2 \times n^2}$:

$$\mathcal{A}_n[(i,j),(i,k),(k,j)] = 1,$$

for all $i, j, k \in [n]$, with all other entries being zero. Using an approach based on the work of Strassen [92], for any positive integer k, if $\operatorname{rank}(\mathcal{A}_k) = q$, then one can construct an arithmetic circuit of size $O(n^{\log_k(q)})$ to perform multiplication of two $n \times n$ matrices. Thus, ω satisfies the following equation:

$$\omega = \inf_{k \in \mathbb{Z}_{>0}} \log_k \operatorname{rank}(\mathcal{A}_k).$$

In other words, for sufficiently large k, we have that $\operatorname{rank}(\mathcal{A}_k) \geq k^\omega$. Specifically, if $n = k^2$, then $\mathcal{A}_k \in \mathbb{F}^{n \times n \times n}$, and $\operatorname{rank}(\mathcal{A}_k) \geq n^{\omega/2}$. Therefore, if $\omega > 2$, this yields superlinear lower bounds on arithmetic circuits and on the rank of the multiplication tensor.

The best known upper bound on ω is 2.371339 [98, 3]. Further details on the matrix multiplication tensor can be found in [5, 19].

4 Circuit Lower Bounds

4.1 Boolean Functions of High Monotone Circuit Size

In this section, we prove Theorem 1.

▶ **Theorem 1.** *If, for some $\varepsilon > 0$ and $k \in \mathbb{Z}_{\geq 3}$, k-SAT cannot be solved in input-oblivious co-nondeterministic time $O(2^{(1/2+\varepsilon)n})$, then there exists a monotone Boolean function family in* coNP *of monotone circuit size $2^{\Omega(n/\log n)}$.*

Combining this with Corollary 2 and Theorem 11, we get win-win circuit lower bounds. Proving any of these two circuit lower bounds is a challenging open problem.

▶ **Corollary 3.** *At least one of the following two circuit lower bounds holds:*

1. E^{NP} *requires series-parallel circuits of size $\omega(n)$;*
2. *There exists a monotone Boolean function family in* coNP *of monotone circuit size $2^{\Omega(n/\log n)}$.*

For the proof of Theorem 1, we need two technical lemmas. Recall that the reduction from SAT to OV (see Theorem 12), given a formula F, produces two sets $A_F, B_F \subseteq \{0,1\}^{m(F)}$ such that $|A_F| = |B_F| = 2^{n(F)/2}$ and $F \in \mathsf{SAT}$ if and only if $(A_F, B_F) \in \mathsf{OV}$.

▶ **Lemma 15.** *Let F be a CNF formula. Then, $F \notin \mathsf{SAT}$ if and only if $(A_F, \overline{B_F})$ can be separated by a monotone function.*

For a CNF formula F, let $f_F \colon \{0,1\}^{m(F)} \to \{0,1\}$ be defined as follows:

$$f_F(x) = [\forall b \in \overline{B_F} \colon b \not\geq x]. \tag{1}$$

It is immediate that f_F is monotone and that $\overline{B_F} \subseteq f_F^{-1}(0)$.

[3] The value of ω may depend on the field over which the calculations are performed [87].

Proof of Lemma 15. Assume that $F \notin \mathsf{SAT}$. We show that the function f_F separates $(A_F, \overline{B_F})$. To do this, it suffices to show that $A_F \subseteq f_F^{-1}(1)$. If this is not the case, then there is $a \in A_F$ such that $f_F(a) = 0$, that is, there exists $b \in \overline{B_F}$ such that $b \geq a$. Hence, there is no $i \in [m]$ such that $b_i = 0$ and $a_i = 1$. In turn, this means that, for $\overline{b} \in B_F$, there is no $i \in [m]$ such that $\overline{b}_i = 1$ and $a_i = 1$, meaning that $(A_F, B_F) \in \mathsf{OV}$, contradicting $F \notin \mathsf{SAT}$.

For the reverse direction, assume that for some monotone function $h\colon \{0,1\}^m \to \{0,1\}$, it holds that $A_F \subseteq h^{-1}(1)$ and $\overline{B_F} \subseteq h^{-1}(0)$. By the monotonicity of h, for every $a \in A_F$ and every $b \in \overline{B_F}$, $b \not\geq a$. Hence, for every $a \in A_F$ and every $b \in \overline{B_F}$, there exists $i \in [m]$ such that $b_i = 0$ and $a_i = 1$. Switching from $\overline{B_F}$ to B_F, we get that, for every $a \in A_F$ and every $b \in B_F$, there exists $i \in [m]$ such that $b_i = 1$ and $a_i = 1$, meaning that $(A_F, B_F) \notin \mathsf{OV}$ and $F \notin \mathsf{SAT}$. ◀

Note that the function f_F can be viewed as a monotone unsatisfiability certificate for F. Such functions have been studied in proof complexity; see Hrubeš–Pudlák [56].

