

On the Unprovability of Circuit Size Bounds in Intuitionistic S_2^1

Lijie Chen*

Jiatu Li†

Igor C. Oliveira‡

April 19, 2024

Abstract

We show that there is a constant k such that Buss’s intuitionistic theory IS_2^1 does not prove that SAT requires co-nondeterministic circuits of size at least n^k . To our knowledge, this is the first unconditional unprovability result in bounded arithmetic in the context of worst-case fixed-polynomial size circuit lower bounds. We complement this result by showing that the upper bound $\mathsf{NP} \subseteq \mathsf{coNSIZE}[n^k]$ is unprovable in IS_2^1 .

Contents

1	Introduction	2
1.1	Context and motivation	2
1.2	Results	2
1.3	Related work	5
2	Preliminaries	5
2.1	Intuitionistic bounded arithmetic	5
2.2	Complexity theory	7
3	Unprovability of Lower Bounds in IS_2^1	7
3.1	Unconditional lower bounds for refuters	7
3.2	Unprovability of lower bounds via refuter lower bounds	12
4	Unprovability of Upper Bounds in IS_2^1	13
4.1	Unconditional uniform lower bound against co-nondeterministic circuits	13
4.2	Unprovability of upper bounds via uniform lower bounds	14
5	Interpretation and Generalization of Our Results	15
A	On the Unprovability of Lower Bounds Against Deterministic Circuits	18

*Miller Institute for Basic Research in Science. University of California at Berkeley. Email: lijiechen@berkeley.edu

†Computer Science & Artificial Intelligence Laboratory. Massachusetts Institute of Technology. Email: jiatuli@mit.edu

‡Department of Computer Science. University of Warwick. Email: igor.oliveira@warwick.ac.uk

1 Introduction

1.1 Context and motivation

Pich and Santhanam [PS21] and subsequently Li and Oliveira [LO23] have shown that certain strong complexity lower bounds are unprovable in bounded arithmetic theory PV_1 and its extensions, such as APC_1 and S_2^1 . For instance, [PS21] established that PV_1 does not prove that there is a language $L \in NP$ that is average-case hard against co-nondeterministic circuits of size $2^{n^{o(1)}}$. Note that, while the unprovability results of [PS21] and [LO23] are unconditional, they only apply to significantly strong complexity lower bounds.

It would be more interesting to understand the (un)provability of circuit lower bounds that are closer to major open problems from complexity theory, such as showing that NP is not contained in $P/poly$. To achieve this, it is necessary to develop techniques to address the following three aspects for which these lower bound statements are stronger than $NP \not\subseteq P/poly$:

- (A) The statements consider average-case instead of worst-case lower bounds.
- (B) They refer to sub-exponential size instead of super-polynomial size lower bounds.
- (C) The lower bound is against co-nondeterministic circuits instead of deterministic circuits.

Aiming for unconditional results, we propose the consideration of these three challenges in the more restricted (but formally necessary) setting of *intuitionistic* theories of bounded arithmetic. Intuitionistic theories distinguish themselves from classical logic systems by aligning more closely with the concept of constructive proofs. Notably, intuitionistic logic systems do not presuppose the principles of the excluded middle and double negation elimination, which give rise to key inference rules in classical logic (see, e.g., [Koh08] for some background and applications).¹ We note that connections between complexity theory and intuitionistic logic have been widely investigated (see, e.g., [Bus86b, CU93, Bus90a, Har92, FM98, Bus90b, Avi00, Mon08, GP13, Jeř17] and references therein).

1.2 Results

We show that *worst-case fixed-polynomial size* lower bounds against *co-nondeterministic* circuits for a language in NP cannot be established in IS_2^1 [Bus86b], the intuitionistic analogue of Buss's theory S_2^1 [Bus86a]. This unconditional result addresses aspects (A) and (B) highlighted above, in the setting of intuitionistic bounded arithmetic.² We also remark that the lower bound we consider, which is weaker than $NP \not\subseteq coNP/poly$, is implied by the widely believed conjecture that PH does not collapse. On the other hand, we are not aware of any standard assumption that implies the strong two-sided error average-case lower bounds for NP against subexponential-size co-nondeterministic circuits considered in previous unprovability results [PS21, LO23].

A natural question is whether one can also unconditionally show that lower bounds against deterministic circuits (instead of co-nondeterministic circuits) are unprovable in IS_2^1 , which corresponds to Item (C) in the discussion above. As we explain in Appendix A, for some formalizations

¹For instance, win-win arguments from complexity theory, such as the non-constructive proof of the existence of a pseudodeterministic algorithm for generating primes from [OS17] by considering if $PSPACE \subseteq BPP$ or not, are not available in intuitionistic theories.

²Note that showing the unprovability of fixed-polynomial size lower bounds is stronger than showing the unprovability of super-polynomial size lower bounds.

of this lower bound statement, unprovability in IS_2^1 yields unprovability in the stronger theory S_2^1 , an observation that first appeared in Ghasemloo and Pich [GP13]. We believe that this connection further motivates the investigation of the unprovability of complexity lower bounds in intuitionistic theories.

Next, we provide more details about our results and formalizations.

Formalizations. Fix a polynomial-time nondeterministic machine M . Given $n_0 \in \mathbb{N}$ and a size-bound $s: \mathbb{N} \rightarrow \mathbb{N}$ that is sufficiently time-constructive and satisfies $n \leq s(n) \leq 2^n$, we consider a sentence $\text{LB}^{\text{exp}}(M, s, n_0)$ stating that:³

$$\forall n \in \text{LogLog} \text{ with } n \geq n_0 \forall D \in \text{coNSIZE}[s(n)] \exists x \in \{0, 1\}^n \text{ Error}_M(D, n, x),$$

where $\text{Error}_M(D, n, x)$ denotes the formula

$$(\exists y, z \in \{0, 1\}^{s(n)} M(x, y) = 1 \wedge D(x, z) = 0) \vee (\forall y, z \in \{0, 1\}^{s(n)} M(x, y) = 0 \wedge D(x, z) = 1).$$

This sentence expresses that, for every $n \geq n_0$, no co-nondeterministic circuit of size at most $s(n)$ decides $L(M)$ on inputs of length n . The notation LB^{exp} emphasizes that a candidate proof of this sentence is able to manipulate concepts of exponential size $2^{O(n)}$, since $n \in \text{LogLog}$. We refer to Section 2.1 for more details on how to formalize this sentence in the setting of IS_2^1 .

If $s(n) = n^k$ for some rational number $k \geq 1$, we can consider a sentence $\text{LB}^{\text{poly}}(M, s, n_0)$ stating that:

$$\forall n \in \text{Log} \text{ with } n \geq n_0 \forall D \in \text{coNSIZE}[s(n)] \exists x \in \{0, 1\}^n \text{ Error}_M(D, n, x).$$

The key difference here is that when $s(n) = n^k$ it is sufficient to assume $n \in \text{Log}$ to obtain a natural formalization of complexity lower bounds. Correspondingly, the notation LB^{poly} emphasizes that a candidate proof of this sentence can manipulate concepts of size polynomial in n .

