Electronic Colloquium on Computational Complexity, Report No. 102 (2025)

Monotone Circuit Complexity of Matching

Bruno Cavalar
University of Oxford

Mika Göös
EPFL

Artur Riazanov
EPFL

Anastasia Sofronova
EPFL

Dmitry Sokolov
EPFL

July 22, 2025

Abstract

We show that the perfect matching function on n-vertex graphs requires monotone circuits of size $2^{n^{\Omega(1)}}$. This improves on the $n^{\Omega(\log n)}$ lower bound of Razborov (1985). Our proof uses the standard approximation method together with a new sunflower lemma for matchings.

1 Introduction

A sobering lesson learned already in the 1980s [Raz85a, Tar88] is that general boolean circuits (using gates $\wedge$, $\vee$, $\neg$) can be much more powerful than monotone circuits (using gates $\wedge$, $\vee$). The earliest demonstration is due to Razborov [Raz85a]. He considered the *bipartite perfect matching* function $\textsc{Match}\colon \{0,1\}^{n^2} \to \{0,1\}$ that takes as input a bipartite graph, represented by its adjacency matrix $x \in \{0,1\}^{n\times n}$, and outputs $\textsc{Match}(x) = 1$ iff the graph contains a perfect matching. While bipartite matching famously admits polynomial-size circuits, Razborov showed that it requires monotone circuits of size $n^{\Omega(\log n)}$. Since then, a long-standing challenge has been to determine whether Razborov's quasi-polynomial bound is tight (e.g., see textbooks [Weg87, Juk12, Wig19]). Our main result is to improve the lower bound to an exponential one.

Theorem 1. $\textsc{Match}$ *requires monotone circuits of size at least* $2^{n^{1/3-o(1)}}$.

That is, for bipartite matching, the gap between the general and monotone circuit complexities is exponential. In fact, such an exponential gap was already known for a different monotone function in class P due to Tardos [Tar88]. Her function is relatively complex, however, as it is computed by solving a semidefinite program. Meanwhile, bipartite matching admits an efficient parallel algorithm (class RNC) [Lov79, Mul87], which is not known for Tardos's function.

Another serious contender for exhibiting the strongest general-vs-monotone separation is $\mathbb{Z}_2$-*satisfiability* [GKRS19]. This is a monotone function encoding the problem "Given a system of linear equations over $\mathbb{Z}_2$, is it satisfiable?" It is complete for the class $\oplus\mathsf{L}$ of problems computed by uniform polynomial-size parity branching programs [Dam90]. Yet, $\mathbb{Z}_2$-satisfiability was shown to require exponential-size monotone circuits by [GGKS20, GKRS19]. Bipartite matching is not known to be comparable to $\mathbb{Z}_2$-satisfiability under deterministic reductions. However, *non-uniformly*, bipartite matching lies in a subclass $\mathsf{SPL} \subseteq \oplus\mathsf{L}$ [ARZ99] and hence is arguably simpler than $\mathbb{Z}_2$-satisfiability.

ISSN 1433-8092

Function	**Class**	**Monotone complexity**	**Reference**
Bipartite matching	RNC	$\exp(\Omega(\log^2 n))$	[Raz85a]
Tardos's function	P	$\exp(n^{\Omega(1)})$	[AB87, Tar88]
Odd factor	L	$\exp(\Omega(\log^2 n))$	[BGW99]
$\mathbb{Z}_2$-satisfiability	$\oplus\mathsf{L}$	$\exp(n^{\Omega(1)})$	[GGKS20, GKRS19]
$\mathbb{Z}_2$-satisfiability, padded	$\mathsf{AC}^0[\oplus]$	$\exp(\Omega(\log^k n))$	[CO23]
Bipartite matching	RNC	$\exp(n^{\Omega(1)})$	This work (Theorem 1)
Odd factor	L	$\exp(n^{\Omega(1)})$	This work (Theorem 2)
Odd factor, padded	AC^0	$\exp(\Omega(\log^k n))$	This work (Theorem 3)

Table 1: Timeline of separations between general and monotone complexities. The parameter k can be taken to be any large constant at the cost of increasing the depth of the AC^0 circuit.

In fact, our proof of Theorem 1 can be extended further to prove a lower bound for a function even simpler than bipartite matching, called *odd factor* [BGW99]. This function is defined by $\text{ODD}(x) = 1$ iff the graph $x \in \{0,1\}^{n\times n}$ contains a spanning subgraph whose degrees are all odd. Equivalently, $\text{ODD}(x) = 1$ iff every connected component of x has even size. This function can be computed in logarithmic space (class L) using Reingold's algorithm [Rei08]. We show monotone lower bounds for ODD as well as "padded" versions of it that can be computed by one of the simplest of all circuit models: constant-depth circuits (class AC^0).

Theorem 2. *There is a monotone* $\text{ODD} \in \mathsf{L}$ *with monotone circuit complexity* $2^{n^{\Omega(1)}}$.

Theorem 3. *For any k there is a monotone* $f_k \in \mathsf{AC}^0$ *with monotone circuit complexity* $n^{\Omega(\log^k n)}$.

In particular, Theorem 3 resolves an open problem of Grigni and Sipser [GS92], who asked if every monotone function in AC^0 can be computed by a polynomial-size monotone circuit. Theorem 3 also rules out a particular approach to obtaining general circuit lower bounds: the papers [CHO$^+$22, CO23] observed that if Theorem 3 had turned out to be false, then $\mathsf{NC}^2 \not\subseteq \mathsf{NC}^1$.

1.1 Technique: Matching sunflowers

We follow the classic *approximation method* introduced by Razborov [Raz85a, Raz85b]. By now, this standard method is featured in several textbooks [Weg87, AB09, Juk12, Wat25]. To prove a lower bound for an n-bit boolean function f, the method starts by defining a distribution $\mathcal{D}$ over the input domain $\{0,1\}^n$. The goal is to show that (i) if f is computed by a small monotone circuit, then f can be approximately computed (relative to $\mathcal{D}$) by a small monotone DNF; and (ii) no small DNF correlates well with f.

