Electronic Colloquium on Computational Complexity, Report No. 111 (2023)

MidBit^+, Torus Polynomials and Non-classical Polynomials: Equivalences for ACC Lower Bounds

Vaibhav Krishan*

July 29, 2023

Abstract

We give a conversion from non-classical polynomials to MidBit^+ circuits and vice-versa. This conversion, along with previously known results, shows that torus polynomials, non-classical polynomials and MidBit^+ circuits can all be converted to each other. Therefore lower bounds against any of these models lead to lower bounds against all three of them. Each of these three models capture the power of ACC circuits, which are circuits composed of $\mathsf{AND}, \mathsf{OR}, \mathsf{MOD}_\mathsf{m}$ gates for some constant natural number m. Hence lower bounds against any of these models lead to comparable lower bounds against ACC.

1 Introduction

Proving that certain Boolean functions cannot be computed by Boolean circuits of small size, also called proving lower bounds, has been a major quest in complexity theory. A lot of recent work, for example [3, 11, 12, 14, 15, 20, 22, 26, 27] and references therein, has focused on proving lower bounds for constant-depth circuits.

Lower bounds against constant-depth circuits consisting of AND, OR, NOT gates were first proved by Furst, Saxe and Sipser [16] and independently by Ajtai [1], and subsequently improved upon by Yao [29] and by Håstad [18]. Later, Razborov [23] and Smolensky [24] proved lower bounds against constant-depth circuits that additionally contain MOD_p gates[1] where p is constant and prime. A few years later, Barrington [4] posed the question of proving lower bounds against ACC circuits, that are constant-depth circuits consisting of $\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}, \mathsf{MOD}_m$ gates for any constant natural number m.

The class of Boolean functions computable by polynomial size ACC circuits is called ACC (ACC will refer to the class unless explicitly stated). It has been conjectured since the 90's, by Yao [28], that the class of Boolean functions

*Department of Computer Science and Engineering, IIT Bombay, Powai, Mumbai

[1] a MOD_p gate outputs "true" if and only if the number of "true" inputs is divisible by p.

ISSN 1433-8092

computable by polynomial-size constant-depth threshold circuits[2], called TC^0, strictly contains ACC. In fact, it is believed that the majority function, which outputs "true" if and only if more than half of its inputs are "true", is not contained in ACC. A few approaches have been proposed to resolve this conjecture but none of them have been successful. Meanwhile, lower bounds against ACC from uniform classes have been proven.

Williams [27], in a breakthrough result, proved that non-deterministic exponential time is not contained in ACC. The lower bound has subsequently been improved upon by several works, such as Chen, Oliveira and Santhanam [14], Murray and Williams [22], Chen [12], and Chen [13].

All these lower bounds use a faster-than-brute-force satisfiability algorithm for ACC, proved by [27]. The algorithm takes as input an ACC circuit of polynomial size and determines whether there exists an input for which the circuit outputs "true". The algorithm runs asymptotically faster than brute-force i.e. evaluating the circuit on all possible values for the inputs. As the first step, the algorithm uses a simulation of ACC circuits by another class of depth-2 circuits. Several such simulation results are known for ACC, which can play a role in obtaining new lower bounds for ACC. We discuss these simulation results below.

1.1 Simulation Results for ACC

The faster-than-brute-force satisfiability algorithm for ACC uses a result, first proved by Beigel and Tarui [5] and subsequently improved upon by Allender and Gore [2], that ACC is contained in SYM^+. SYM^+ is the class of functions computable by depth-2 circuits of size $2^{(\log n)^{O(1)}}$ where the top gate is a symmetric function[3] and the bottom layer has AND gates of fan-in $(\log n)^{O(1)}$. [5] conjectured that TC^0 is not contained in SYM^+. This will automatically imply that TC^0 is not contained in ACC. Green, Köbler and Torán [17] modified this simulation to obtain a seemingly tighter simulation for ACC, hence leading to a probably easier approach towards proving ACC lower bounds.