By $\mathcal{F}_{n,k,\beta}$ denote the set of k-CNF formulas with n variables and at most βn clauses. Any formula $F \in \mathcal{F}_{n,k,\beta}$ can be encoded in binary using at most $\gamma n \log n$ bits (for some $\gamma = \gamma(k,\beta)$): each variable is encoded using $\log n$ bits, all other symbols (parentheses as well as negations, disjunctions, and conjunctions require a constant number of bits). Hence,

$$|\mathcal{F}_{n,k,\beta}| \leq 2^{\gamma n \log n}.$$

Let us call $\mathcal{W}_{n,k}$ the set of all binary strings of length n and weight k:

$$\mathcal{W}_{n,k} = \{x \in \{0,1\}^n \colon w(x) = k\}.$$

The following lemma can be established using standard techniques. The proof is provided in the full version [28].

▶ **Lemma 16.** *There exists an injective encoding $e\colon \{0,1\}^{\leq n} \to \mathcal{W}_{4n,2n}$ such that computing and inverting e takes time linear in n.*

As a simple corollary, applying Lemma 16 to Boolean formulas, one can obtain the following result.

▶ **Lemma 17.** *There exists a parameter $l = O(n \log n)$ and an injective encoding*

$$e\colon \mathcal{F}_{n,k,\beta} \to \mathcal{W}_{l,l/2}$$

such that computing and inverting e takes time polynomial in n.

Proof. Given a formula $F \in \mathcal{F}_{n,k,\beta}$, we can encode it in binary using at most $\gamma n \log n$ bits (as stated above). Then, applying Lemma 16, we obtain the desired result. ◀

Proof of Theorem 1. Assuming that coNP has small monotone circuits, we design a fast co-nondeterministic algorithm for SAT. Assume that all monotone functions over N variables in coNP can be computed by monotone circuits of size $2^{o(N/\log N)}$.

Let F be a k-CNF over n variables. Thanks to Corollary 10, we may assume that $m(F) \leq \beta n$, where $\beta = \beta(k, \frac{\varepsilon}{2})$. Then, solving $\mathcal{F}_{n,k,\beta}$ in co-nondeterministic time $O(2^{n/2+\varepsilon' n})$ for $\varepsilon' = \frac{\varepsilon}{2}$ will be sufficient for a contradiction.

Below, we define a universal function f containing f_F, for all $F \in \mathcal{F}_{n,k,\beta}$, as subfunctions (recall the definition (1) of f_F). Let $N = l + m$, where $l = O(n \log n)$ and e is the function from Lemma 17 and $m = \beta n$ (hence, $N = O(n \log n)$). We define $f\colon \{0,1\}^N \to \{0,1\}$ as follows. Let (c, x), where $c \in \{0,1\}^l$ and $x \in \{0,1\}^m$, be an input of f. Then,

$$f(c,x) = \begin{cases} 1, & \text{if } w(c) > l/2, \\ 0, & \text{if } w(c) < l/2, \\ 1, & \text{if } w(c) = l/2 \text{ and } e^{-1}(c) = \varnothing, \\ 0, & \text{if } w(c) = l/2 \text{ and } e^{-1}(c) \in \mathsf{SAT}, \\ f_F(x), & \text{if } w(c) = l/2 \text{ and } F = e^{-1}(c) \notin \mathsf{SAT}. \end{cases}$$

We now ensure three important properties of f.