The technical crux of the proof is to identify situations when a monotone DNF $\bigvee_{S\in\mathcal{S}} t_S$ (where $t_S \coloneqq \bigwedge_{i\in S} x_i$ and $\mathcal{S} \subseteq 2^{[n]}$) can be safely replaced with the single term t_K where $K \coloneqq \bigcap \mathcal{S}$ is the *core*, and doing so does not incur much error (relative to $\mathcal{D}$). This replacement procedure, often called "plucking", simplifies the DNF in case $|\mathcal{S}| \geq 2$. Previous works [Ros14, CKR20, BM25] have employed the notion of "robust sunflowers" to find such DNFs for plucking. A family $\mathcal{S} \subseteq 2^{[n]}$, $|\mathcal{S}| \geq 2$, is an *ε-robust sunflower* if the core $K = \bigcap \mathcal{S}$ satisfies

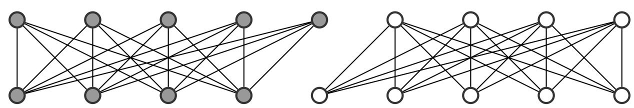
$$\Pr_{\boldsymbol{x}\sim\{0,1\}^n} [\, \exists S \in \mathcal{S} \colon t_{S\setminus K}(\boldsymbol{x}) = 1 \,] \geq 1 - \varepsilon. \tag{1}$$

This says that, for a uniform random $\boldsymbol{x}$, whenever t_K accepts $\boldsymbol{x}$, it is highly likely that $\bigvee_{S \in \mathcal{S}} t_S$ accepts it too. That is, the approximation error is small. Recent works [ALWZ21, Rao20] have proved optimal bounds on the size of families $\mathcal{S}$ that are guaranteed to contain a subfamily $\mathcal{S}' \subseteq \mathcal{S}$ that is a robust sunflower.

Lemma 1 (Robust Sunflower Lemma [Rao20])**.** *There exists a universal constant $c > 0$ such that every family $\mathcal{S}$ of ℓ-sets of size $|\mathcal{S}| \geq (c \log(\ell/\varepsilon))^\ell$ contains an ε-robust sunflower.*

For our purposes, we need instead a sunflower lemma tailored to the bipartite matching problem. The first difference is that, instead of a uniform distribution, we will pluck relative to the following *odd cut* distribution $\mathcal{D}_0$ over $\text{Match}^{-1}(0)$ (which Razborov [Raz85a] also used).

Definition 1 (Odd cut distribution $\mathcal{D}_0$)**.** To sample $\boldsymbol{x} \sim \mathcal{D}_0$, first sample a uniform random colouring $\boldsymbol{c} \in \{0,1\}^{2n}$ of the vertices of $K_{n,n}$ with an odd number of 1s. To build the bipartite graph $\boldsymbol{x}$, connect any two vertices (on opposite sides) that have the same colour under $\boldsymbol{c}$. The resulting graph is a union of two odd-sized bicliques:



Suppose $\mathcal{M}$ is a family of ℓ-matchings (each matching has ℓ edges) in $K_{n,n}$. We say that the family $\mathcal{M}$, $|\mathcal{M}| \geq 2$, is an *ε-matching sunflower* if the core $K = \bigcap \mathcal{M}$ satisfies

$$\Pr[\,\exists M \in \mathcal{M}, \forall e \in M \setminus K : e \text{ is monochromatic under } \boldsymbol{c}\,] \geq 1 - \varepsilon. \tag{2}$$

This says that, for an input $\boldsymbol{x} \sim \mathcal{D}_0$, whenever t_K accepts $\boldsymbol{x}$, it is highly likely that $\bigvee_{M \in \mathcal{M}} t_M$ accepts it too. That is, the approximation error is small. We prove the following in Section 2.

Lemma 2 (Matching Sunflower Lemma)**.** *There exists a universal constant $c > 0$ such that every family $\mathcal{M}$ of ℓ-matchings of size $|\mathcal{M}| \geq (c\ell \log^2(\ell/\varepsilon))^\ell$ contains an ε-matching sunflower.*

Our proof is surprisingly simple: it is by a *reduction* to the robust sunflower lemma. Implicit in the original proof of Razborov [Raz85a] is that one can take $|\mathcal{M}| \geq 4^{\ell^2}(c\ell \log(1/\varepsilon))^{2\ell}$ in the above lemma. This is exponentially worse in terms of ℓ. Our improvement above is what directly translates into an exponential monotone circuit lower bound. We discuss in Section 3 how Lemma 2 plugs into the standard approximation method to prove Theorems 1–3.

1.2 Other related work

The analogue of Theorem 1 for monotone *formulas* has been known for a long time. Raz and Wigderson [RW92] proved that bipartite matching requires monotone formulas of size $2^{\Omega(n)}$ and this is tight. The same lower bound holds for odd factor [BGW99] and this was shown to imply quasi-polynomial lower bounds for the AC^0-computable padded odd factor in [CO23].