Other simulation results are also known for ACC. Bhrushundi, Hosseini, Lovett and Rao [10] proved that torus polynomials and non-classical polynomials, defined in Definitions 1.6 and 1.8 respectively, can approximate functions in ACC efficiently, with appropriately defined notions of efficiency. We discuss all three simulation results below.

1.1.1 MidBit^+

MidBit^+ was studied by [17] where they proved that ACC is contained in MidBit^+. The formal definition of a MidBit^+ circuit follows the definition of the MidBit function. Note that we use 0 for "false" and 1 for "true" throughout.

[2] A threshold circuit consists of threshold gates. A threshold gate outputs "true" if and only if a linear combination of its inputs is more than a threshold.

[3] A function is symmetric if it is invariant under every permutation of its variables on all its inputs.

Definition 1.1 (MidBit)**.** *Define the function* $bin : \mathbb{N} \times \mathbb{N} \to \{0,1\}$ *as* $bin(x,i) = b_i$*, where* b_i *is the* i^{th} *bit in the binary expansion of* x *and* b_0 *is the least significant bit. The function* $\mathsf{MidBit} : \{0,1\}^n \to \{0,1\}$ *is defined as* $\mathsf{MidBit}(x_1,\ldots,x_n) = bin(\sum_{i=1}^{n} x_i, \lfloor \ell/2 \rfloor)$ *where* $\ell = \lfloor \log_2(n) \rfloor + 1$*.*

Definition 1.2 (MidBit^+)**.** *A Boolean circuit is called a* MidBit^+ *circuit if it has the following form:*

1. *The circuit has depth* 2*.*
2. *Inputs are fed into* AND *gates, which comprise the first layer of the circuit.*
3. *The output gate is the* MidBit *function.*

A function $f : \{0,1\}^n \to \{0,1\}$ *belongs to the class* MidBit^+ *if it can be computed by a* MidBit^+ *circuit with fan-in of* AND *gates bounded by* $\log^{O(1)}(n)$ *and fan-in of the output gate bounded by* $2^{\log^{O(1)}(n)}$*.* MidBit^+ *will denote the class unless explicitly stated.*

[17] proved the following.

Claim 1.3 ([17])**.** *Let* $f : \{0,1\}^n \to \{0,1\}$ *belong* ACC*. Then* f *also belongs to* MidBit^+*. In other words,* $\mathsf{ACC} \subseteq \mathsf{MidBit}^+$*.*

Compared to SYM^+ circuits, MidBit^+ circuits are simpler as the top gate is a fixed function rather than any symmetric function. Proving that a function does not belong to MidBit^+ implies that it does not belong to ACC either. Hence proving lower bounds for MidBit^+ is an approach towards proving ACC lower bounds.

Without loss of generality we assume that, each variable is fed into each AND gate at most once, and an AND gate has variables feeding into it if and only if it neither 0 nor 1 is feeding into it. We define some additional terms for a MidBit^+ circuit.

Definition 1.4. *Let* C *be a* MidBit^+ *circuit computing a function* $f : \{0,1\}^n \to \{0,1\}$*. Let the set of* AND *gates be* $A_1,\ldots,A_s$*. Let the number of times* A_i *appears in the circuit be* c_i*. We define the following terms for* C*.*

- *The multiplicity of* C *is* $gcd(c_1,\ldots,c_s)$*. The circuits we consider and construct will always have multiplicity of the form* $2^{2\ell}$ *for some non-negative integer* ℓ*.*
- *Let the circuit have multiplicity* $m = 2^{2\ell}$ *for all subsequent definitions. Let* $B(x)$ *denote the number of* AND *gates that turn "true" on a given input* x*. If* $B(x) = f(x)2^{k+2\ell} + E(x)2^{k+1+2\ell} \mod 2^{k+1+2\ell}$*, with* $E(x) \geq 0$*, for all* $x \in \{0,1\}^n$*, then the non-classical depth of the circuit is defined to be* k *and the error of the circuit is defined to be* $\max_{x\in\{0,1\}^n} E(x)$*.*

- *Let the gate A_i use the set of variables $S_i \subseteq [n]$. Let j denote the highest power of 2 that divides $c_i/2^{2\ell}$. Then the non-classical degree of A_i, denoted by $deg(A_i)$, is defined to be $|S_i| + k - j$. The non-classical degree of C is defined to be $\max_{i:S_i \neq \emptyset}(deg(A_i))$ (note that only non-constant gates are being considered).*

  Note that the degree of the monomial obtained by multiplying the inputs of A_i is $|S_i|$. The additional term $k - j$ is introduced so that when a non-classical monomial (defined in Definition 1.8) is constructed from A_i, it has the same degree as the non-classical degree of A_i.