Theorem 1.1. *The following results hold:*

- (i) [Exponential Regime] *Let $\delta > 0$ be a rational number, $n_0 \in \mathbb{N}$, and M be a polynomial-time nondeterministic machine. Then $\text{IS}_2^1 \not\vdash \text{LB}^{\text{exp}}(M, 2^{n^\delta}, n_0)$.*
- (ii) [Polynomial Regime] *Let $n_0 \in \mathbb{N}$ and M be a polynomial-time nondeterministic machine. Then there is an integer $k \geq 1$ such that $\text{IS}_2^1 \not\vdash \text{LB}^{\text{poly}}(M, n^k, n_0)$.*

The proof of Theorem 1.1 combines two main steps. First, we employ a strong witnessing result [Bus86b, CU93] for the intuitionistic theory IS_2^1 to show that the provability of a circuit lower bound yields a computationally bounded *refuter*. A refuter for a lower bound of the form $L \notin \mathcal{C}$, where \mathcal{C} is a complexity class, is an algorithm $R(1^n, E)$ that, given an input length n and a device E from \mathcal{C} , outputs an n -bit string x such that $E(x) \neq L(x)$. We then establish unconditionally that such a refuter does not exist. Consequently, the lower bound sentence cannot be proved in the theory IS_2^1 .

Elaborating on our approach, we now present some new consequences for refuters that might be of independent interest. We use SAT_n to denote the decision version of the satisfiability problem for De Morgan Boolean formulas represented by n -bit strings (our results are robust to encoding details). As in the discussion above, we say that a refuter R for a language L succeeds against a class \mathcal{C} of devices on input length n if, for every $E \in \mathcal{C}$, $R(1^n, E)$ outputs $x \in \{0, 1\}^n$ such that $E(x) \neq L(x)$. Observe that the existence of such a refuter implies that $L \notin \mathcal{C}$.

³Note that $n \in \text{LogLog}$ essentially means that bounded quantifiers refer to objects of length $2^{O(n)}$; similarly, $n \in \text{Log}$ means that bounded quantifiers refer to objects of length $\text{poly}(n)$. For a formal definition, see Section 2.1.

Theorem 1.2. *The following results hold:*

- (i) *Let k and c be rational numbers such that $1 < k < c < k^2$. Then there is no refuter $R(1^n, E_n)$ for SAT_n against $E_n \in \text{coNSIZE}[n^k]$ that has nonuniform circuit size $\leq n^c$ and succeeds on every large enough input length n .*
- (ii) *There is an integer $k \geq 1$ such that there is no refuter $R(1^n, E_n)$ for SAT_n against $E_n \in \text{coNSIZE}[n^k]$ of nonuniform polynomial size which succeeds on every large enough input length.*

These two items are incomparable. The first item of Theorem 1.2 is an impossibility result for refuting even very weak lower bounds (n^k gates for any fixed $k > 1$), but only addresses refuters of size smaller than n^{k^2} . On the other hand, the second item holds against any polynomial size refuter, but does not provide an explicit constant k for the size bound.

At a high level, the proof of Theorem 1.2 proceeds by contradiction. From the existence of such a refuter, we obtain a worst-case *upper bound* on the complexity of SAT_n . This step requires an extension of a technique from previous papers [Kra11, Pic15] to the setting of *worst-case complexity* and to the *polynomial circuit size* regime. To achieve this, we employ a new *bootstrapping argument* (see Section 3.1) that invokes the refuter over different input lengths and aggregates the information obtained from it. Finally, the circuit size upper bound extracted from the refuter contradicts the original assumption that a refuter exists, since it implicitly assumes a corresponding circuit lower bound. We note that the second item of Theorem 1.2 requires an extra non-constructive ingredient, and as a result, the proof does not produce an explicit constant k .

Finally, we complement Theorem 1.1 by establishing the unprovability in IS_2^1 of the upper bound $\text{NP} \subseteq \text{coNSIZE}[n^k]$, for any fixed $k \in \mathbb{N}$. The formalization of this upper bound statement is presented in Section 4.

Theorem 1.3 (Informal, see Theorem 4.5). *For any constant $k \in \mathbb{N}$, $\text{IS}_2^1 \not\vdash \text{“NP} \subseteq \text{coNSIZE}[n^k]\text{”}$.*

The proof of Theorem 1.3 relies on an adaptation of an approach from [KO17], which reduces the *unprovability of non-uniform circuit upper bounds* to establishing certain *uniform circuit lower bounds* (in the standard sense of complexity theory). In our result, the required uniform circuit lower bound is achieved by a modification of an argument from [SW14].

Theorem 1.3 contributes to an active line of research on the unprovability of circuit upper bounds [CK07, KO17, BM20, BKO20, CKKO21, ABM23]. In contrast to other unconditional results from these papers, which hold even for stronger theories, Theorem 1.3 establishes the unprovability of *co-nondeterministic* circuit upper bounds for NP .⁴

Altogether, Theorem 1.1 Item (ii) and Theorem 1.3 bring us closer to an unconditional *independence* result for IS_2^1 with respect to worst-case fixed-polynomial bounds in co-nondeterministic circuit complexity.⁵

Finally, we note that both Theorem 1.1 and Theorem 1.3 can be extended to “semi-classical” formalizations of the statements, where the sub-formula inside the outermost existential quantifier

⁴Note that establishing the unprovability of co-nondeterministic circuit size upper bounds is stronger than establishing the unprovability of deterministic circuit size upper bounds.

⁵Formally, our results show that for each $L \in \text{NP}$ there is k such that the lower bound $L \notin \text{coNSIZE}[n^k]$ is unprovable, and that for each k there is $L \in \text{NP}$ such that the upper bound $L \in \text{coNSIZE}[n^k]$ is unprovable. While our unprovability results are robust to the use of different machines of the same time complexity to represent L in a sentence, we note that they do not give a fixed pair (L', k') for which both lower bounds and upper bounds are unprovable.

can be replaced by any classically equivalent formula. Details of this extension are discussed in Section 5.

1.3 Related work

Ghasemloo and Pich [GP13] studied connections between natural proofs [RR97] and intuitionistic logic. In a bit more detail, the theory of natural proofs can be used to establish the *conditional* unprovability of circuit lower bounds in classical theories admitting certain interpolation theorems (see [Raz95, Kra97]), and [GP13] investigates what (conditional) consequences this can have for the provability of lower bounds in intuitionistic theories.

Theorem 1.1 should be contrasted with a result from Cook and Urquhart [CU93, Theorem 10.16] establishing the unprovability of super-polynomial lower bounds for the extended Frege propositional proof system in the related intuitionistic theory IPV^ω (see also [KP89, Corollary 4] and [Bus90b, Section 6.2]). We note that the two results consider different lower bound questions, formalizations, and choice of parameters. Moreover, their proofs rely on completely different approaches. For instance, in terms of the formalization, the sentences LB^{exp} and LB^{poly} do not state that the input x under $\text{Error}_M(x)$ is a tautology, which appears to be crucial in the results from [CU93, KP89]. Furthermore, to our knowledge, it is not known how to extend their results to the fixed-polynomial size regime, as in Theorem 1.1 Item (ii).

It is perhaps interesting to compare Theorem 1.2 Item (ii) with the positive results of Gutfreund, Shaltiel, and Ta-Shma [GST07] about the existence of refuters (see also [CJSW21] and references therein). For instance, [GST07, Lemma 4.1] roughly states that, if $NP \neq RP$, then it is possible to produce counter-examples to the correctness of a randomized input machine of complexity n^k in time of order n^{k^2} . While the parameters of our lower bound and of their upper bound nearly match, we note that their results and Theorem 1.2 Item (ii) refer to different complexity lower bounds and consider slightly different refuter guarantees.

Acknowledgements. We would like to thank Erfan Khaniki and Dimitrios Tsintsilidas for discussions on proof complexity lower bounds in intuitionistic bounded arithmetic. Igor C. Oliveira received support from the Royal Society University Research Fellowship URF\R1\191059; the UKRI Frontier Research Guarantee Grant EP/Y007999/1; and the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick. Lijie Chen is supported by a Miller Research Fellowship. Jiayu Li is supported by an Akamai Presidential Fellowship.