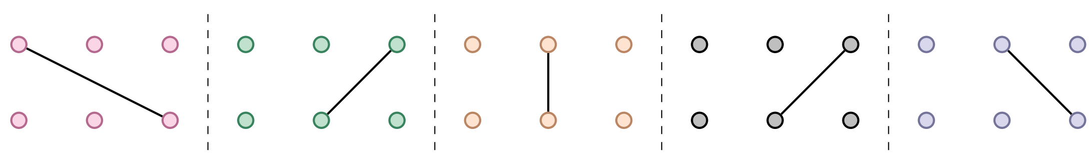
Previous works have also studied general-vs-monotone separations in the setting of constant-depth circuits. Okol'nishnikova [Oko82] and Ajtai and Gurevich [AG87] exhibited a monotone function in AC^0 that requires monotone constant-depth circuits of size $n^{\omega(1)}$. This was quantitatively improved by Chen, Oliveira, and Servedio [COS17] showing a lower bound of $2^{n^{\Omega(1)}}$. Our Theorem 3 thus improves qualitatively on [Oko82, AG87, CO23] but is incomparable to [COS17]. It remains open whether there exists a monotone function in AC^0 with exponential monotone circuit complexity. This would be an ultimate general-vs-monotone separation, generalising all the aforementioned results. The analogous question for *arithmetic* circuits was only recently settled [CDM21].

Besides general-vs-monotone separations, another foremost goal in monotone complexity is to find explicit functions with maximal monotone circuit complexities. The clique function has been studied the most [Raz85a, AB87, Juk99, Ros14, CKR20, LMM+22, BM25, dRV25]. Currently, the largest explicit lower bound is $2^{n^{1/2-o(1)}}$ [CKR20] with previous records being held by [And87, HR00]. The question of proving "truly" exponential lower bounds of the form $2^{\Omega(n)}$ remains open. It has been solved for monotone *formulas* by Pitassi and Robere [PR17].

2 Matching Sunflower Lemma (Lemma 2)

In this section, we prove Lemma 2. Suppose $\mathcal{M}$ is a family of ℓ-matchings with $|\mathcal{M}| \geq (c\ell \log^2(\ell/\varepsilon))^\ell$, where c is a large enough constant (to be determined). Our goal is to show that $\mathcal{M}$ contains an ε-matching sunflower. We start by simplifying the family $\mathcal{M}$ by making it "blocky".

Reduction to blocky families. Consider partitioning the vertices of $K_{n,n}$ into ℓ blocks according to a random labelling $\boldsymbol{b} \sim [\ell]^{2n}$. We say that a matching M is *consistent* with $\boldsymbol{b}$ if every edge $uv \in M$ is monochromatic under $\boldsymbol{b}$ (that is, $\boldsymbol{b}_u = \boldsymbol{b}_v$) and all edges receive distinct labels. Here is an illustration (with blocks of the same size, for simplicity):



For a fixed ℓ-matching $M \in \mathcal{M}$, there are $\ell^{2\ell}$ ways of labelling its endpoints, and $\ell!$ of these yield a consistent labelling. Thus $\Pr[\, M \text{ is consistent with } \boldsymbol{b}\,] = \ell!/\ell^{2\ell} \geq \ell^{-\ell}2^{-O(\ell)}$. By averaging, there exists a fixed labelling b that is consistent with at least $|\mathcal{M}|\ell^{-\ell}2^{-O(\ell)}$ matchings in $\mathcal{M}$. Let us delete all matchings inconsistent with b, and continue to denote the resulting set by $\mathcal{M}$ for simplicity. For large enough c, the number of remaining matchings is $|\mathcal{M}| \geq (c' \log(4\ell/\varepsilon))^{2\ell}$, where c' is the universal constant from the robust sunflower lemma (Lemma 1).

Finding a "vertex" sunflower. Define $\mathcal{V} := \{V(M) : M \in \mathcal{M}\}$, $V(M) := \bigcup M$, as the family of endpoints of matchings in $\mathcal{M}$. Since $\mathcal{M}$ is blocky, $\mathcal{V}$ and $\mathcal{M}$ are in 1-to-1 correspondence: for every $V \in \mathcal{V}$ there is a *unique* matching $M \in \mathcal{M}$ with $V = V(M)$. Thus $\mathcal{V}$ is a family of 2ℓ-sets of size $|\mathcal{V}| = |\mathcal{M}| \geq (c' \log(4\ell/\varepsilon))^{2\ell}$. We can now apply Lemma 1 to find an $\varepsilon/2$-robust sunflower $\mathcal{V}' \subseteq \mathcal{V}$ ("vertex" sunflower) with core $K := \bigcap \mathcal{V}'$. Note that every block contains 0, 1, or 2 vertices from K. Let K_1 be the vertices in K that are unique in their block, and K_2 be the vertices that share a block with another vertex, so $K = K_1 \sqcup K_2$.

Finding a matching sunflower. Let $\mathcal{M}' \subseteq \mathcal{M}$ be the matchings corresponding to vertex sets $\mathcal{V}'$ according to the 1-to-1 correspondence. Let D be the matching that connects pairs of vertices in K_2 that share a block. The following claim completes the proof.

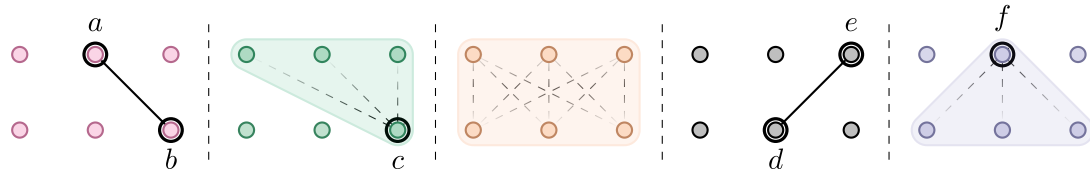


Figure 1: Matching sunflower $\mathcal{M}' \subseteq \mathcal{M}$ with core $D = \{ab, de\}$ constructed out of a vertex sunflower $\mathcal{V}' \subseteq \mathcal{V}$ with core $K = \bigcap \mathcal{V}' = K_1 \sqcup K_2$ where $K_1 = \{c, f\}$ and $K_2 = \{a, b, d, e\}$. The shaded regions indicate where non-core edges of matchings in $\mathcal{M}'$ can occur.

Claim 1. $\mathcal{M}'$ is an ε-matching sunflower (with core $D = \bigcap \mathcal{M}'$).