Tracking these additional parameters will allow us to state our result as an equivalence.

1.1.2 Torus Polynomials

Another approach for ACC lower bounds was developed by [10] where they defined torus polynomials and showed that ACC can be well approximated using torus polynomials of low degree. We define the torus and torus polynomials below.

Definition 1.5 (Torus)**.** *Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ denote the torus. For a number $r \in \mathbb{T}$, let $r \mod 1$ denote its representative in the interval $[-1/2, 1/2)$.*

Definition 1.6 (Torus Polynomial)**.** *A real polynomial $P : \{0,1\}^n \to \mathbb{R}$ is a* torus polynomial *approximating a Boolean function $f : \{0,1\}^n \to \{0,1\}$ within error ε if*

$$\forall x \in \{0,1\}^n, \left| P(x) - \frac{f(x)}{2} \mod 1 \right| \leq \varepsilon$$

The minimum degree d required to approximate a function f by a torus polynomial within error ε will be denoted by $deg_\varepsilon(f)$.

The following can be obtained using Corollary 2.11 of [10].

Claim 1.7 (Corollary 2.1 of [10])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function belonging to ACC. Then $deg_{\frac{1}{n^{O(1)}}}(f) \leq \log(n)^{O(1)}$.*

In addition to the approximation result for ACC, [10] also proved that the majority function requires $\widetilde{\Omega}(\sqrt{n})$ degree to approximate within $\frac{1}{20n}$ error using torus polynomials that are symmetric as real polynomials[4]. They also conjectured that a similar lower bound holds even without the torus polynomial being symmetric which, if true, will imply majority is not contained in ACC.

The upper bound result mentioned above leads to another approach for ACC lower bounds. Let there be a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and $\varepsilon = \frac{1}{n^{O(1)}}$ such that $deg_{\frac{1}{n^{O(1)}}}(f) = \log(n)^{\omega(1)}$. Then clearly $f \notin \mathsf{ACC}$. In particular, finding such a $f \in \mathsf{TC}^0$ will resolve the long-standing conjecture $\mathsf{ACC} \subsetneq \mathsf{TC}^0$.

[4] A polynomial is symmetric if it is invariant under every permutation of variables.

1.1.3 Non-classical Polynomials

Non-classsical polynomials were introduced by Tao and Ziegler [25] in the context of higher-order Fourier analysis (see the survey by Hatami, Hatami and Lovett [19] for applications of higher-order Fourier analysis in theoretical computer science). Non-classical polynomials were studied by Bhowmick and Lovett [7], and Bhrushundi, Harsha and Srinivasan [9] in the context of approximating Boolean functions, see also [8]. Following is a definition of non-classical polynomials over $\mathbb{F}_2^n$. The definition we give is commonly known as the global definition of non-classical polynomials in the literature.

Definition 1.8 (Non-classical Polynomial)**.** *A function $P : \{0,1\}^n \to \mathbb{T}$ is a non-classical polynomial (over $\mathbb{F}_2$) of degree at most d and depth at most k if it can be written as*

$$P(x) = \alpha + \sum_{\substack{\emptyset \subset S \subseteq [n]; 0 \leq j \leq k \\ |S|+j \leq d}} \frac{c_{S,j}}{2^{j+1}} \prod_{i \in S} x_i \mod 1$$

where $c_{S,j} \in \{0,1\}$ and $\alpha \in [0,1)$.

Each term, except for the constant term, appearing in the expression is called a non-classical monomial.