2 Preliminaries

2.1 Intuitionistic bounded arithmetic

We consider the intuitionistic theory of bounded arithmetic IS_2^1 introduced in [Bus86b] (see also [CU93, Bus90a] for equivalent definitions). Our unprovability results are quite robust, as they rely on the consequences of a witnessing theorem for IS_2^1 and do not crucially depend on details of the formalization. Nevertheless, for concreteness and in order to complement the discussion in Appendix A, below we provide more details about the theory. The exposition assumes basic familiarity with bounded arithmetic (see, e.g., [Bus97] for the necessary background).

Informally, IS_2^1 can be defined as theory S_2^1 but with intuitionistic predicate logic and polynomial induction restricted to Σ_1^{b+} -formulas, i.e., Σ_1^b -formulas that do not contain implications and negations. Next, we review some details for a reader that might not be familiar with this terminology. Since we will not make use of sequent calculus, we follow the Hilbert-style equivalent presentation from [CU93].

As in the case of S_2^1 , we take the language of IS_2^1 to consist of non-logical symbols $0, S, +, \times, \#, \lfloor \frac{1}{2}x \rfloor, |x|$, and \leq with their usual interpretations over the intended standard model \mathbb{N} . The logical symbols are $\wedge, \vee, \rightarrow, \forall, \exists$, and $=$. The connectives \neg and \leftrightarrow can be introduced with appropriate abbreviations using \rightarrow [CU93, Page 109]. In terms of non-logical axioms, IS_2^1 consists of 21 basic axioms (see [CU93, Page 111]) and the axiom scheme Σ_1^{b+} -PIND (discussed below). The standard logical axiom schemes and rules of inference governing intuitionistic predicate logic are listed in [CU93, Page 110].

For a term t not containing a variable x , we can define *bounded quantifiers* $(\exists x \leq t)\varphi$ and $(\forall x \leq t)\varphi$ via the abbreviations $\exists x (x \leq t \wedge \varphi)$ and $\forall x (St \leq x \vee \varphi)$, respectively. *Sharply bounded quantifiers* are bounded quantifiers of the form $(\exists x \leq |t|)$ and $(\forall x \leq |t|)$. We recall that $\Pi_0^b = \Sigma_0^b$ is the set of formulas whose quantifiers are all sharply bounded. Similarly, formulas containing only bounded quantifiers can be classified into hierarchies Σ_i^b and Π_i^b by counting alternations of bounded quantifiers while ignoring sharply bounded quantifiers (cf. [CU93, Page 111]). A formula is *positive* if it contains no occurrence of the symbol \rightarrow (recall that negations are introduced via abbreviation and do not need to be explicitly discussed). A formula is Σ_1^{b+} if it is both Σ_1^b and positive.

We define $\forall n \in \text{Log}$ (resp. $\exists n \in \text{Log}$) as the abbreviation of $\forall N \forall n = |N|$ (resp. $\exists N \exists n = |N|$), namely bounded quantifiers refer to objects of length up to $\text{poly}(n)$. Similarly, we define $\forall n \in \text{LogLog}$ (resp. $\exists n \in \text{LogLog}$) as the abbreviation of $\forall N \forall n = ||N||$ (resp. $\exists N \exists n = ||N||$).

Finally, IS_2^1 admits the aforementioned Σ_1^{b+} -PIND axiom scheme, consisting of formulas (possibly with additional parameters) of the form $\varphi(0) \wedge \forall x (\varphi(\lfloor \frac{1}{2}x \rfloor) \rightarrow \varphi(x)) \rightarrow \forall y \varphi(y)$, where φ is a Σ_1^{b+} -formula.

Formalization of complexity lower bounds. We now discuss the formalization of the sentences $\text{LB}^{\text{exp}}(M, s, n_0)$ and $\text{LB}^{\text{poly}}(M, s, n_0)$ informally introduced in Section 1.2. Recall that M is a non-deterministic polynomial-time machine, $s: \mathbb{N} \rightarrow \mathbb{N}$, and $n_0 \in \mathbb{N}$. The fixed machine M and the constant n_0 can be explicitly encoded using a term of IS_2^1 built from the constant symbol 0 and from the other function symbols. Note that, in order to formalize the lower bound statements, it is sufficient to be able to define any polynomial-time function in the language of IS_2^1 . Indeed, this allows us to employ appropriate formulas to specify the output of M on a given input, check if an object D encodes a circuit of a given size, evaluate a given circuit on an input pair (x, y) , decode an n -bit string from an object x , etc.

It turns out that, as established in [CU93, Corollary 2.7], every function f computable in polynomial time is Σ_1^{b+} -definable in IS_2^1 . This means that there is a Σ_1^{b+} -formula $\phi(x, y)$ such that $\mathbb{N} \models \forall x \phi(x, f(x))$ and IS_2^1 proves that $\forall x \exists! y \phi(x, y)$ (see [CU93, Pages 114-115]). Given this result, it is not hard to fully specify the sentences $\text{LB}^{\text{exp}}(M, s, n_0)$ and $\text{LB}^{\text{poly}}(M, s, n_0)$ using a formula for each relevant polynomial-time function. Since the details of how this can be done appear on previous works on the provability of lower bounds in bounded arithmetic (see, e.g., [Pic15]), below we comment only on the part of the formalization that affects the running time of computations obtained from proofs in IS_2^1 .

For instance, the sentence $\text{LB}^{\text{exp}}(M, s, n_0)$ can be expressed as

$$\forall N \forall n \forall D \exists x \left(n = ||N|| \wedge n \geq n_0 \wedge |x| = n \wedge \text{Circuit}_s(D, n) \rightarrow \text{Error}_M(D, n, x) \right),$$

where Circuit_s is a Σ_1^{b+} -formula that checks if D is the description of a co-nondeterministic circuit of size at most $s(n)$, and Error_M is built as a formula that checks if $M(x) \neq D(x)$. Note that if x is an n -bit number and D encodes a circuit of size at most 2^n , a polynomial-time function f over inputs N , n , and D computes in time $\text{poly}(|N|, |D|, |n|) = 2^{O(n)}$. On the other hand, when $s(n) \leq n^k$ for some k , we can take $n = |N|$ in the formalization of $\text{LB}^{\text{poly}}(M, s, n_0)$, which yields an algorithm from a proof in IS_2^1 whose running time is $\text{poly}(n)$ instead of $\text{poly}(2^n)$.

We stress that our unprovability results do not depend on the specific formulas employed to formalize the lower bound sentence. For more detailed discussion, see Section 5.

2.2 Complexity theory

We assume basic familiarity with complexity theory, e.g., the definition of complexity classes P , NP and coNP (see [AB09]). We use $\text{NSIZE}[s(n)]$ (resp. $\text{coNSIZE}[s(n)]$) to denote the set of languages decidable by families of nondeterministic (resp. co-nondeterministic) circuits of size $s(n)$.

Uniform circuits. Let \mathcal{C} be a non-uniform complexity class, e.g., $\mathcal{C} = \text{SIZE}[\text{poly}(n)]$ or $\mathcal{C} = \text{NSIZE}[\text{poly}(n)]$. We say that $L \in \text{P-uniform } \mathcal{C}$ if L is decidable by a family of \mathcal{C} -circuits $\{C_n : \{0, 1\}^n \rightarrow \{0, 1\}\}_{n \in \mathbb{N}}$ such that there is a polynomial-time Turing machine M such that $M(1^n)$ outputs the description of C_n .

Refuters. Let L be a language, R be a uniform algorithm, and \mathcal{C} be a complexity class. We often abuse notation and also view \mathcal{C} as a class of computational devices. We say that R is a refuter for $L \in \mathcal{C}$ (or a refuter for L against \mathcal{C}) on input length n if, for every $E \in \mathcal{C}$, $R(1^n, E)$ outputs $x \in \{0, 1\}^n$ such that $E(x) \neq L(x)$. We extend this definition in the natural way when R represents a non-uniform family of circuits. In this case, we might omit the input 1^n and simply write R_n .