1. *The function f is monotone.* For the sake of contradiction, assume that $(c, x) \geq (c', x')$, but $0 = f(c, x) < f(c', x') = 1$. Since $f(c', x') = 1$, $w(c') \geq l/2$. If $w(c') > l/2$, then $f(c, x) = 1$, since $w(c) \geq w(c') > l/2$. Hence, assume that $w(c') = w(c) = l/2$. Since $c \geq c'$, we conclude that $c = c'$ (implying that $e^{-1}(c) \neq \varnothing$). If $e^{-1}(c) \in \mathsf{SAT}$, then $f(c', x') = 0$. Hence $e^{-1}(c) \notin \mathsf{SAT}$, thus $f(c, x) = f_F(x)$ and $f(c', x') = f_F(x')$, where $F = e^{-1}(c) = e^{-1}(c')$, and f_F is clearly monotone.
2. *The function f is explicit: $f \in \mathsf{coNP}$.* Assume that $f(c, x) = 0$. This can happen for one of the three following reasons.
   - $w(c) < l/2$. This is easily verifiable.
   - $w(c) = l/2$ and $e^{-1}(c) \in \mathsf{SAT}$. To verify this, one computes $F = e^{-1}(c)$ (in time $O(\mathsf{poly}(n))$, due to Lemma 17), guesses its satisfying assignment and verifies it.
   - $w(c) = l/2$, $F = e^{-1}(c) \notin \mathsf{SAT}$, and $f_F(x) = 0$. Recall that if $F \in \mathsf{SAT}$, then we can guess and verify its satisfying assignment, so in this case there is no need to verify the unsatisfiability of F. Since $f_F(x) = 0$, there exists $b \in \overline{B_F}$ such that $b \geq x$. To verify this, one computes $F = e^{-1}(c)$, guesses the corresponding assignment to the second half of the variables of F, ensures that it produces the vector $\bar{b}$, and verifies that $b \geq x$.
3. Assume that f can be computed by a monotone circuit C of size $2^{o(N/\log N)}$. We show that, then, for any $k \in \mathbb{Z}_{\geq 3}$, k-SAT can be solved in input-oblivious co-nondeterministic time $O(2^{n/2+\varepsilon' n})$. Since $N = O(n \log n)$, then $\mathrm{size}(C) = O(2^{\varepsilon' n})$.
   Let $F \in \mathcal{F}_{n,k,\beta}$ be an unsatisfiable formula with n variables and no more than βn clauses. The following algorithm verifies its unsatisfiability in input-oblivious nondeterministic time $O(2^{n/2+\varepsilon' n})$.
   a. In time $O(n^2 2^{n/2})$, generate the sets A_F and B_F.
   b. Guess a monotone circuit C. This can be done in time $O(\mathrm{size}(C)) = O(2^{\varepsilon' n})$.
   c. Compute $c = e(F)$ and substitute the first l variables of C by c. Call the resulting circuit C_F.
   d. Verify that C_F separates $(A_F, \overline{B_F})$. To do this, check that $C_F(a) = 1$, for every $a \in A_F$. Then, check that $C_F(b) = 0$, for every $b \in \overline{B_F}$. The running time of this step is
   $$O((|A_F| + |\overline{B_F}|) \cdot \mathrm{size}(C_F)) = O(2^{n/2} \cdot 2^{\varepsilon' n}) = O(2^{n/2+\varepsilon' n}).$$
   e. If C_F is monotone and separates $(A_F, \overline{B_F})$, then we are certain that $F \notin \mathsf{SAT}$, thanks to Lemma 15. ◀

5 Matrices of High Rigidity

In this section, we show that low rigidity matrices can be utilized to solve MAX-3-SAT more efficiently. Thus, assuming that MAX-3-SAT cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, for any $\varepsilon > 0$, we obtain a generator of high rigidity matrices. Throughout this section, rigidity decompositions are considered over a finite field $\mathbb{F}$.

▶ **Theorem 6.** *If, for every $\varepsilon > 0$,* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, then, for all $\delta > 0$, there is a generator $g\colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k\times k}$ computable in time polynomial in k such that, for infinitely many k, there exist a seed s for which $g(s)$ has $k^{\frac{1}{2}-\delta}$-rigidity $k^{2-\delta}$.*

The proof is given in the full version [28].

As a simple corollary, one might weaken the assumption and obtain weaker matrix rigidity, while still improving the known explicit construction by [89]. Recall that [89] constructed matrices with r-rigidity $\frac{n^2}{r}\log(n/r)$.

▶ **Corollary 18.** *If* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{0.92n})$, then there exists a generator $g\colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k\times k}$ computable in time polynomial in k such that, for infinitely many k, there exists a seed s for which $g(s)$ has $k^{0.34}$-rigidity $k^{1.68}$.*

Proof. The generator is the one constructed in the proof of Theorem 6. Substituting $r = k^{0.34}$ and $s = k^{1.68}$ into the running time $O(k^2 s + r^2 k^3)$ yields the running time $O(k^{3.68}) = O(2^{0.92n})$, thereby completing the proof. Note that $r \cdot s = k^{2.02}$, so further weakening the assumption will result in worse rigidity than that achieved in [89]. ◀

6 Tensors of High Rank

In this section, we show how to generate high rank tensors under an assumption that MAX-3-SAT is hard. Throughout this section, we assume that $\mathbb{F}$ is a finite field over which rigidity and tensor decompositions are considered.

We start with the lemma that shows how one can compute the value of a specific function for all inputs using fast matrix multiplication algorithms.

▶ **Lemma 19.** *Let $q \geq k$ and $A, B, C \in \mathbb{F}^{q\times k}$. Let also $f\colon [k]^3 \to \mathbb{F}$ be defined as follows:*

$$f(i,j,m) = \sum_{l\in[q]} A[l,i]B[l,j]C[l,m].$$

One can compute $f(i,j,m)$, for all $(i,j,m) \in [k]^3$ simultaneously, in time $O(qk^\omega)$.