A non-classical polynomial P approximates a Boolean funcion $f : \{0,1\}^n \to \{0,1\}$ within error ε if for all $x \in \{0,1\}^n$, $\left| P(x) - \frac{f(x)}{2} \mod 1 \right| \leq \varepsilon$. The minimum degree d required to approximate a Boolean function f by a non-classical polynomial within error ε will be denoted by $\widetilde{deg}_\varepsilon(f)$.

[10] proved that torus polynomials can be converted to non-classical polynomials and vice-versa. The following result is essentially proved in Claim 1.8 of [10].

Claim 1.9 (Claim 1.8 of [10])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Then $deg_\varepsilon(f) \leq d$ implies $\widetilde{deg}_{O(\varepsilon)}(f) \leq O(d \log(n) + \log(1/\varepsilon))$.*

Hence proving lower bounds for non-classical polynomials is equivalent to proving lower bounds for torus polynomials, providing two equivalent approaches for proving ACC lower bounds.

1.2 Our Contribution

[10], as mentioned earlier, showed how to convert torus polynomials to non-classical polynomials and vice-versa. This conversion can increase the error bound when used to convert torus polynomials to non-classical polynomials. They also showed how to convert MidBit^+ circuits to torus polynomials. Krishan [21] recently showed a partial converse of this result by showing how to convert torus polynomials to MidBit^+ circuits. Note that their conversion can also increase the error i.e. if a torus polynomial is first converted to its corresponding MidBit^+ circuit and then back to a torus polynomial, the error of this new torus polynomial can be higher than the original.

We consider a different approach by proving conversions between non-classical polynomials and MidBit^+ circuits. We show how to convert a non-classical polynomial with certain parameters to a MidBit^+ circuit with same values for corresponding parameters. This conversion can be reversed to obtain a non-classical polynomial with the same parameters and error bound as the original non-classical polynomial.

The formal statement of our conversion result is as follows.

Theorem 1.1 (Main Result)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function and let $\varepsilon < 1/20$. Then the following statements are equivalent:*

1. *There exists a non-classical polynomial P of degree at most d and depth at most k that approximates f within error ε.*
2. *There exists an $\varepsilon' \in [0, \varepsilon)$ and a MidBit^+ circuit with non-classical degree at most d, non-classical depth at most k and multiplicity $2^{2\ell}$ for some ℓ, and the following. Let the number of AND gates that turn "true" on $x \in \{0,1\}^n$ be $B(x)$. Then*

$$B(x) = f(x)2^{k+2\ell} + E(x)2^{k+1+2\ell} \mod 2^{k+1+2\ell}$$

where $2\varepsilon' \leq E(x) \leq 2(\varepsilon' + \varepsilon)$.

Lower bounds for either torus polynomials or non-classical polynomials lead are already known to yield qualitatively equivalent ACC lower bounds. Our conversion result shows that lower bounds for non-classical polynomials and MidBit^+ lead to quantitavely equivalent ACC lower bounds, provided that the lower bound keeps track of all the parameters (including the ones that we have defined for MidBit^+ circuits). Therefore lower bounds for each of these three models lead to comparable ACC lower bounds.

Paper Organization We present our conversion result and some consequences in Section 2. We close with some open questions in Section 3.

2 Equivalence of Non-classical Polynomials and MidBit^+ circuits

We prove our main result first.

Theorem 1.1 (Main Result)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function and let $\varepsilon < 1/20$. Then the following statements are equivalent:*

1. *There exists a non-classical polynomial P of degree at most d and depth at most k that approximates f within error ε.*

2. *There exists an* $\varepsilon' \in [0, \varepsilon)$ *and a* MidBit^+ *circuit with non-classical degree at most* d*, non-classical depth at most* k *and multiplicity* $2^{2\ell}$ *for some* ℓ*, and the following. Let the number of* AND *gates that turn "true" on* $x \in \{0,1\}^n$ *be* $B(x)$*. Then*

$$B(x) = f(x)2^{k+2\ell} + E(x)2^{k+1+2\ell} \mod 2^{k+1+2\ell}$$

where $2\varepsilon' \leq E(x) \leq 2(\varepsilon' + \varepsilon)$*.*

Proof. 1 $\implies$ 2 Let the non-classical polynomial be

$$P(x) = \alpha + \sum_{\substack{\emptyset \neq S \subseteq [n], j \\ 0 \leq j \leq k \\ |S|+j \leq d}} \frac{c_{S,j}}{2^{j+1}} \prod_{i \in S} x_i$$

that has degree at most d, depth at most k, and approximates $f : \{0,1\}^n \to \{0,1\}$ within error ε.