3 Unprovability of Lower Bounds in IS_2^1

In this section, we prove new results about refuters and employ them to establish the unprovability of lower bounds against co-nondeterministic circuits in theory IS_2^1 .

3.1 Unconditional lower bounds for refuters

In this section, we prove each item of Theorem 1.2 and establish some related results needed in the proof of Theorem 1.1. First, we prove the following lemma, which shows that *worst-case* upper bounds can be extracted from refuters, even in situations where the refuter runs in exponential time. This result will be useful in the proof of Theorem 1.1 Item (i).

Lemma 3.1. *Let $L \in \text{NP}$, and let $\delta > 0$. Suppose that there is a uniform algorithm $R(1^n, D)$ such that, for every co-nondeterministic circuit D on n input variables and of size at most 2^{n^δ} , $R(1^n, D)$ runs in time $2^{O(n)}$ and outputs a string $w \in \{0, 1\}^n$ such that $D(w) \neq L(w)$. Then, for every language $L' \in \text{NP}$ and for every constant $\varepsilon > 0$, we have $L' \in \text{DTIME}[2^{n^\varepsilon}]$.*

Proof. Suppose that $L \in \text{NTIME}[n^d]$ for some $d \in \mathbb{N}$. Let M' be a nondeterministic machine that decides L' and runs in time at most $n^{c'}$, where $c' \in \mathbb{N}$. Let $\varepsilon > 0$ be an arbitrary constant. Finally, let $\gamma = \gamma(d, \varepsilon) > 0$ be a small enough constant to be defined later. We argue that the following deterministic algorithm $B^\gamma(x)$ decides L' in time $O(2^{n^\varepsilon})$:

1. Let $x \in \{0, 1\}^n$ be the input string.
2. B^γ computes the description of a co-nondeterministic circuit E' of size at most $n^{2c'}$ that decides the complement of L' . In other words, $E'(u) = 1 - L'(u)$ for every string $u \in \{0, 1\}^n$.
3. B^γ produces the code of a co-nondeterministic circuit $D_x(y)$, where $y \in \{0, 1\}^{n^\gamma}$, such that $D_x(y)$ ignores its input y and computes according to $E'(x)$.
(In other words, while the length of the main input string y of $D_x(y)$ is smaller than the length of the main input string u of $E'(u)$, they share the same non-deterministic input string, and E' sets u to be the fixed string x .)
4. B^γ computes $w = R(1^{n^\gamma}, D_x) \in \{0, 1\}^{n^\gamma}$.
5. Finally, B^γ determines the bit $b = L(w)$ by a brute force computation, then sets b as its output bit.

First, we argue that B^γ decides L' . Since D_x is a co-nondeterministic circuit over $m = n^\gamma$ input strings and of size at most $n^{2c'} = m^{2c'/\gamma} \leq 2^{m^\delta}$ (for a large enough m and assuming that γ is constant), $R(1^{n^\gamma}, D_x)$ outputs a string $w \in \{0, 1\}^{n^\gamma}$ such that $L(w) = 1 - D_x(w)$. Consequently,

$$b = L(w) = 1 - D_x(w) = 1 - E'(x) = 1 - (1 - L'(x)) = L'(x),$$

i.e., the output bit of $B(x)$ is correct.

Next, we argue that B runs in time at most $O(2^{n^\varepsilon})$. Clearly, Steps 1–3 all run in $\text{poly}(n)$ time. Moreover, Step 4 runs in time $2^{O(n^\gamma)}$ under the assumption on the running time of $R(1^{n^\gamma}, D_x)$. This is at most 2^{n^ε} if we set $\gamma \leq \varepsilon/2$. Finally, since $L \in \text{NTIME}[n^d]$, the brute force computation in Step 5 can be performed in deterministic time $2^{O(\ell^d)}$ over an input of length ℓ . Since $\ell = n^\gamma = |w|$ in our case, if $\gamma \leq \varepsilon/2d$ we get that Step 5 runs in time at most 2^{n^ε} . Overall, if we set $\gamma \triangleq \varepsilon/2d$, it follows that B^γ runs in time at most $O(2^{n^\varepsilon})$. This completes the proof that $L' \in \text{DTIME}[2^{n^\varepsilon}]$.⁶ \square

In the proof of the next result, we employ a more sophisticated *bootstrapping argument* consisting of iterated applications of the refuter over different input lengths. Recall that we use SAT_n to denote the satisfiability problem for De Morgan Boolean formulas represented by n -bit strings.

Theorem 3.2 (Restatement of Theorem 1.2 Item (i)). *Let k and c be rational numbers such that $1 < k < c < k^2$. Then there is no refuter $R(1^n, E_n)$ for SAT_n against $E_n \in \text{coNSIZE}[n^k]$ that has nonuniform circuit size $\leq n^c$ and succeeds on every large enough input length n .*

Proof. In order to simplify some calculations, we prove the result for a fixed but arbitrary language $L \in \text{NTIME}[n]$. Since $\text{poly}(\log n)$ overheads in our estimates do not affect the result, it is easy to check that the same proof works for a language in $\text{NTIME}[n \cdot \text{poly}(\log n)]$, such as $\text{SAT} = \{\text{SAT}_n\}$. Similarly, we will tacitly assume that machines of complexity t can be simulated by circuit of size t

⁶We observe that, in the proof of Lemma 3.1, it is enough for the refuter R to work on co-nondeterministic circuits of size $n^{\alpha(n)}$, where $\alpha(n) \rightarrow \infty$. However, the formulation above will be sufficient for our purposes.

(instead of $O(t \cdot \log t)$). Our construction and upper bounds can be easily adjusted to account for these small overheads, since for constants $a < b$, $n^a \cdot \text{poly}(\log n) < n^b$ for every large enough n .

Let $1 < k < c < k^2$, and let M be a linear time nondeterministic machine that decides L . Now consider an arbitrary $n_0 \in \mathbb{N}$, and suppose towards a contradiction that $R = \{R_\ell\}_{\ell \geq n_0}$ is a sequence of nonuniform circuits R_ℓ of size $\leq \ell^c$ such that, given a circuit $E_\ell \in \text{coNSIZE}[\ell^k]$ over ℓ input bits, $R_\ell(E_\ell)$ outputs a string x_ℓ such that $M(x_\ell) \neq E_\ell(x_\ell)$. We will prove the following claim.

Claim 3.3. *There is an input length $\ell_0 \geq n_0$ and a deterministic circuit B_{ℓ_0} of size at most ℓ_0^k that agrees with the language L over $\{0, 1\}^{\ell_0}$, i.e., $B_{\ell_0}(x) = M(x)$ for all $x \in \{0, 1\}^{\ell_0}$.*

Note that the existence of B_{ℓ_0} is in contradiction with the existence of the refuter R_{ℓ_0} , since there is no string x_{ℓ_0} such that $B_{\ell_0}(x_{\ell_0}) \neq M(x_{\ell_0})$. Consequently, in order to complete the proof of Theorem 3.2, it is sufficient to establish Claim 3.3.