The proof of Lemma 19 is given in the full version [28].

▶ **Theorem 20.** *Let $k = n^{O(1)}2^{n/4}$ and $r \geq \sqrt{k}$, $q \geq k$. There exist functions $g_1\colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k\times k}$ and $g_2\colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k\times k\times k}$ computable in time $k^{O(1)}$ such that, if, for any e, $g_1(e)$ has r-rigidity s and $g_2(e)$ has rank at most q, then,* MAX-*3*-SAT *for formulas with n variables can be solved in co-nondeterministic time*

$$O(k^2 s + qr^2 k^{\omega-1}).$$

The proof is given in the full version [28].

The following theorem establishes a trade-off between matrix rigidity and tensor rank by applying Theorem 20.

▶ **Theorem 21.** *Let $\alpha \in [0.5, 1]$ and $\beta \in [1, 2]$ be constants satisfying $\alpha \leq \frac{5-\beta-\omega}{2}$. If* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, for any $\varepsilon > 0$, then, for all $\delta > 0$, there exist functions $g_1 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k}$ and $g_2 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k \times k}$ computable in time $k^{O(1)}$, such that, for infinitely many k, at least one of the following is satisfied, for at least one e:*

- *$g_1(e)$ has $k^{\alpha-\delta}$-rigidity $k^{2-\delta}$;*
- *$\operatorname{rank}(g_2(e))$ is at least $k^{\beta-\delta}$.*

The proof is provided in the full version [28].

One way to balance matrix rigidity and tensor rank is to ensure that a tensor has superlinear rank while maximizing matrix rigidity, as demonstrated in the next corollary.

▶ **Corollary 22.** *If* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$ for any $\varepsilon > 0$, then for all $\delta > 0$ there are two polynomial-time generators $g_1 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k}$ and $g_2 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k \times k}$ such that for infinitely many k at least one of the following is satisfied:*

- *$g_1(s)$ has $k^{2-\frac{\omega}{2}-\delta}$-rigidity $k^{2-\delta}$, for some s.*
- *$\operatorname{rank}(g_2(s))$ is at least $k^{1+\delta}$, for some s.*

One can also improve matrix rigidity in our trade-off by conditioning on the value of ω.

▶ **Theorem 7.** *If, for any $\varepsilon > 0$,* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, then, for all $\delta > 0$ and some $\Delta > 0$, there are two generators $g_1 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k}$ and $g_2 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k \times k}$ computable in time polynomial in k such that, for infinitely many k, at least one of the following is satisfied:*

- *$g_1(s)$ has $k^{1-\delta}$-rigidity $k^{2-\delta}$, for some s;*
- *$\operatorname{rank}(g_2(s))$ is at least $k^{1+\Delta}$, for some s.*

Proof. We modify the second generator in Corollary 22 by adding a new input on which the generator will output a tensor of matrix multiplication $\mathcal{A}_{\sqrt{k}}$ of size $k \times k \times k$. If $\omega = 2$, then the statement follows from Corollary 22. If $\omega \geq 2 + 2\Delta$ for some $\Delta > 0$, then for infinitely many k, we have $\operatorname{rank}(\mathcal{A}_{\sqrt{k}}) \geq k^{1+\delta}$. ◀

If one wants to obtain improved matrix rigidity under weaker assumptions, then the following corollary holds, similar to Corollary 18.

▶ **Corollary 23.** *If* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{0.85n})$, then, for some $\Delta > 0$, there exist two generators $g_1 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k}$ and $g_2 \colon \{0,1\}^{\log^{O(1)} k} \to \mathbb{F}^{k \times k \times k}$, computable in time polynomial in k, such that, for infinitely many k, at least one of the following conditions holds:*

- *$g_1(s)$ has $k^{0.7}$-rigidity $k^{1.4}$, for some s;*
- *$\operatorname{rank}(g_2(s))$ is at least $k^{1+\Delta}$, for some s.*

We are now ready to prove our conditional answer to the open problem from [41]. At first, recall the result of [42].

▶ **Theorem 24** (Theorem 4.4 in [42])**.** *If T is an n-by-n matrix that has rigidity m^3 for rank m, then the corresponding bilinear function F requires canonical circuits of size $\exp(m)$.*

Then, the following corollary holds.