Proof. First note that $|\mathcal{M}'| = |\mathcal{V}'| \geq 2$. Let us then check that $D = \bigcap \mathcal{M}'$. We have $D \subseteq \bigcap \mathcal{M}'$ since, for every edge $uv \in D$ and $M \in \mathcal{M}'$, the endpoints u, v belong to $V(M)$ and share a block, hence $uv \in M$. Conversely, suppose for contradiction there is an edge $e \in (\bigcap \mathcal{M}') \setminus D$. Then every $V \in \mathcal{V}'$ contains e as a subset. Since $e \notin D$, at least one endpoint is outside K, say $u \in e \setminus K$. Then

$$\Pr_{\boldsymbol{x} \sim \{0,1\}^{2n}} [\exists V \in \mathcal{V}' \colon t_{V \setminus K}(\boldsymbol{x}) = 1] \leq \Pr_{\boldsymbol{x} \sim \{0,1\}^{2n}} [\boldsymbol{x}_u = 1] = 1/2.$$

This contradicts (1) for $\mathcal{V}'$ (where we can assume $\varepsilon < 1/2$ wlog). We conclude that $D = \bigcap \mathcal{M}'$.

Let $\boldsymbol{x} \sim \{0,1\}^{2n}$ be a uniform colouring. Our goal will be to show

$$\Pr[\exists M \in \mathcal{M}', \forall uv \in M \setminus D \colon \boldsymbol{x}_u = \boldsymbol{x}_v] \geq 1 - \varepsilon/2. \tag{3}$$

This would conclude the proof, as conditioning on the event "$\boldsymbol{x}$ has odd many 1s" (which is what we really care about) can only double the error parameter. To prove (3), we show it holds under conditioning on any event "$\boldsymbol{x}_{K_1} = \alpha$" where $\alpha \in \{0,1\}^{K_1}$ (which partitions the probability space).

Consider first the simplest case $\boldsymbol{x}' := (\boldsymbol{x} \mid \boldsymbol{x}_{K_1} = 1_{K_1})$, where we condition all the colours in K_1 to be 1. Note that $\boldsymbol{x}$ and $\boldsymbol{x}'$ have the same (uniform) marginal distribution outside K. This means we can invoke the robust sunflower property of $\mathcal{V}'$ for $\boldsymbol{x}'$: with probability $1 - \varepsilon/2$ over $\boldsymbol{x}' = x'$ there exists $V = V(M) \in \mathcal{V}'$ such that $t_{V \setminus K}(x') = 1$. We claim that all edges $uv \in M \setminus D$ are coloured 1 under x'. Indeed, if $uv \cap K = \emptyset$, then both endpoints are coloured $x'_u = x'_v = 1$ by the sunflower property. Otherwise, say, $uv \cap K = \{v\}$. Here one endpoint is coloured $x'_u = 1$ by the sunflower property, and the other endpoint has $x'_v = 1$ because of our conditioning.

More generally, we can apply the same logic for $\boldsymbol{x}' := (\boldsymbol{x} \mid \boldsymbol{x}_{K_1} = \alpha)$ for any $\alpha \in \{0,1\}^{K_1}$. All we need to do is flip the colours in all blocks that contain $v \in K_1$ with $\alpha_v = 0$. Let $x^\alpha \in \{0,1\}^{2n}$ be the fixed colouring that assigns colour 1 to all blocks that contain $v \in K_1$ with $\alpha_v = 0$. Formally, $x^\alpha_v = 1$ iff $b(v) \in b(\{u \in K_1 : \alpha_u = 0\})$, so in particular $x^\alpha_{K_1} \oplus 1_{K_1} = \alpha$. Note that $\boldsymbol{x}' \oplus x^\alpha$ has a uniform marginal distribution outside K. This means we can invoke the robust sunflower property of $\mathcal{V}'$ for $\boldsymbol{x}' \oplus x^\alpha$: with probability $1 - \varepsilon/2$ over $\boldsymbol{x}' \oplus x^\alpha = x' \oplus x^\alpha$ there exists $V = V(M) \in \mathcal{V}'$ such that $t_{V \setminus K}(x' \oplus x^\alpha) = 1$. We claim that all edges $uv \in M \setminus D$ are monochromatic under x'. The case $uv \cap K = \emptyset$ is the same as above. For $uv \cap K = \{v\}$, we have from the sunflower property that $x'_u \oplus x^\alpha_u = 1$. This implies $x'_u = x^\alpha_u \oplus 1 = x^\alpha_v \oplus 1 = \alpha_v = x'_v$, as desired. $\square$

3 Approximation method (Theorems 1–3)

Consider the input distribution $\mathcal{D} \coloneqq (\mathcal{D}_0 + \mathcal{D}_1)/2$ where $\mathcal{D}_i$ is supported on $\textsc{Match}^{-1}(i)$ so that

- $\mathcal{D}_1$ is the uniform distribution over perfect matchings in $K_{n,n}$;
- $\mathcal{D}_0$ is the odd cut distribution from Definition 1.

To prove Theorem 1, we start with a small monotone circuit computing $\textsc{Match}$ and aim for a contradiction. Our goal is to approximate the circuit (relative to $\mathcal{D}$) with a small monotone DNF

$$F_{\mathcal{M}} \coloneqq \bigvee_{M \in \mathcal{M}} t_M \qquad \text{where} \qquad t_M \coloneqq \bigwedge_{e \in M} x_e,$$

and where $\mathcal{M}$ is a set of (partial) matchings in $K_{n,n}$. We say that $F = F_{\mathcal{M}}$ is *r-small* if, for every ℓ, $\mathcal{M}$ contains at most r^ℓ matchings of size ℓ, that is, $|\mathcal{M} \cap \mathcal{P}_\ell| \le r^\ell$ where $\mathcal{P}_\ell$ is the set of all ℓ-matchings. The approximation method proceeds in two steps:

Lemma 3. *Suppose a monotone circuit of size 2^w computes* $\textsc{Match}$. *Then there is an $O(w^3 \log^2 n)$-small monotone DNF F such that* $\Pr_{\boldsymbol{x}\sim\mathcal{D}}[F(\boldsymbol{x}) = \textsc{Match}(\boldsymbol{x})] \ge 1 - o(1)$.