Proof of Claim 3.3. We will consider a large enough input length $\ell_0 \geq n_0$ specified later in the proof. Consider a deterministic circuit $B_{\ell_0}(x)$ that computes as follows:⁷

1. B_{ℓ_0} is given a string $x_{\ell_0} \in \{0, 1\}^{\ell_0}$ as input.
2. It computes the description of a co-nondeterministic circuit $E_{\ell_1}(z)$ operating on inputs of length ℓ_1 that ignores its input string z and satisfies $E_{\ell_1}(z) \triangleq \overline{M}_{\ell_0}(x_{\ell_0})$.
(The value $\ell_1 < \ell_0$ will be defined below.)
3. B_{ℓ_0} obtains $x_{\ell_1} \triangleq R_{\ell_1}(E_{\ell_1})$.
(Note that if $\text{size}(E_{\ell_1}) \leq \ell_1^k$ then $M_{\ell_1}(x_{\ell_1}) = 1 - E_{\ell_1}(x_{\ell_1}) = 1 - \overline{M}_{\ell_0}(x_{\ell_0}) = M_{\ell_0}(x_{\ell_0})$.)
4. Finally, it computes $b_{\ell_1} \triangleq M_{\ell_1}(x_{\ell_1})$, and uses this bit to decide $M_{\ell_0}(x_{\ell_0})$.

Let $t(\ell_1)$ denote the circuit complexity of computing the bit $b_{\ell_1} \triangleq M_{\ell_1}(x_{\ell_1})$ on an arbitrary input x_{ℓ_1} , i.e., the complexity of deciding if $x_{\ell_1} \in L$ over strings of length ℓ_1 . As explained below, in order for B_{ℓ_0} to be correct and of the desired size, it is sufficient that:

- $\text{size}(E_{\ell_1}) \approx \ell_0 \leq \ell_1^k$ (i.e., the refuter receives a circuit of bounded size in Step 3), where we have used that M runs in nondeterministic linear time and omitted polylog factors.
(In our analysis below, we can assume that the description of E_{ℓ_1} can be computed in size at most $\ell_0^k/4$, since $k > 1$ and M runs in linear time.)
- We need that $\ell_1 \geq n_0$, so the output of $R_{\ell_1}(E_{\ell_1})$ is defined in Step 3.
- The circuit size of the refuter in Step 3, which is upper bounded by ℓ_1^c , satisfies $\ell_1^c \leq \ell_0^k/4$.
- The circuit size $t(\ell_1)$ needed to compute b_{ℓ_1} satisfies $t(\ell_1) \leq \ell_0^k/4$.

If these conditions are met, $B_{\ell_0}(x) = M(x)$ for all $x \in \{0, 1\}^{\ell_0}$, and its overall size $t(\ell_0)$ is strictly less than ℓ_0^k (with some room to spare that will be handy later in the proof), since

$$t(\ell_0) \leq \ell_0^k/4 + \ell_0^k/4 + t(\ell_1) \leq \ell_0^k/4 + \ell_0^k/4 + \ell_0^k/4 < \ell_0^k.$$

⁷We often write M_u to emphasize that we run the machine M on inputs of length u .

Constraints. The conditions described above yield the following inequalities (omitting some small order factors):

$$\ell_0^{1/k} \leq \ell_1 \leq \ell_0^{k/c} \quad (\text{thus } c < k^2) \quad \text{and} \quad t(\ell_1) \leq \ell_0^k/4.$$

Recall that the condition $c < k^2$ is one of our assumptions. Jumping ahead, we will define ℓ_1 as a function of ℓ_0 , k , and c , and employ a recursive approach to compute b_{ℓ_1} so that the conditions are satisfied.

Key insight. Note that in order to compute $b_{\ell_1} = M_{\ell_1}(x_{\ell_1})$ we must solve the *same problem* on a *smaller input length*, i.e., we would like to design a circuit B_{ℓ_1} that computes L on input length $\ell_1 \ll \ell_0$. Consequently, using that we have refuter circuits for all input lengths $\geq n_0$, it is enough to iterate the same construction!

Recursion. The circuit B_{ℓ_1} computes analogously to B_{ℓ_0} , by considering a smaller input length $\ell_2 \ll \ell_1$ and a corresponding co-nondeterministic circuit B_{ℓ_2} . Similarly to the analysis from above, we obtain the following constraints to guarantee its correctness and the desired circuit size bound (again omitting small order factors to focus on the relevant asymptotics):

$$\text{size}(E_{\ell_2}) \approx \ell_1 \leq \ell_2^k \quad \text{and} \quad \ell_2^c \leq \ell_1^k/4 \quad \text{and} \quad t(\ell_2) \leq \ell_1^k/4.$$

This forces that

$$\ell_1^{1/k} \leq \ell_2 \leq \ell_1^{k/c} \quad \text{and} \quad t(\ell_1) \leq \ell_1^k/4 + \ell_1^k/4 + t(\ell_2) < 3 \cdot \ell_1^k/4 \leq \ell_0^k/4,$$

where we assumed that $t(\ell_2) \leq \ell_1^k/4$ and used that $\ell_1 \ll \ell_0$. In other words, we need

$$\ell_1^{1/k} \leq \ell_2 \leq \ell_1^{k/c} \quad (\text{thus } c < k^2) \quad \text{and} \quad t(\ell_2) \leq \ell_1^k/4.$$

Parameters and base case. Consider the input lengths $\ell_0 > \ell_1 > \dots > \ell_d$ explored in this way, together with the corresponding circuits $B_{\ell_0}, \dots, B_{\ell_d}$, where each B_{ℓ_i} contains $B_{\ell_{i+1}}$ as a subroutine. In other words, B_{ℓ_0} appears close to the input string, followed by B_{ℓ_1} , and so on. (The string $x_{\ell_{i+1}}$ computed by B_{ℓ_i} serves as the input string to $B_{\ell_{i+1}}$.) We start with an input length ℓ_0 sufficiently larger than n_0 so that all calls to the non-uniform refuter R consider input lengths not smaller than n_0 . In order to satisfy the inequalities from above, our parameters are defined as follows:

- *Sequence of input lengths.* We let $\ell_{i+1} \triangleq \ell_i^{1/k}$. Therefore, $\ell_d = \ell_0^{1/k^d}$.
- *Number of stages.* We take d large enough, so that $\ell_d = \ell_0^{1/k^d} = \log \ell_0$, i.e., $d \triangleq (\log \log \ell_0 - \log \log \ell_d) / \log k = O(\log \log \ell_0)$.
- *Initial input length ℓ_0 .* We want $\ell_d = \log \ell_0 \geq n_0$, so we set $\ell_0 \triangleq 2^{n_0}$.

In the base case (input length ℓ_d and circuit B_{ℓ_d}), we simply consult the hardcoded truthtable of the language L computed by machine M on inputs of length $\ell_d = \log \ell_0$. The truthtable of L on input length $\log \ell_0$ can be nonuniformly stored using ℓ_0 bits. This can be seen as an overhead in the final size of B_{ℓ_0} that is of order $\ell_0 \leq \ell_0^k/4$, where this inequality uses that $k > 1$ and assumes

that $\ell_0 \geq n_0$ is large enough. Since the overall size bound for B_{ℓ_0} is given by a simple additive function obtained from the concatenation of a sequence of circuits, it is not hard to see that its total size is at most ℓ_0^k , as desired. \square

As explained above, this completes the proof of Theorem 3.2. \square

The next lemma will be used in the proof of Theorem 1.1 Item (ii).

Lemma 3.4. *Let $L \in \text{NTIME}[n^c]$ for some constant $c \geq 1$. If there is a polynomial-time refuter R for the lower bound $L \notin \text{i.o.-coNSIZE}[s(n)]$ for some monotone time-constructible function $s : \mathbb{N} \rightarrow \mathbb{N}$ such that $\omega(n^c \cdot \log n) \leq s(n) \leq \text{poly}(n)$, then $L \in \text{P}$.*

Proof. The proof is similar to the construction described above. Consider the following polynomial-time algorithm that aims to solve L . On a given input instance $x \in \{0, 1\}^n$ for L , we construct a $\text{coNSIZE}[O(n^c \cdot \log n)]$ -circuit $D_x : \{0, 1\}^{n/2} \times \{0, 1\}^{n^c} \rightarrow \{0, 1\}$ of the form $D_x(u, v)$ (where v is the nondeterministic input string) that ignores its primary input $u \in \{0, 1\}^{n/2}$ and computes according to $\bar{L}(x)$. Since $L \in \text{NTIME}[n^c]$, the transformation from machines to circuits guarantees that $D_x(u, \cdot)$ is a co-nondeterministic circuit of size at most $O(n^c \cdot \log n) \leq s(n/2)$, assuming that n is sufficiently large. Moreover, a description of D_x can be computed from x and from the nondeterministic machine for L in time polynomial in n .