▶ **Corollary 8.** *If, for every $\varepsilon > 0$,* MAX-*3*-SAT *cannot be solved in co-nondeterministic time $O(2^{(1-\varepsilon)n})$, then, for any $\delta > 0$, one can construct an explicit family of $2^{\log^{O(1)} n}$ functions such that, for infinitely many n, at least one of them is either bilinear and requires canonical circuits of size $2^{\Omega(n^{2/3-\delta})}$ or trilinear and requires arithmetic circuits of size $\Omega(n^{1.25})$.*

Proof. It suffices to verify that $\alpha = \frac{2}{3}$ and $\beta = 1.25$ satisfy the inequalities in the statement of Theorem 21 for any possible value of $\omega < 2.38$. Then, Theorem 24 provides the desired bound on the canonical circuit size. ◀

References

1 Boris Alexeev, Michael A. Forbes, and Jacob Tsimerman. Tensor rank: Some lower and upper bounds. In *CCC*, pages 283–291. IEEE Computer Society, 2011. doi:10.1109/CCC.2011.28.

2 Josh Alman and Lijie Chen. Efficient construction of rigid matrices using an NP oracle. In *FOCS*, pages 1034–1055. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00067.

3 Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *SODA*, pages 2005–2039. SIAM, 2025. doi:10.1137/1.9781611978322.63.

4 Josh Alman and Jingxun Liang. Low rank matrix rigidity: Tight lower bounds and hardness amplification. In *STOC*, pages 1383–1394. ACM, 2025. doi:10.1145/3717823.3718287.

5 Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. *TheoretiCS*, 3, 2024. doi:10.46298/THEORETICS.24.21.

6 Josh Alman and R. Ryan Williams. Probabilistic rank and matrix rigidity. In *STOC*, pages 641–652. ACM, 2017. doi:10.1145/3055399.3055484.

7 Noga Alon and Ravi B. Boppana. The monotone circuit complexity of Boolean functions. *Comb.*, 7(1):1–22, 1987. doi:10.1007/BF02579196.

8 Prashanth Amireddy, Ankit Garg, Neeraj Kayal, Chandan Saha, and Bhargav Thankey. Low-depth arithmetic circuit lower bounds: Bypassing set-multilinearization. In *ICALP*, volume 261 of *LIPIcs*, pages 12:1–12:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.12.

9 Alexander E. Andreev. A method for obtaining lower bounds on the complexity of individual monotone functions. *Doklady Akademii Nauk*, 282(5):1033–1037, 1985.

10 Alexander E. Andreev. A method for obtaining efficient lower bounds for monotone complexity. *Algebra and Logic*, 26(1):1–18, 1987.

11 Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theor. Comput. Sci.*, 22:317–330, 1983. doi:10.1016/0304-3975(83)90110-X.

12 Tatiana Belova, Alexander S. Kulikov, Ivan Mihajlin, Olga Ratseeva, Grigory Reznikov, and Denil Sharipov. Computations with polynomial evaluation oracle: ruling out superlinear SETH-based lower bounds. In *SODA*, pages 1834–1853. SIAM, 2024. doi:10.1137/1.9781611977912.73.

13 Amey Bhangale, Prahladh Harsha, Orr Paradise, and Avishay Tal. Rigid matrices from rectangular PCPs. *SIAM J. Comput.*, 53(2):480–523, 2024. doi:10.1137/22M1495597.

14 C. S. Bhargav, Sagnik Dutta, and Nitin Saxena. Improved lower bound, and proof barrier, for constant depth algebraic circuits. *ACM Trans. Comput. Theory*, 16(4):23:1–23:22, 2024. doi:10.1145/3689957.

15 Vishwas Bhargava, Sumanta Ghosh, Mrinal Kumar, and Chandra Kanta Mohapatra. Fast, algebraic multivariate multipoint evaluation in small characteristic and applications. *J. ACM*, 70(6):42:1–42:46, 2023. doi:10.1145/3625226.

16 Andreas Björklund, Radu Curticapean, Thore Husfeldt, Petteri Kaski, and Kevin Pratt. Fast deterministic chromatic number under the asymptotic rank conjecture. In *SODA*, pages 2804–2818. SIAM, 2025. doi:10.1137/1.9781611978322.91.

17 Andreas Björklund and Petteri Kaski. The asymptotic rank conjecture and the set cover conjecture are not both true. In *STOC*, pages 859–870. ACM, 2024. doi:10.1145/3618260.3649656.

18 Markus Bläser. A $\frac{5}{2}n^2$-lower bound for the rank of $n \times n$ matrix multiplication over arbitrary fields. In *FOCS*, pages 45–50. IEEE Computer Society, 1999.

19 Markus Bläser. Fast matrix multiplication. *Theory Comput.*, 5:1–60, 2013. doi:10.4086/TOC.GS.2013.005.

20 Jaroslaw Blasiok and Linus Meierhöfer. Hardness of clique approximation for monotone circuits. In *CCC*, volume 339 of *LIPIcs*, pages 4:1–4:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CCC.2025.4.