Lemma 4. *If a monotone DNF F is $o(n)$-small, then* $\Pr_{\boldsymbol{x}\sim\mathcal{D}}[F(\boldsymbol{x}) = \textsc{Match}(\boldsymbol{x})] \le 1/2 + o(1)$.

These lemmas imply that any monotone circuit of size S for $\textsc{Match}$ has $\log^3 S \log^2 n \ge \Omega(n)$. Thus $S \ge \exp(n^{1/3 - o(1)})$ which proves Theorem 1. It remains to prove these two lemmas, which we do in Sections 3.1–3.2. Finally, we discuss how to derive Theorems 2–3 in Section 3.3.

3.1 Proof of Lemma 4

The lemma is trivially true if $F = F_{\mathcal{M}}$ contains the empty term (so that $F \equiv 1$). Otherwise,

$$\begin{aligned}
\Pr_{\boldsymbol{x}\sim\mathcal{D}}[F(\boldsymbol{x}) = \textsc{Match}(\boldsymbol{x})] &= \tfrac{1}{2} \Pr_{\boldsymbol{x}\sim\mathcal{D}_0}[F(\boldsymbol{x}) = 0] + \tfrac{1}{2} \Pr_{\boldsymbol{x}\sim\mathcal{D}_1}[F(\boldsymbol{x}) = 1] \\
&\le \tfrac{1}{2} + \sum_{\ell \in [n]} \sum_{M \in \mathcal{M} \cap \mathcal{P}_\ell} \Pr_{\boldsymbol{x}\sim\mathcal{D}_1}[t_M(\boldsymbol{x}) = 1] \\
&\le \tfrac{1}{2} + \sum_{\ell \in [n]} o(n)^\ell (e/n)^\ell \qquad\qquad (o(n)\text{-smallness and Claim 2 below}) \\
&\le \tfrac{1}{2} + \sum_{\ell \in \mathbb{N}\setminus\{0\}} o(1)^\ell \\
&\le \tfrac{1}{2} + o(1).
\end{aligned}$$

Claim 2. $\Pr_{\boldsymbol{x}\sim\mathcal{D}_1}[t_M(\boldsymbol{x}) = 1] \le (e/n)^\ell$ for every $M \in \mathcal{P}_\ell$.

Proof. The probability corresponds to the fraction of perfect matchings of $K_{n,n}$ that contain M. This is equal to $(n-\ell)!/n!$ and we verify by induction on ℓ and n that $(n-\ell)!/n! \le (e/n)^\ell$. Note that this holds for $\ell = 1$ and all $n \in \mathbb{N}$. When $n \ge \ell \ge 2$ we obtain

$$\frac{(n-\ell)!}{n!} = \frac{1}{n}\frac{(n-\ell)!}{(n-1)!} \le \frac{1}{n}\left(\frac{e}{n-1}\right)^{\ell-1} = \left(\frac{e}{n}\right)^\ell \frac{1}{e}\left(1 + \frac{1}{n-1}\right)^{\ell-1} \le \left(\frac{e}{n}\right)^\ell. \qquad \square$$

3.2 Proof of Lemma 3

Notation. Let C be a circuit with $\mathrm{size}(C) \le 2^w$ computing Match. We say that $F_{\mathcal{M}}$ has *width* k if every matching in $\mathcal{M}$ has at most k edges, and we say that $F_{\mathcal{M}}$ is a (k,r)*-DNF* if $F_{\mathcal{M}}$ is r-small and has width k. We set $\varepsilon := n^{-3w}$. Let $r(\ell,\varepsilon)$ be such that any set $\mathcal{M}$ of ℓ-matchings of size at least $r(\ell,\varepsilon)^\ell$ contains an ε-matching sunflower. Let $r := \max_{\ell \in [2w]} r(\ell,\varepsilon)$. From Lemma 2, we obtain that $r \le O(w \log^2(w/\varepsilon)) \le O(w^3 \log^2 n)$. We may assume here that $w^3 \log^2 n \le o(n)$ (so that $r \le o(n)$) as otherwise the result is trivial.

Proof overview (via plucking). We will construct a (w,r)-DNF for C gate-by-gate, inductively, starting at the input gates until we reach the output gate. Every input variable is already a (w,r)-DNF. Consider then an $\vee$ gate. The challenge is that if we were to naively combine our inductively constructed (w,r)-DNFs by $\vee$, the number of terms might increase, potentially violating r-smallness. For an $\wedge$ gate, a naive combination would also increase the width from w up to $2w$. In order to maintain smallness of our DNF, we approximate the naive combination by running Algorithm 1. The following claim summarises the properties of the resulting DNF.