Using the polynomial-time refuter R on input $(1^{n/2}, D_x)$ and the size upper bound for D_x , we can find a string $z_1 \in \{0, 1\}^{n/2}$ such that $D_x(z_1) \neq L(z_1)$. Using the definition of the circuit D_x , which as a co-nondeterministic circuit satisfies $D_x(u) = 1 - L(x)$ for every input $u \in \{0, 1\}^{n/2}$, this implies that

$$L(z_1) = 1 - D_x(z_1) = 1 - (1 - L(x)) = L(x) .$$

To sum up, in time polynomial in n , we have reduced the problem of deciding L on $x \in \{0, 1\}^n$ to that of deciding L on $z_1 \in \{0, 1\}^{n/2}$. We now recursively evaluate $L(z_1)$ in a similar fashion until the input length is smaller than a large enough constant C . In other words, we produce a sequence z_1, \dots, z_k of inputs, where each $|z_i| = n/2^i$, $k \leq \log n$, $|z_k| \leq C$, and

$$L(x) = L(z_1) = \dots = L(z_k) .$$

Since $L(z_k)$ can be computed in constant time by brute force and there are at most $\log n$ stages of the recursion, it follows that we can decide $L(x)$ in time polynomial in $n = |x|$, i.e., $L \in \text{P}$. \square

We can also obtain the following consequence.

Theorem 3.5 (Restatement of Theorem 1.2 Item (ii)). *There is an integer $k \geq 1$ such that there is no refuter $R(1^n, E_n)$ for SAT_n against $E_n \in \text{coNSIZE}[n^k]$ of nonuniform polynomial size which succeeds on every large enough input length.*

Proof. Assume such a polynomial size refuter exists for $k = 2$, since we are done otherwise. Using that $\text{SAT} \in \text{NTIME}[n \cdot \text{poly}(\log n)]$ and arguing as in the proof of Lemma 3.4, it follows that there is a constant $c \in \mathbb{N}$ and a sequence of non-uniform circuits of size n^c that compute SAT_n on every large enough input length n . But then there is no refuter witnessing that $\text{SAT}_n \notin \text{coNSIZE}[n^k]$ when $k = c + 1$, since this lower bound is simply false when the input length is large enough. \square

3.2 Unprovability of lower bounds via refuter lower bounds

In this section, we prove each item of Theorem 1.1. We will need the following witnessing result for the intuitionistic theory IS_2^1 .

Theorem 3.6 (Witnessing Theorem for IS_2^1 [Bus86b, CU93]). *Let φ be an arbitrary formula, and suppose that*

$$\text{IS}_2^1 \vdash \forall x \exists y \varphi(x, y).$$

Then there is a polynomial-time computable function f such that

$$\mathbb{N} \models \forall a \varphi(a, f(a)).$$

Furthermore, there is a Σ_1^{b+} -formula $\psi(x, y)$ such that:

$$(i) \text{IS}_2^1 \vdash \forall x \forall y (\psi(x, y) \rightarrow \varphi(x, y)).$$

$$(ii) \text{IS}_2^1 \vdash \forall x \forall y \forall z (\psi(x, y) \wedge \psi(x, z) \rightarrow y = z).$$

$$(iii) \text{IS}_2^1 \vdash \forall x \exists y \psi(x, y).$$

In particular, Theorem 3.6 shows that the outermost existential quantifier of a sentence of arbitrary quantifier complexity provable in IS_2^1 can be efficiently witnessed by a polynomial-time function.

Theorem 3.7 (Restatement of Theorem 1.1 Part (i)). *Let $\delta > 0$ be a rational number, $n_0 \in \mathbb{N}$, and M be a polynomial-time nondeterministic machine. Then $\text{IS}_2^1 \not\vdash \text{LB}^{\text{exp}}(M, 2^{n^\delta}, n_0)$.*

Proof. We argue by contradiction. Under the assumption that

$$\text{IS}_2^1 \vdash \text{LB}^{\text{exp}}(M, 2^{n^\delta}, n_0)$$

for a choice of M , $\delta > 0$, and n_0 , it follows from Theorem 3.6 that there is a refuter $R(1^n, D)$ that runs in time $2^{O(n)}$ and outputs an input $x \in \{0, 1\}^n$ such that $\text{Error}_M(D, n, x)$, whenever D is a co-nondeterministic circuit of size at most 2^{n^δ} and $n \geq n_0$. By Lemma 3.1, for any choice of $\varepsilon > 0$, we get that $L(M) \in \text{DTIME}[2^{n^\varepsilon}]$. Taking $\varepsilon < \delta$, this upper bound and the provability of $\text{LB}^{\text{exp}}(M, 2^{n^\delta}, n_0)$ contradict the soundness of IS_2^1 . \square

Theorem 3.8 (Restatement of Theorem 1.1 Part (ii)). *Let $n_0 \in \mathbb{N}$ and M be a polynomial-time nondeterministic machine. Then there is an integer $k \geq 1$ such that $\text{IS}_2^1 \not\vdash \text{LB}^{\text{poly}}(M, n^k, n_0)$.*

Proof. Let n^c be an upper bound on the nondeterministic time complexity of M . Towards a contradiction, assume that $\text{IS}_2^1 \vdash \text{LB}^{\text{poly}}(M, n^k, n_0)$ for every $k \geq 1$. In particular, this holds for $k = 2c$. It follows from Theorem 3.6 that there is a refuter $R(1^n, D)$ that runs in time $\text{poly}(n)$ and outputs an input $x \in \{0, 1\}^n$ such that $\text{Error}_M(D, n, x)$, whenever D is a co-nondeterministic circuit of size at most n^k and $n \geq n_0$. Consequently, by Lemma 3.4, we get that $L(M) \in \text{P}$. In particular, the sentence $\text{LB}^{\text{poly}}(M, n^k, n_0)$ is *false* for some large enough constant k . This and the assumption that $\text{IS}_2^1 \vdash \text{LB}^{\text{poly}}(M, n^k, n_0)$ for every $k \geq 1$ contradict the soundness of IS_2^1 . \square

4 Unprovability of Upper Bounds in IS_2^1

In this section, we prove unconditionally that a natural formalization of the circuit *upper bound* $\text{NP} \subseteq \text{coNSIZE}[n^k]$ is unprovable in IS_2^1 , for every fixed constant k .

4.1 Unconditional uniform lower bound against co-nondeterministic circuits

We first prove that $\text{NP} \not\subseteq \text{P-uniform coNSIZE}[n^k]$, for every constant k . That is, for each k , there is a language $L \in \text{NP}$ that cannot be decided by any polynomial-time uniform family of co-nondeterministic circuits of size n^k . This uniform lower bound is obtained through an adaptation of a technique of Santhanam and Williams [SW14] showing that $\text{P} \not\subseteq \text{P-uniform SIZE}[n^k]$.

First, we will need the following standard “no complementary speedup theorem”.