21 Nader H. Bshouty. A lower bound for matrix multiplication. *SIAM J. Comput.*, 18(4):759–765, 1989. doi:10.1137/0218052.

22 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *ITCS*, pages 261–270. ACM, 2016. doi:10.1145/2840728.2840746.

23 Bruno Pasqualotto Cavalar, Mika Göös, Artur Riazanov, Anastasia Sofronova, and Dmitry Sokolov. Monotone circuit complexity of matching. *Electron. Colloquium Comput. Complex.*, TR25-102, 2025. URL: https://eccc.weizmann.ac.il/report/2025/102.

24 Bruno Pasqualotto Cavalar, Mrinal Kumar, and Benjamin Rossman. Monotone circuit lower bounds from robust sunflowers. *Algorithmica*, 84(12):3655–3685, 2022. doi:10.1007/S00453-022-01000-3.

25 Lijie Chen, Shuichi Hirahara, and Hanlin Ren. Symmetric exponential time requires near-maximum circuit size. In *STOC*, pages 1990–1999. ACM, 2024. doi:10.1145/3618260.3649624.

26 Lijie Chen and Xin Lyu. Inverse-exponential correlation bounds and extremely rigid matrices from a new derandomized XOR lemma. In *STOC*, pages 761–771. ACM, 2021. doi:10.1145/3406325.3451132.

27 Lijie Chen, Ron D. Rothblum, Roei Tell, and Eylon Yogev. On exponential-time hypotheses, derandomization, and circuit lower bounds. *J. ACM*, 70(4):25:1–25:62, 2023. doi:10.1145/3593581.

28 Nikolai Chukhin, Alexander S. Kulikov, Ivan Mihajlin, and Arina Smirnova. Conditional complexity hardness: Monotone circuit size, matrix rigidity, and tensor rank under NSETH and beyond. *Electron. Colloquium Comput. Complex.*, TR25-038, 2025. URL: https://eccc.weizmann.ac.il/report/2025/038.

29 Gil Cohen, Ivan Bjerre Damgård, Yuval Ishai, Jonas Kölker, Peter Bro Miltersen, Ran Raz, and Ron D. Rothblum. Efficient multiparty protocols via log-depth threshold formulae - (extended abstract). In *CRYPTO (2)*, volume 8043 of *Lecture Notes in Computer Science*, pages 185–202. Springer, 2013. doi:10.1007/978-3-642-40084-1_11.

30 Zeev Dvir. On matrix rigidity and locally self-correctable codes. *Comput. Complex.*, 20(2):367–388, 2011. doi:10.1007/S00037-011-0009-1.

31 Zeev Dvir and Benjamin L. Edelman. Matrix rigidity and the Croot-Lev-Pach Lemma. *Theory Comput.*, 15:1–7, 2019. doi:10.4086/TOC.2019.V015A008.

32 Zeev Dvir, Alexander Golovnev, and Omri Weinstein. Static data structure lower bounds imply rigidity. In *STOC*, pages 967–978. ACM, 2019. doi:10.1145/3313276.3316348.

33 Zeev Dvir and Allen Liu. Fourier and circulant matrices are not rigid. *Theory Comput.*, 16:1–48, 2020. doi:10.4086/TOC.2020.V016A020.

34 Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-3n lower bound for the circuit complexity of an explicit function. In *FOCS*, pages 89–98. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.19.

35 Matthias Fitzi and Ueli M. Maurer. Efficient byzantine agreement secure against general adversaries. In *DISC*, volume 1499 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 1998. doi:10.1007/BFB0056479.

36 Michael A. Forbes. Low-depth algebraic circuit lower bounds over any field. In *CCC*, volume 300 of *LIPIcs*, pages 31:1–31:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CCC.2024.31.

37 Joel Friedman. A note on matrix rigidity. *Comb.*, 13(2):235–239, 1993. doi:10.1007/BF01303207.

38 Karthik Gajulapalli, Alexander Golovnev, Satyajeet Nagargoje, and Sidhant Saraogi. Range avoidance for constant depth circuits: Hardness and algorithms. In *APPROX/RANDOM*, volume 275 of *LIPIcs*, pages 65:1–65:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.65.

39 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory Comput.*, 16:1–30, 2020. doi:10.4086/TOC.2020.V016A013.

40 Oded Goldreich. Improved bounds on the AN-complexity of $O(1)$-linear functions. *Comput. Complex.*, 31(2):7, 2022. doi:10.1007/S00037-022-00224-7.

41 Oded Goldreich and Avishay Tal. Matrix rigidity of random Toeplitz matrices. *Comput. Complex.*, 27(2):305–350, 2018. doi:10.1007/S00037-016-0144-9.