Note that $P(0) = \alpha$. Hence $|\alpha - f(0)/2 \mod 1| \leq \varepsilon$. Let $\alpha = f(0)/2 + \delta$ without loss of generality where $\delta \in [-\varepsilon, \varepsilon]$. We consider two cases.

P.1 First we consider the case when $P(x) - f(x)/2 + \delta \mod 1 = 0$ for all $x \in \{0,1\}^n$. In this case $P(x) - \delta \mod 1 = f(x)/2$. Define $P_{exact}(x) = P(x) - \delta$. Then $P_{exact}(x) \mod 1 = f(x)/2$ for all $x \in \{0,1\}^n$. Define $P_{int}(x) = 2^{k+1} P_{exact}(x)$. Then $P_{int}(x) = 2^k f(x) \mod 2^{k+1}$ and all its coefficients are integers. We will construct a MidBit^+ circuit from P_{int} as follows.

Let $P_{int}(x) = 2^k f(0) + \sum_{\emptyset \neq S \subseteq [n]} c_S x^S$. For each monomial x^S create an AND gate, the inputs to this gate being the variables appearing in the monomial. Create c_S many copies of this AND gate. Do this for each monomial appearing in P_{int}. Create $2^k f(0)$ many copies of an AND gate which takes 1 as its input. Feed all of these into a MidBit gate. This is the first stage of creating the circuit.

Before describing the next steps, we consider the other case.

P.2 Now let there be some $x \in \{0,1\}^n$ such that $P(x) \neq f(x)/2 + \delta$.

Claim 2.1. $1/2^{k+2} \leq \varepsilon$

Proof. Consider two cases as following

- $|\delta| \geq 1/2^{k+2}$. In this case $\varepsilon \geq |P(0) - f(0)/2| = \delta \geq 1/2^{k+2}$.
- $|\delta| < 1/2^{k+2}$. In this case consider any $x \in \{0,1\}^n$ such that $P(x) \neq f(x)/2 + \delta$. Then $P(x) - f(x)/2 - \delta \mod 1 \neq 0$. The minimum non-zero value for $|P(x) - f(x)/2 - \delta \mod 1|$ is $1/2^{k+1}$. Hence $\varepsilon \geq 1/2^{k+1} - |\delta| \geq 1/2^{k+2}$.

■

As the first step we replace α by $q/2^{k+1}$ such that q is the smallest natural number with $q/2^{k+1} \geq \alpha + \varepsilon$. Note that $q/2^{k+1} - \alpha - \varepsilon < 1/2^{k+1}$. Define $\varepsilon' = (q/2^{k+1} - \alpha - \varepsilon)/2$. Then $\varepsilon' < 1/2^{k+2} \leq \varepsilon$. Consider the polynomial $P_{rational}(x) = q/2^{k+1} + P(x)$. Then $P_{rational}(x) - f(x)/2 \mod 1 \in [2\varepsilon', 2(\varepsilon' + \varepsilon)]$. Moreover all the coefficients of $P_{rational}$ are rational numbers with denominator a power of 2 such that the maximum power of 2 that appears in the denominator is at most $k+1$.

Now consider $P_{int}(x) = 2^{k+1} P_{rational}(x)$. Then $P_{int}(x) = f(x)2^k + E(x)2^{k+1} \mod 2^{k+1}$ where $E(x) \in [2\varepsilon', 2(\varepsilon' + \varepsilon)]$. The polynomial P_{int} is now a polynomial with integer coefficients. We will construct a MidBit^+ circuit from P_{int} as follows.

Let $P_{int}(x) = q + \sum_{\emptyset \neq S \subseteq [n]} c_S x^S$. For each monomial x^S create an AND gate, the inputs to this gate being the variables appearing in the monomial. Create c_S many copies of this AND gate. Do this for each monomial appearing in P_{int}. Create q many copies of an AND gate which takes 1 as its input. Feed all of these into a MidBit gate. This is the first stage of creating the circuit.