Algorithm 1 Plucking procedure $\mathrm{pluck}(\mathcal{M})$

1: **while** $\exists \ell \in [2w] \colon |\mathcal{M} \cap \mathcal{P}_\ell| > r^\ell$ **do**
2: Let $\mathcal{M}' \subseteq \mathcal{M} \cap \mathcal{P}_\ell$ be an ε-matching sunflower with core K
3: Let $\mathcal{M} \leftarrow (\mathcal{M} \setminus \mathcal{M}') \cup \{K\}$
4: **end while**

Claim 3. If $F_{\mathcal{M}}$ has width $2w$, then $F_{\mathrm{pluck}(\mathcal{M})}$ is a $(2w,r)$-DNF with $F_{\mathrm{pluck}(\mathcal{M})} \ge F_{\mathcal{M}}$ and

$$\Pr_{\boldsymbol{x} \sim \mathcal{D}_0}[F_{\mathrm{pluck}(\mathcal{M})}(\boldsymbol{x}) > F_{\mathcal{M}}(\boldsymbol{x})] \le n^{-w}. \tag{4}$$

Proof. First note that Line 2 of the algorithm is always possible as $r \ge r(\ell,\varepsilon)$ for all $\ell \in [2w]$. Also note that in Line 3, the size of the family $\mathcal{M}$ decreases by at least one. This means that the algorithm will terminate in at most $|\mathcal{M}| \le n^{2w}$ iterations. Let us then verify (4) by calculating the errors incurred in Line 3. An error occurs for input x only if $t_K(x) = 1$ but $t_M(x) = 0$ for all $M \in \mathcal{M}'$ (see Figure 2). This can occur only if $t_{M \setminus K}(x) = 0$ for all $M \in \mathcal{M}'$. But the ε-matching sunflower property (2) for $\mathcal{M}'$ implies that this happens only with probability at most ε over $\boldsymbol{x} \sim \mathcal{D}_0$. A union bound over all iterations shows that (4) is at most $n^{2w}\varepsilon = n^{2w}n^{-3w} = n^{-w}$. $\square$

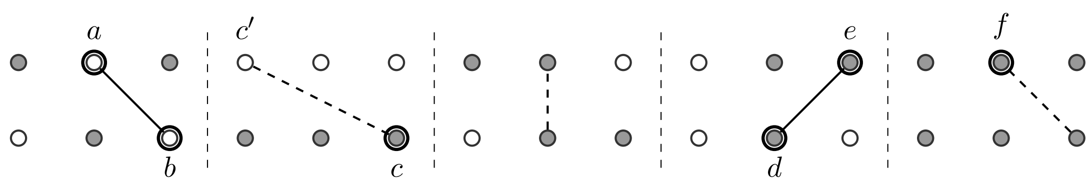


Figure 2: Example of an input $x \in \mathrm{supp}(\mathcal{D}_0)$ (given by the black/white vertex colouring) that causes an error when plucking a sunflower $\mathcal{M}'$ (from Figure 1). Edges ab and de in the core $K = \bigcap \mathcal{M}'$ are monochromatic, which means $t_K(x) = 1$. However, the dashed edges represent a petal $M \setminus K$, $M \in \mathcal{M}'$, that contains a bichromatic edge cc', which means $t_M(x) = 0$.

Approximation. Suppose that we have inductively constructed (w,r)-DNFs $F_{\mathcal{M}}$, $F_{\mathcal{M}'}$ for two gates that feed into a gate g that computes a binary operation $\circ \in \{\vee, \wedge\}$. Our goal is to find a (w,r)-DNF $F_{\mathcal{G}}$ that approximates $F_{\mathcal{M}} \circ F_{\mathcal{M}'}$ with tiny error:

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}_0}[F_{\mathcal{G}}(\boldsymbol{x}) > (F_{\mathcal{M}} \circ F_{\mathcal{M}'})(\boldsymbol{x})] \;\le\; 2^{-\omega(w)}, \tag{5}$$

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}_1}[F_{\mathcal{G}}(\boldsymbol{x}) < (F_{\mathcal{M}} \circ F_{\mathcal{M}'})(\boldsymbol{x})] \;\le\; 2^{-\omega(w)}. \tag{6}$$

If $\circ = \vee$, we set $\mathcal{G} \coloneqq \mathrm{pluck}(\mathcal{M} \cup \mathcal{M}')$. To analyse this, we note that plucking incurs n^{-w} errors on $\mathcal{D}_0$ by Claim 3, which verifies (5). On the other hand, plucking introduces no errors on $\mathcal{D}_1$, verifying (6). If $\circ = \wedge$, we first approximate

$$\mathcal{G}' \coloneqq \mathrm{pluck}\left(\{M \cup M' : M \in \mathcal{M},\, M' \in \mathcal{M}',\, M \cup M' \text{ is a matching}\}\right),$$

and then delete all matchings of size larger than w from $\mathcal{G}'$; call the resulting family $\mathcal{G}$. To analyse this, we note that plucking incurs n^{-w} errors on $\mathcal{D}_0$ and no errors on $\mathcal{D}_1$ (we only omitted terms that were not matchings). Moreover, deleting wide terms incurs no error on $\mathcal{D}_0$, and the errors on $\mathcal{D}_1$ can be bounded as follows:

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}_1}[F_{\mathcal{G}}(\boldsymbol{x}) < F_{\mathcal{G}'}(\boldsymbol{x})] \le \sum_{\ell=w}^{2w} \sum_{M \in \mathcal{G}' \cap \mathcal{P}_\ell} \Pr_{\boldsymbol{x}\sim\mathcal{D}_1}[t_M(\boldsymbol{x}) = 1] \overset{\text{(Claim 2)}}{\le} \sum_{\ell=w}^{2w} r^\ell \, (e/n)^\ell \le \sum_{\ell=w}^{\infty} o(1)^\ell \le 2^{-\omega(w)}.$$

This verifies (5)–(6). We now conclude the proof by observing that the DNF F for the output gate has tiny overall error, by summing up all the individual contributions in Equations (5)–(6):

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}}[F(\boldsymbol{x}) \neq \textsc{Match}(\boldsymbol{x})] = \tfrac{1}{2} \Pr_{\boldsymbol{x}\sim\mathcal{D}_1}[F(\boldsymbol{x}) < C(\boldsymbol{x})] + \tfrac{1}{2} \Pr_{\boldsymbol{x}\sim\mathcal{D}_0}[F(\boldsymbol{x}) > C(\boldsymbol{x})] \le \mathrm{size}(C) \cdot 2^{-\omega(w)} \le o(1).$$

3.3 Proofs of Theorems 2–3

To prove Theorem 2 we note that the above proof only ever assumed that the circuit C computes $\textsc{Match}$ correctly on the support of $\mathcal{D}$. But $\textsc{Match}(x) = \textsc{Odd}(x)$ for all $x \in \mathrm{supp}(\mathcal{D})$, and hence the lower bound also applies to $\textsc{Odd}$.