Theorem 4.1 (Folklore). *For every constant $b \geq 1$, $\text{coNTIME}[n^{b+1}] \not\subseteq \text{NTIME}[n^b]_{/o(n)}$.*

Proof Sketch. Fix any constant $b \geq 1$. Consider the following co-nondeterministic Turing machine M that runs in time $O(n^{b+1})$: On any input of the form $x = (\hat{M}, \alpha, \pi) \in \{0, 1\}^n$, where \hat{M} is interpreted as the encoding of a (clocked) nondeterministic Turing machine running in time $O(n^b)$, α_n is interpreted as the advice to \hat{M} on input length n , and $\pi \in \{0\}^*$ is a padding string, M simulates $\hat{M}(x)_{\alpha_n}$ (i.e., \hat{M} on input x with advice string α_n) and accepts if and only if $\hat{M}(x)_{\alpha_n}$ rejects. It is not hard to show that the language $L(M) \notin \text{NTIME}[n^b]_{/o(n)}$. \square

Next, we introduce an auxiliary definition and establish a proposition that will be useful.

Definition 4.2. Let $C = \{C_n\}_{n \geq 1}$ be a family of polynomial-size nondeterministic circuits, and $m = m(n)$ be a constructive function. The positive and negative m -padded *direct connect language* for C , short for m -pDCL(C) and m -nDCL(C), are defined as follows:

$$\begin{aligned} m\text{-pDCL}(C) &\triangleq \{(n, 1^m, i) \mid \langle C_n \rangle_i = 1\}, \\ m\text{-nDCL}(C) &\triangleq \{(n, 1^m, i) \mid \langle C_n \rangle_i = 0\}, \end{aligned}$$

where $\langle C_n \rangle$ is the string that encodes the circuit C_n .

Proposition 4.3. *If $C = \{C_n\}_{n \geq 1}$ is a P-uniform family of (nondeterministic) circuits, the padded direct connect languages $n^\varepsilon\text{-pDCL}(C), n^\varepsilon\text{-nDCL}(C) \in \text{P}$ for any fixed constant $\varepsilon \in (0, 1)$.*

Proof. Given $(n, 1^{n^\varepsilon}, i)$, one can first print the description $y = \langle C_n \rangle$ of C_n in $\text{poly}(n)$ time, then check whether the i -th bit is 0 or 1. Since the input length is at least n^ε , this algorithm runs in polynomial time. \square

Theorem 4.4. *For every positive integer k , $\text{NP} \not\subseteq \text{P-uniform coNSIZE}[n^k]$.*

Proof. We argue that $\text{coNP} \not\subseteq \text{P-uniform NSIZE}[n^k]$. The theorem follows from this separation by a simple complementation argument. Let k be any constant, and H_b be the hard language in Theorem 4.1 for some $b > k$ to be determined later. Towards a contradiction, we assume that for every language $L \in \text{coNP}$, there is a family of P-uniform nondeterministic circuits $C = \{C_n\}_{n \geq 1}$ of size cn^k that decides L , where c is an arbitrary constant that can depend on L . In particular, this uniform fixed polynomial upper bound holds for H_b . We will use the fixed polynomial upper bound for coNP to optimize H_b to the extent that it violates Theorem 4.1.

1. (*Pad Down*). Let $\varepsilon \triangleq 1/2k$ and $C = \{C_n\}_{n \geq 1}$ be the P-uniform family of nondeterministic circuits of size cn^k that decides H_b . By Proposition 4.3, we know that $n^\varepsilon\text{-pDCL}(C)$ and $n^\varepsilon\text{-nDCL}(C)$ are in P, and therefore also in coNP.
2. (*Compress C_n*). Let $m = m(n) \leq O(\log n) + n^\varepsilon$ be the length of the input $(n, 1^{n^\varepsilon}, i)$ for $n^\varepsilon\text{-pDCL}(C)$ and $n^\varepsilon\text{-nDCL}(C)$. Since $n^\varepsilon\text{-pDCL}(C) \in \text{coNP}$, by the upper bound for coNP, we get that there is a family of nondeterministic circuits $D^p = \{D_m^p\}_{m \geq 1}$ of size $c'm^k$ that decides $n^\varepsilon\text{-pDCL}(C)$. Similarly, there is a family of nondeterministic circuits $D^n = \{D_m^n\}_{m \geq 1}$ of size $c''m^k$ that decides $n^\varepsilon\text{-nDCL}(C)$. Note that the sizes of both circuits D_m^p and D_m^n are $O(m^k) \leq O(n^{\varepsilon k}) = O(\sqrt{n})$, which gives a succinct representation of C_n .
3. (*Speedup with Advice*). Now we speedup the computation of H_b with C_n, D_m^p , and D_m^n . Let $M'_{/\alpha_n}$ be the following nondeterministic algorithm with advice $\{\alpha_n\}_{n \geq 1}$ of length $o(n)$:

- The advice is defined as $\alpha_n \triangleq (D_m^p, D_m^n)$.
- Let $\alpha_n = (D_m^p, D_m^n)$ be the advice and $x \in \{0, 1\}^n$ be the input. Let $\ell \triangleq |\langle C_n \rangle|$. Note that since C_n is of size $O(n^k)$, $\ell \leq \tilde{O}(n^k)$. The algorithm nondeterministically guesses a string $y \in \{0, 1\}^\ell$ for every $i \in [\ell]$, then verifies that for every $i \in [\ell]$, $y_i = \langle C_n \rangle_i$. Concretely, for every $i \in [\ell]$, the algorithm works as follows: if $y_i = 1$, it simulates $D_m^p(n, 1^{n^\varepsilon}, i)$ and immediately rejects if D_m^p rejects; otherwise, the algorithm simulates $D_m^n(n, 1^{n^\varepsilon}, i)$ and immediately rejects if D_m^n rejects. Note that since both D_m^p and D_m^n are nondeterministic circuits of size $O(\sqrt{n})$, the running time for this step is $O(\ell) + \tilde{O}(\sqrt{n}) = \tilde{O}(n^k)$. After this step, we know that $y = \langle C_n \rangle$.
- Finally, the algorithm simulates $C_n(x)$, which takes $\tilde{O}(n^k)$ time.

In summary, this algorithm takes an advice of length $o(n)$ and simulates $C_n(x)$ in time $\tilde{O}(n^k)$. Since C_n decides H_b , this algorithm also decides H_b . Therefore, $H_b \in \text{NTIME}[n^{k+1}]_{/o(n)}$.

This leads to a contradiction by choosing $b = k + 2$. □

4.2 Unprovability of upper bounds via uniform lower bounds

We now combine the uniform lower bound (Theorem 4.4) and the witnessing theorem for IS_2^1 (Theorem 3.6) to establish the unprovability of $\text{NP} \subseteq \text{coNSIZE}[n^k]$ in IS_2^1 . First, we explain how to formalize this upper bound statement. This is similar to a formalization in [KO17].

Formalization. The fixed polynomial circuit upper bound $\text{NP} \subseteq \text{coNSIZE}[n^k]$ states that for every polynomial-time nondeterministic Turing machine M and every input length n , there is a co-nondeterministic circuit C of size $O(n^k)$ such that $C(x) = M(x)$. For each k , we capture the upper bound statement using a collection of sentences

$$\text{“NP} \subseteq \text{coNSIZE}[n^k]\text{”} \triangleq \{\text{UB}_M(k, c) \mid M \text{ is an NP machine and } c \in \mathbb{N}\},$$

where $\text{UB}_M(k, c)$ is the sentence:

$$\text{UB}_M(k, c) \triangleq \forall n \in \text{Log} \exists C \in \text{coNSIZE}[cn^k] \forall x \in \{0, 1\}^n \neg \text{Error}_M(C, n, x).$$

The sentence $\text{UB}_M(k, c)$ can be expressed in a natural way, similarly to the lower bound sentence LB^{poly} described in Section 2.1.