After the first stage of creating the circuit in either case, consider the fan-in of the MidBit gate in the circuit constructed so far. There are two cases to consider:

C.1 Consider the case when the fan-in is at most $2^{2k+1} - 1$. Add dummy AND gates that take 0 as their input, if required, to ensure that the fan-in of the MidBit gate becomes exactly $2^{2k+1} - 1$. Then the output of the circuit will be $bin(P_{int}(x), k)$, which is exactly the same as $f(x)$. Hence the constructed MidBit^+ circuit computes f.

C.2 Now consider the case when the fan-in is at least 2^{2k+1}. In this case the MidBit gate outputs $bin(P_{int}(x), k')$ for some $k' > k$ while $bin(P_{int}(x), k) = f(x)$. Consider duplicating each AND gate $2^{2\ell}$ times for $\ell = k' - k$. Then the MidBit gate in this new circuit outputs $bin(P_{int}(x), 2k' - k)$ while $bin(2^{2\ell} P_{int}(x), 2k' - k) = f(x)$. Hence for some $\ell > 0$, the MidBit^+ circuit, constructed after duplication, computes f.

To see that the constructed circuit has non-classical depth at most k, note that the polynomial P_{int}, constructed in either Case P.1 or Case P.2, evaluates to $P_{int}(x) = f(x)2^k + E(x)2^{k+1} \mod 2^{k+1}$ with $E(x) \geq 0$. For the circuit constructed in Case C.1, if $B(x)$ denotes the number of AND gates that turn "true" on a given input $x \in \{0,1\}^n$, then $B(x) = 2^k P_{int}(x) \mod 2^{k+1}$. Hence the depth of the circuit is at most k. For the circuit constructed in Case C.2, $B(x) = 2^{k+2\ell} P_{int}(x) \mod 2^{k+1+2\ell}$, hence even in this case the depth is at most k.

For the non-classical degree of the constructed circuit, note that if P contains a non-classical monomial with its input set being S and denominator 2^{j+1}, then the degree of P will be at least $|S| + j$. Corresponding to this monomial in

P, a monomial will appear in P_{int} such that the power of 2 that can divide its coefficient will be $k-j$. When this monomial in P_{int} gets converted into an AND gate A in the final circuit in Case C.1, the power of 2 that divides the fan-in of A will be $k-j$. Therefore the non-classical degree of A will be $|S|+j$, which implies that the non-classical degree of the constructed circuit will be at least $|S|+j$. Note that the value of $|S|+j$ for any non-classical monomial in P is bounded by d, therefore the non-classical degree of the circuit will also be bounded by d. For the circuit in Case C.2, note that the degree is not affected by multiplication of $2^{2\ell}$, hence the same argument holds.

2 $\implies$ 1 Let there be a MidBit^+ circuit with parameters as per Item 2 computing a function $f:\{0,1\}^n \to \{0,1\}$. Start with an empty polynomial. For each AND gate with a variable feeding into it, create a monomial by multiplying each variable that is feeding into it and add it to the polynomial. Ignore all AND gates with a 0 feeding into them. For each AND gate with a 1 feeding into it, add 1 to the polynomial. Let this polynomial be P_{int}.

The value of ℓ is the highest power of 4 that divides all the coefficients of P_{int}. The value of k can be obtained by as $\lfloor(\lfloor\log_2(M)\rfloor+1)/2\rfloor - 2\ell$ where M is the fan-in of the MidBit gate. Divide P_{int} by $2^{k+1+2\ell}$ to get the polynomial $P_{rational}$. The value of ε' is the minimum value attained by $|P_{rational}(x) - f(x)/2 \mod 1|$ over $x \in \{0,1\}^n$. Subtract $2\varepsilon'+\varepsilon$ from $P_{rational}$ to get the final polynomial. This will be a non-classical polynomial with exactly the desired error bound.