To prove Theorem 3 we apply a standard padding argument and a folklore depth-reduction result. Indeed, it is known that any function in L can be computed by an AC^0-circuit of size 2^{n^ε}, where we can take $\varepsilon > 0$ as any fixed constant [AHM+08, Lemma 8.1]. Let $\varepsilon \coloneqq 1/(4(k+1))$ and define the padded function $f_k \colon \{0,1\}^N \to \{0,1\}$ by $f_k(x,y) \coloneqq \textsc{Odd}_n(x)$, where $N \coloneqq 2^{n^\varepsilon}$ and $\textsc{Odd}_n$ is odd factor on n-vertex graphs. It follows that f_k can be computed by an AC^0 circuit of size $2^{n^\varepsilon} = N$, and its monotone complexity is at least $\exp(n^{1/3-o(1)}) \ge N^{\Omega(\log^k N)}$.

4 Discussion

Let us make some final comments about our proof. First, quantitatively improved lower bounds for bipartite matching follow immediately from improved bounds on matching sunflowers. Indeed, our proof in Section 3 shows more generally that monotone circuits of size $O(n/r_w)^w$ can be approximated by $O(r_w)$-small DNFs where $r_w \coloneqq \max_{\ell \in [2w]} r(\ell, n^{-3w})$. This implies the following closed-form expression for the lower bound on the monotone complexity of bipartite matching (also derivable from the "abstract sunflowers" of [Cav20]):

$$\max_{w\in[n]} \Omega(n/r_w)^w.$$

For example, plugging in Razborov's [Raz85a] bound $r(\ell,\varepsilon) \leq (2^\ell \ell \log(1/\varepsilon))^{2\ell}$ would recover his $n^{\Omega(\log n)}$ lower bound. For another example, instead of using the optimised bounds for robust sunflowers [ALWZ21, Rao20] in our Lemma 2, we could plug in an earlier bound of Rossman [Ros14]. Using Rossman's bound would already yield an exponential lower bound for bipartite matching, albeit with a constant smaller than $1/3$ in the exponent.

We also note that our proof extends to other distributions (and functions) than the odd cut distribution $\mathcal{D}_0$ in Definition 1. For example, we could generate a bipartite graph out of a random vertex colouring $\boldsymbol{c}$ whose number of 1s on opposite sides of the graph differ by 1. As remarked in [FV98], these are rejecting inputs for the $\mathbb{Z}_q$-*satisfiability* problem (satifiability of systems of linear equations modulo q). This recovers the exponential monotone circuit lower bound for $\mathbb{Z}_q$-satisfiability first proved in [GGKS20, GKRS19].

Finally, we can ensure in Theorem 3 that the AC^0 functions are graph properties (functions that output the same value on isomorphic graphs) by applying a more sophisticated padding argument from [CO23, Lemma 3.6]. This contrasts with results of [Ros08, Ros17], stating that *homomorphism-preserved* graph properties in AC^0 can also be computed by small monotone DNFs.

Acknowledgements

We thank Susanna de Rezende, Shuo Pang, Jonas Conneryd, and Noel Arteche for discussions. B.C. acknowledges support of Royal Society University Research Fellowship URF\R1\211106. M.G., A.R., A.S., and D.S. are supported by the Swiss State Secretariat for Education, Research, and Innovation (SERI) under contract number MB22.00026.

References

[AB87] Noga Alon and Ravi Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987. doi:10.1007/BF02579196. 2, 4

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. doi:10.1017/cbo9780511804090. 2

[AG87] Miklós Ajtai and Yuri Gurevich. Monotone versus positive. *Journal of the ACM*, 34(4):1004–1015, 1987. doi:10.1145/31846.31852. 3

[AHM+08] Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael Saks. Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. *SIAM Journal on Computing*, 38(1):63–84, 2008. doi:10.1137/060664537. 8

[ALWZ21] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. Improved bounds for the sunflower lemma. *Annals of Mathematics*, 194(3), 2021. doi:10.4007/annals.2021.194.3.5. 3, 9

[And87] Alexander Andreev. A method for obtaining efficient lower bounds for monotone complexity. *Algebra and Logic*, 26(1):1–18, 1987. doi:10.1007/bf01978380. 4

[ARZ99] Eric Allender, Klaus Reinhardt, and Shiyu Zhou. Isolation, matching, and counting uniform and nonuniform upper bounds. *Journal of Computer and Systems Sciences*, 59(2):164–181, 1999. doi:10.1006/JCSS.1999.1646. 1

[BGW99] László Babai, Anna Gál, and Avi Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999. doi:10.1007/s004930050058. 2, 3

[BM25] Jarosław Błasiok and Linus Meierhöfer. Hardness of clique approximation for monotone circuits. In *Proceedings of the 40th Computational Complexity Conference (CCC)*, LIPIcs. Schloss Dagstuhl, 2025. doi:10.48550/ARXIV.2501.09545. 2, 4

[Cav20] Bruno Cavalar. Sunflower theorems in computational complexity. Master's dissertation, Instituto de Matemática e Estatística, University of São Paulo, 2020. doi:10.11606/D.45.2020.tde-25112020-162107. 9

[CDM21] Arkadev Chattopadhyay, Rajit Datta, and Partha Mukhopadhyay. Lower bounds for monotone arithmetic circuits via communication complexity. In *Proceedings of 53rd Symposium on Theory of Computing (STOC)*, pages 786–799. ACM, 2021. doi:10.1145/3406325.3451069. 4