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

43 Alexander Golovnev, Alexander S. Kulikov, and R. Ryan Williams. Circuit depth reductions. In *ITCS*, volume 185 of *LIPIcs*, pages 24:1–24:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.24.

44 Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *ITCS*, volume 124 of *LIPIcs*, pages 38:1–38:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ITCS.2019.38.

45 Dima Grigoriev. Application of separability and independence notions for proving lower bounds of circuit complexity. *Journal of Soviet Mathematics*, 14:1450–1457, 1980.

46 Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *STOC*, pages 577–582. ACM, 1998. doi:10.1145/276698.276872.

47 Dima Grigoriev and Alexander Razborov. Exponential complexity lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. In *FOCS*, pages 269–278. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743456.

48 Venkatesan Guruswami, Xin Lyu, and Xiuhan Wang. Range avoidance for low-depth circuits and connections to pseudorandomness. In *APPROX/RANDOM*, volume 245 of *LIPIcs*, pages 20:1–20:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.20.

49 Danny Harnik and Ran Raz. Higher lower bounds on monotone size. In *STOC*, pages 378–387. ACM, 2000. doi:10.1145/335305.335349.

50 Juris Hartmanis and Richard E Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

51 Johan Håstad. Almost optimal lower bounds for small depth circuits. *Adv. Comput. Res.*, 5:143–170, 1989.

52 Johan Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990. doi:10.1016/0196-6774(90)90014-6.

53 F. C. Hennie and Richard E Stearns. Two-tape simulation of multitape turing machines. *J. ACM*, 13(4):533–546, 1966. doi:10.1145/321356.321362.

54 Martin Hirt and Ueli M. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In *PODC*, pages 25–34. ACM, 1997. doi:10.1145/259380.259412.

55 Martin Hirt and Ueli M. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *J. Cryptol.*, 13(1):31–60, 2000. doi:10.1007/S001459910003.

56 Pavel Hrubes and Pavel Pudlák. Random formulas, monotone circuits, and interpolation. In *FOCS*, pages 121–131. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.20.

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

58 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/JCSS.2001.1774.

59 Hamidreza Jahanjou, Eric Miles, and Emanuele Viola. Local reduction. *Inf. Comput.*, 261:281–295, 2018. doi:10.1016/J.IC.2018.02.009.

60 Stasys Jukna. Combinatorics of monotone computations. *Comb.*, 19(1):65–85, 1999. doi:10.1007/S004930050046.

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

62 Richard M. Karp and Richard J. Lipton. Some connections between nonuniform and uniform complexity classes. In *STOC*, pages 302–309. ACM, 1980. doi:10.1145/800141.804678.

63 Oliver Korten. The hardest explicit construction. In *FOCS*, pages 433–444. IEEE, 2021. doi:10.1109/FOCS52979.2021.00051.

64 Alexander Kozachinskiy and Vladimir V. Podolskii. Multiparty Karchmer-Wigderson games and threshold circuits. *Theory Comput.*, 18:1–33, 2022. doi:10.4086/TOC.2022.V018A015.

65 J. M. Landsberg. New lower bounds for the rank of matrix multiplication. *SIAM J. Comput.*, 43(1):144–149, 2014. doi:10.1137/120880276.

66 Jiatu Li and Tianqi Yang. $3.1n - o(n)$ circuit lower bounds for explicit functions. In *STOC*, pages 1180–1193. ACM, 2022. doi:10.1145/3519935.3519976.

67 Zeyong Li. Symmetric exponential time requires near-maximum circuit size: Simplified, truly uniform. In *STOC*, pages 2000–2007. ACM, 2024. doi:10.1145/3618260.3649615.

68 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. *J. ACM*, 72(4):26:1–26:35, 2025. doi:10.1145/3734215.

69 Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In *SODA*, pages 1236–1252. SIAM, 2018. doi:10.1137/1.9781611975031.80.

70 Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Found. Trends Theor. Comput. Sci.*, 4(1-2):1–155, 2009. doi:10.1561/0400000011.

71 Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with sunflowers. In *ITCS*, volume 215 of *LIPIcs*, pages 104:1–104:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.104.

72 Alex Massarenti and Emanuele Raviolo. The rank of $n \times n$ matrix multiplication is at least $3n^2 - 2\sqrt{2}n^{\frac{3}{2}} - 3n$. *Linear Algebra and its Applications*, 438(11):4500–4509, 2013.

73 Alex Massarenti and Emanuele Raviolo. Corrigendum to "The rank of $n \times n$ matrix multiplication is at least $3n^2 - 2\sqrt{2}n^{\frac{3}{2}} - 3n$". *Linear Algebra and its Applications*, 445:369–371, 2014.