It is now easy to see, by reversing the argument for the degree and depth bound in the first implication, that the degree and depth of the constructed non-classical polynomial are as they are claimed to be. □

2.1 Consequences

Consider the exact majority function, that outputs "true" if and only if exactly half of its inputs are "true"[5]. Denote the exact majority function on n inputs by EMAJ_n. The following result can be obtained as a consequence of Lemma 4.1 of [10].

Lemma 2.2 (Lemma 4.1 of [10])**.** *EMAJ_n has a torus polynomial of degree $\log^{O(1)}(n)$ approximating it within error ε for any constant ε.*

Invoking the previous Lemma for a small enough ε, and combining it with Claim 1.9 and Theorem 1.1, we obtain the following corollary.

Corollary 1. $\mathsf{EMAJ}_n \in \mathsf{MidBit}^+$.

The following result is essentially proved in Theorem 3.5 of [17].

Lemma 2.3 (Theorem 3.5 of [17])**.** $\mathsf{MidBit}^+ \circ \mathsf{ACC} \in \mathsf{MidBit}^+$*, where $\circ$ denotes function composition.*

As a consequence of the previous Theorem and Corollary 1, we observe the following containment.

Claim 2.4. $\mathsf{EMAJ} \circ \mathsf{ACC} \in \mathsf{MidBit}^+$.

[5]Note that it always evaluates to "false" if the number of inputs is odd.

3 Open Questions

We state some open questions which we believe will help further the understanding of the three computation models we have studied in this work.

3.1 Non-explicit Parameters in Theorem 1.1

In the statement of Theorem 1.1, we do not give an explicit value for ℓ and ε' when constructing a MidBit^+ circuit from a non-classical polynomial. The main hurdle to determine the value of ℓ is to understand the fan-in of the output gate of the constructed MidBit^+ circuit, for which we were not able to find an explicit and clean expression. For ε', the main hurdle is to get an expression which works with case P.1 and P.2. We leave it open to find a parameter of non-classical polynomials that can be used to calculate ℓ. We considered $P(x)$ evaluated on $x = 1^n$ as a possibility but were unable to figure out the exact value for ℓ using this parameter along with the others.

3.2 Lossy Conversions

The conversion between non-classical polynomials and MidBit^+ circuits we have proved, in Theorem 1.1, is lossless. That is it proves that the two models with their parameters are equivalent. This is not true for conversions between torus polynomials to non-classical polynomials described by [10]. Neither is this true for the procedure to convert torus polynomials to MidBit^+ circuits described by [21].

Recall the definition of non-classical degree of a gate A_i in a MidBit^+ circuit from Definition 1.4. Its degree can be defined as the number of variables that are fed into A_i. The degree of the circuit can be defined as the maximum degree over all its gates. The procedure to convert torus polynomials to MidBit^+ circuits, described in [21], preserves the degree but increases the error. This procedure, when reversed, results in a torus polynomial with increased error. We conjecture that this increase in error is unavoidable. We believe this is due to the fact that torus polynomials can use real coefficients while fan-in of the gates in a MidBit^+ circuits are discrete. Approximating real coefficients by discrete fan-ins will introduce an error for each coefficient and these errors may not cancel out.

Conjecture 1. *There exists a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and an error function $\varepsilon : \mathbb{N} \to \mathbb{R}$ such that $deg_{\varepsilon(n)}(f) = d(n)$ but the degree of any MidBit^+ circuit that computes f within error $\varepsilon(n)$ must be more than $d(n)$.*

3.3 A Result of Beigel, Tarui and Toda [6]

Beigel, Tarui and Toda [6] proved that the class of functions computable by probabilistic $\mathsf{EMAJ} \circ \mathsf{ACC}$ circuits of polynomial size is contained in SYM^+. We conjecture that it is in fact contained in MidBit^+.

Conjecture 2. *Let f be computable by a probabilistic* $\mathsf{EMAJ} \circ \mathsf{ACC}$ *circuit of polynomial size. Then* $f \in \mathsf{MidBit}^{+}$.

Acknowledgements We would like to thank Nutan Limaye for useful discussions, a crucial suggestion that made it cleaner to state our result as well as feedback on earlier drafts. We would like to thank Srikanth Srinivasan and Shachar Lovett for useful discussions. Finally, we would like to thank anonymous CSR 2021 reviewers for their helpful feedback and comments.