[CHO$^+$22] Lijie Chen, Shuichi Hirahara, Igor Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. Beyond natural proofs: Hardness magnification and locality. *Journal of the ACM*, 69(4):1–49, 2022. doi:10.1145/3538391. 2

[CKR20] Bruno Cavalar, Mrinal Kumar, and Benjamin Rossman. Monotone circuit lower bounds from robust sunflowers. In *Proceedings of the 14th Latin American Symposium on Theoretical Informatics (LATIN)*, volume 12118 of *Lecture Notes in Computer Science*, pages 311–322. Springer, 2020. doi:10.1007/978-3-030-61792-9_25. 2, 4

[CO23] Bruno Cavalar and Igor Oliveira. Constant-depth circuits vs. monotone circuits. In *Proceedings of the 38h Computational Complexity Conference (CCC)*, volume 264 of *LIPIcs*, pages 29:1–29:37. Schloss Dagstuhl, 2023. doi:10.4230/LIPIcs.CCC.2023.29. 2, 3, 9

[COS17] Xi Chen, Igor Oliveira, and Rocco Servedio. Addition is exponentially harder than counting for shallow monotone circuits. In *Proceedings of 49th Symposium on Theory of Computing (STOC)*, pages 1232–1245. ACM, 2017. doi:10.1145/3055399.3055425. 3

[Dam90] Carsten Damm. Problems complete for $\oplus$L. *Information Processing Letters*, 36(5):247–250, 1990. doi:10.1016/0020-0190(90)90150-v. 1

[dRV25] Susanna de Rezende and Marc Vinyals. Lifting with colourful sunflowers. In *Proceedings of the 40th Computational Complexity Conference (CCC)*, LIPIcs. Schloss Dagstuhl, 2025. 4

[FV98] Tomás Feder and Moshe Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766. 9

[GGKS20] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory of Computing*, 16(1):1–30, 2020. doi:10.4086/toc.2020.v016a013. 1, 2, 9

[GKRS19] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *Proceedings of 10th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 124 of *LIPIcs*, pages 38:1–38:19. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.ITCS.2019.38. 1, 2, 9

[GS92] Michelangelo Grigni and Michael Sipser. Monotone complexity. In *Boolean Function Complexity*, London Mathematical Society Lecture Note Series, pages 57–75. Cambridge University Press, 1992. doi:10.1017/CBO9780511526633.006. 2

[HR00] Danny Harnik and Ran Raz. Higher lower bounds on monotone size. In *Proceedings of 32nd Symposium on Theory of Computing (STOC)*, pages 378–387. ACM, 2000. doi:10.1145/335305.335349. 4

[Juk99] Stasys Jukna. Combinatorics of monotone computations. *Combinatorica*, 19(1):65–85, 1999. doi:10.1007/s004930050046. 4

[Juk12] Stasys Jukna. *Boolean function complexity*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012. Advances and frontiers. doi:10.1007/978-3-642-24508-4. 1, 2

[LMM$^+$22] Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with sunflowers. In *Proceedings of 13th Innovations in Theoretical Computer Science Conference (ITCS)*. Schloss Dagstuhl, 2022. doi:10.4230/LIPICS.ITCS.2022.104. 4

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *Proceedings of Fundamentals of Computation Theory (FCT)*, pages 565–574. Akademie-Verlag, Berlin, 1979. 1

[Mul87] Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7(1):101–104, 1987. doi:10.1007/BF02579205. 1

[Oko82] Elizaveta Okol'nishnikova. The effect of negations on the complexity of realization of monotone Boolean functions by formulas of bounded depth. *Metody Diskretnogo Analiza*, (38):74–80, 1982. 3

[PR17] Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In *Proceedings of 49th Symposium on Theory of Computing (STOC)*, pages 1246–1255. ACM, 2017. doi:10.1145/3055399.3055478. 4

[Rao20] Anup Rao. Coding for sunflowers. *Discrete Analysis*, 2020(2), 2020. doi:10.19086/da.11887. 3, 9

[Raz85a] Alexander Razborov. Lower bounds of monotone complexity of the logical permanent function. *Matematicheskie Zametki*, 37(6):887–900, 942, 1985. doi:10.1007/BF01157687. 1, 2, 3, 4, 9

[Raz85b] Alexander Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Doklady Akademii Nauk SSSR*, 281(4):798–801, 1985. 2

[Rei08] Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4):1–24, 2008. doi:10.1145/1391289.1391291. 2

[Ros08] Benjamin Rossman. Homomorphism preservation theorems. *Journal of the ACM*, 55(3):1–53, 2008. doi:10.1145/1379759.1379763. 9

[Ros14] Benjamin Rossman. The monotone complexity of k-clique on random graphs. *SIAM Journal on Computing*, 43(1):256–279, 2014. doi:10.1137/110839059. 2, 4, 9

[Ros17] Benjamin Rossman. An improved homomorphism preservation theorem from lower bounds in circuit complexity. In *Proceedings of 8th Innovations in Theoretical Computer Science Conference (ITCS)*, LIPIcs, pages 27:1–27:17. Schloss Dagstuhl, 2017. doi:10.4230/LIPIcs.ITCS.2017.27. 9

[RW92] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM*, 39(3):736–744, 1992. doi:10.1145/146637.146684. 3

[Tar88] Éva Tardos. The gap between monotone and nonmonotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988. doi:10.1007/BF02122563. 1, 2

[Wat25] Thomas Watson. Complexity in computer science. Book draft, 2025. 2

[Weg87] Ingo Wegener. *The complexity of Boolean functions*. Wiley-Teubner, 1987. URL: https://eccc.weizmann.ac.il/static/books/The_Complexity_of_Boolean_Functions/. 1, 2

[Wig19] Avi Wigderson. *Mathematics and Computation: A Theory Revolutionizing Technology and Science*. Princeton University Press, 2019. doi:10.2307/j.ctvckq7xb. 1

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il