74 Jesper Nederlof. Bipartite TSP in $O(1.9999^n)$ time, assuming quadratic time matrix multiplication. In *STOC*, pages 40–53. ACM, 2020.

75 Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364, 2005. doi:10.1145/1066100.1066101.

76 Nicholas Pippenger and Michael J. Fischer. Relations among complexity measures. *J. ACM*, 26(2):361–381, 1979. doi:10.1145/322123.322138.

77 Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In *STOC*, pages 1246–1255. ACM, 2017. doi:10.1145/3055399.3055478.

78 Kevin Pratt. A stronger connection between the asymptotic rank conjecture and the set cover conjecture. In *STOC*, pages 871–874. ACM, 2024. doi:10.1145/3618260.3649620.

79 Pavel Pudlák and Zdeněk Vavřín. Computation of rigidity of order $\frac{n^2}{r}$ for one simple matrix. *Commentationes Mathematicae Universitatis Carolinae*, 32(2):213–218, 1991.

80 Sivaramakrishnan Natarajan Ramamoorthy and Cyrus Rashtchian. Equivalence of systematic linear data structures and matrix rigidity. In *ITCS*, volume 151 of *LIPIcs*, pages 35:1–35:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.35.

81 C. Ramya. Recent progress on matrix rigidity - A survey. *CoRR*, abs/2009.09460, 2020. arXiv:2009.09460.

82 Ran Raz. On the complexity of matrix product. *SIAM J. Comput.*, 32(5):1356–1369, 2003. doi:10.1137/S0097539702402147.

83 Ran Raz and Amir Shpilka. Lower bounds for matrix product in bounded depth circuits with arbitrary gates. *SIAM J. Comput.*, 32(2):488–513, 2003. doi:10.1137/S009753970138462X.

84 Alexander Razborov. Lower bounds on the monotone complexity of some Boolean function. In *Soviet Math. Dokl.*, volume 31, pages 354–357, 1985.

85 Alexander Razborov. Lower bounds for the size of circuits of bounded depth with basis (AND, XOR). *Math. notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

86 Alexander Razborov. On rigid matrices. *Manuscript (in russian)*, 1989.

87 Arnold Schönhage. Partial and total matrix multiplication. *SIAM J. Comput.*, 10(3):434–455, 1981. doi:10.1137/0210032.

88 Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell Syst. Tech. J.*, 28(1):59–98, 1949. doi:10.1002/J.1538-7305.1949.TB03624.X.

89 Mohammad Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. A remark on matrix rigidity. *Inf. Process. Lett.*, 64(6):283–285, 1997. doi:10.1016/S0020-0190(97)00190-7.

90 Amir Shpilka. Lower bounds for matrix product. *SIAM J. Comput.*, 32(5):1185–1200, 2003. doi:10.1137/S0097539702405954.

91 Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *STOC*, pages 77–82. ACM, 1987. doi:10.1145/28395.28404.

92 Volker Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.

93 Volker Strassen. Die berechnungskomplexität von elementarsymmetrischen funktionen und von interpolationskoeffizienten. *Numerische Mathematik*, 20:238–251, 1973.

94 Volker Strassen. Die berechnungskomplexität der symbolischen differentiation von interpolationspolynomen. *Theor. Comput. Sci.*, 1(1):21–25, 1975. doi:10.1016/0304-3975(75)90010-9.

95 Volker Strassen. The asymptotic spectrum of tensors and the exponent of matrix multiplication. In *FOCS*, pages 49–54. IEEE Computer Society, 1986. doi:10.1109/SFCS.1986.52.

96 Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *MFCS*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977. doi:10.1007/3-540-08353-7_135.

97 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the international congress of mathematicians: Rio de janeiro 2018*, pages 3447–3487. World Scientific, 2018.

98 Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *SODA*, pages 3792–3835. SIAM, 2024. doi:10.1137/1.9781611977912.134.

99 Ben Lee Volk and Mrinal Kumar. A polynomial degree bound on equations for non-rigid matrices and small linear circuits. *ACM Trans. Comput. Theory*, 14(2):6:1–6:14, 2022. doi:10.1145/3543685.

100 R. Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. doi:10.1016/J.TCS.2005.09.023.

101 R. Ryan Williams. *Algorithms and resource requirements for fundamental problems*. PhD thesis, Carnegie Mellon University, 2007.

102 R. Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM J. Comput.*, 42(3):1218–1244, 2013. doi:10.1137/10080703X.

103 Ryan Williams. The orthogonal vectors conjecture and non-uniform circuit lower bounds. In *FOCS*, pages 1372–1387. IEEE, 2024. doi:10.1109/FOCS61266.2024.00088.

104 Henning Wunderlich. On a theorem of Razborov. *Comput. Complex.*, 21(3):431–477, 2012. doi:10.1007/S00037-011-0021-5.