References

[1] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of pure and applied logic*, 24(1):1–48, 1983.

[2] Eric Allender and Vivek Gore. On strong separations from ac0. In *International Symposium on Fundamentals of Computation Theory*, pages 1–15. Springer, 1991.

[3] Swapnam Bajpai, Vaibhav Krishan, Deepanshu Kush, Nutan Limaye, and Srikanth Srinivasan. A# sat algorithm for small constant-depth circuits with ptf gates. *Algorithmica*, 84(4):1132–1162, 2022.

[4] David A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.

[5] Richard Beigel and Jun Tarui. On acc (circuit complexity). In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 783–792. IEEE, 1991.

[6] Richard Beigel, Jun Tarui, and Seinosuke Toda. On probabilistic acc circuits with an exact-threshold output gate. In *International Symposium on Algorithms and Computation*, pages 420–429. Springer, 1992.

[7] Abhishek Bhowmick and Shachar Lovett. Nonclassical polynomials as a barrier to polynomial lower bounds. In *30th Conference on Computational Complexity*, page 72, 2015.

[8] Abhishek Bhrushundi. *Towards understanding the approximation of Boolean functions by nonclassical polynomials*. PhD thesis, Rutgers The State University of New Jersey, School of Graduate Studies, 2020.

[9] Abhishek Bhrushundi, Prahladh Harsha, and Srikanth Srinivasan. On polynomial approximations over z/2ˆ kz. In *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[10] Abhishek Bhrushundi, Kaave Hosseini, Shachar Lovett, and Sankeerth Rao. Torus polynomials: An algebraic approach to acc lower bounds. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[11] Arkadev Chattopadhyay and Nikhil Mande. A short list of equalities induces large sign rank. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 47–58. IEEE, 2018.

[12] Lijie Chen. Non-deterministic quasi-polynomial time is average-case hard for acc circuits. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1281–1304. IEEE, 2019.

[13] Lijie Chen. New lower bounds and derandomization for acc, and a derandomization-centric view on the algorithmic method. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[14] Ruiwen Chen, Igor C Oliveira, and Rahul Santhanam. An average-case lower bound against acc0. In *Latin American Symposium on Theoretical Informatics*, pages 317–330. Springer, 2018.

[15] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-case lower bounds and satisfiability algorithms for small threshold circuits. *arXiv preprint arXiv:1806.06290*, 2018.

[16] Merrick Furst, James B Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, 1984.

[17] Frederic Green, Johannes Köbler, and Jacobo Torán. The power of the middle bit. In *[1992] Proceedings of the Seventh Annual Structure in Complexity Theory Conference*, pages 111–117. IEEE, 1992.

[18] Johan Håstad. *Computational limitations of small-depth circuits*. MIT press, 1987.

[19] Hamed Hatami, Pooya Hatami, Shachar Lovett, et al. Higher-order fourier analysis and applications. *Foundations and Trends® in Theoretical Computer Science*, 13(4):247–448, 2019.

[20] Daniel M Kane and Ryan Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 633–643, 2016.

[21] Vaibhav Krishan. Upper bound for torus polynomials. In *Computer Science–Theory and Applications: 16th International Computer Science Symposium in Russia, CSR 2021, Sochi, Russia, June 28–July 2, 2021, Proceedings*, pages 257–263. Springer, 2021.

[22] Cody Murray and Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for np and nqp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 890–901, 2018.

[23] Alexander A Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

[24] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 77–82, 1987.

[25] Terence Tao and Tamar Ziegler. The inverse conjecture for the gowers norm over finite fields in low characteristic. *Annals of Combinatorics*, 16(1):121–188, 2012.

[26] Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 194–202, 2014.

[27] Ryan Williams. Nonuniform acc circuit lower bounds. *Journal of the ACM (JACM)*, 61(1):1–32, 2014.

[28] AC-C Yao. On acc and threshold circuits. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 619–627. IEEE, 1990.

[29] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 1–10. IEEE, 1985.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il