For a fixed k , we say that a theory T proves “ $\text{NP} \subseteq \text{coNSIZE}[n^k]$ ” if for every polynomial-time nondeterministic Turing machine M there is a constant c such that $T \vdash \text{UB}_M(k, c)$.

We are now ready to prove the main result of this section.

Theorem 4.5. *For every $k \in \mathbb{N}$, $\text{IS}_2^1 \not\vdash$ “ $\text{NP} \subseteq \text{coNSIZE}[n^k]$ ”.*

Proof. Towards a contradiction, we assume that $\text{IS}_2^1 \vdash$ “ $\text{NP} \subseteq \text{coNSIZE}[n^k]$ ” for some $k \in \mathbb{N}$, that is, for every NP machine M there is a constant c_M such that $\text{IS}_2^1 \vdash \text{UB}_M(k, c_M)$. For convenience, we write

$$\text{UB}_M(k, c) = \forall n \in \text{Log} \exists C \in \text{coNSIZE}[c_M n^k] \varphi(n, C),$$

where $\varphi(n, C) \triangleq \forall x \in \{0, 1\}^n \neg \text{Error}_M(C, n, x)$. By the witnessing theorem for IS_2^1 (Theorem 3.6), for each NP machine M , there is a polynomial-time algorithm A_M that takes 1^n as input and outputs a co-nondeterministic circuit C_n of size at most $c_M n^k$ such that $\varphi(n, C_n)$ holds in the standard model \mathbb{N} . In other words, $C = \{C_n\}_{n \geq 1}$ is a P-uniform co-nondeterministic circuit family that decides $L(M)$. (Note that the algorithm takes 1^n instead of n as input as $\forall n \in \text{Log}$ is a shorthand for $\forall v \forall n = |v|$.) This immediately implies that every language in NP can be computable by a P-uniform family of co-nondeterministic circuits of size $O(n^k)$, which is impossible by Theorem 4.4. \square

5 Interpretation and Generalization of Our Results

We make two remarks about the interpretation and generalization of our results.

Interpretation of the unprovability results. In this paper, we established the unprovability of both co-nondeterministic circuit size upper bounds and lower bounds in the intuitionistic theory IS_2^1 . Roughly speaking, the standard interpretation of the results is that the question of whether NP requires large co-nondeterministic circuits cannot be constructively resolved, either because of the *lack of relevant non-logical axioms* or because of *the absence of the law of excluded middle*.

Due to the nature of intuitionistic logic, it could still be the case that while $\text{NP} \subseteq \text{coNSIZE}[n^k]$ is unprovable in IS_2^1 , $\neg \text{“NP} \subseteq \text{coNSIZE}[n^k]\text{”}$ is indeed provable in IS_2^1 , as $\neg \varphi \vdash \varphi$ is not an admissible inference rule in intuitionistic logic. As explained in more detail next, our results can be considered necessary steps towards the classical independence of worst-case lower bounds. (See also [Bus90b, Section 6.2] for related considerations in a different context.)

Robustness of our unprovability results. Recall that the proofs of our unprovability results in Theorem 3.7, Theorem 3.8, and Theorem 4.5 only rely on the soundness of IS_2^1 over the standard model and on the witnessing theorem to extract a polynomial-time algorithm for the *outermost existential quantifier*. Therefore, the unprovability result holds even if one substitutes a subformula of the lower bound or upper bound sentences inside the outermost existential quantifier by any formula that coincides with the intended meaning over the standard model \mathbb{N} .

In particular, the unprovability result holds even if any subformula inside the outermost existential quantifier is replaced by any classically equivalent formula. This suggests that our unprovability result holds in “semi-classical” setting, i.e., classical reasoning is allowed inside the outermost existential quantifier.

A more formal way to phrase the observation is via the well-known *double-negation translation* (i.e. Gödel–Gentzen translation). The double-negation translation of a first-order sentence φ , denoted by $\varphi^{\mathbf{N}}$, is defined inductively by the following rules:

- $A^{\mathbf{N}} \triangleq \neg\neg A$ for atomic A (i.e. A is a predicate);
- $(\varphi \wedge \psi)^{\mathbf{N}} \triangleq \varphi^{\mathbf{N}} \wedge \psi^{\mathbf{N}}$; $(\neg\varphi)^{\mathbf{N}} \triangleq \neg\varphi^{\mathbf{N}}$; $(\varphi \rightarrow \psi)^{\mathbf{N}} \triangleq \varphi^{\mathbf{N}} \rightarrow \psi^{\mathbf{N}}$;
- $(\varphi \vee \psi)^{\mathbf{N}} \triangleq \neg\neg(\varphi^{\mathbf{N}} \vee \psi^{\mathbf{N}})$;
- $(\forall x \varphi)^{\mathbf{N}} \triangleq \forall x \varphi^{\mathbf{N}}$;
- $(\exists x \varphi)^{\mathbf{N}} \triangleq \neg\neg\exists x \varphi^{\mathbf{N}}$.

For a set Π of formulas, we define $\Pi^{\mathbf{N}} \triangleq \{\varphi^{\mathbf{N}} \mid \varphi \in \Pi\}$.

Theorem 5.1 (see, e.g., [Bus98]). *For a set Π of formulas and a formula φ , Π proves φ classically if and only if $\Pi^{\mathbf{N}}$ proves $\varphi^{\mathbf{N}}$ intuitionistically. In particular, φ and ψ are classically equivalent if and only if $\varphi^{\mathbf{N}}$ and $\psi^{\mathbf{N}}$ are intuitionistically equivalent.*

Corollary 5.2. *In Theorem 3.7, Theorem 3.8, and Theorem 4.5, the corresponding unprovability result holds even if the subformula inside the outermost existential quantifier is replaced by its double-negation translation.*

References

- [AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.
- [ABM23] Albert Atserias, Sam Buss, and Moritz Müller. On the consistency of circuit lower bounds for non-deterministic time. In *Symposium on Theory of Computing (STOC)*, pages 1257–1270, 2023.
- [Avi00] Jeremy Avigad. Interpreting classical theories in constructive ones. *Journal of Symbolic Logic*, 65(4):1785–1812, 2000.
- [BKO20] Jan Bydzovsky, Jan Krajíček, and Igor C. Oliveira. Consistency of circuit lower bounds with bounded theories. *Logical Methods in Computer Science*, 16(2), 2020.
- [BM20] Jan Bydzovsky and Moritz Müller. Polynomial time ultrapowers and the consistency of circuit lower bounds. *Arch. Math. Log.*, 59(1-2):127–147, 2020.
- [Bus86a] Samuel R. Buss. *Bounded Arithmetic*. Bibliopolis, 1986.
- [Bus86b] Samuel R. Buss. The polynomial hierarchy and intuitionistic bounded arithmetic. In *Structure in Complexity Theory (CCC)*, pages 77–103, 1986.
- [Bus90a] Samuel R. Buss. A note on bootstrapping intuitionistic bounded arithmetic. *Proof Theory: a selection of papers from the Leeds Theory Programme*, pages 142–169, 1990.
- [Bus90b] Samuel R. Buss. On model theory for intuitionistic bounded arithmetic with applications to independence results. In *Feasible Mathematics: A Mathematical Sciences Institute Workshop, Ithaca, New York, June 1989*, pages 27–47. Springer, 1990.

- [Bus97] Samuel R. Buss. Bounded arithmetic and propositional proof complexity. In *Logic of Computation*, pages 67–121. Springer Berlin Heidelberg, 1997.
- [Bus98] Samuel R. Buss. *Handbook of Proof Theory*. Elsevier, 1998.
- [CH99] Thierry Coquand and Martin Hofmann. A new method for establishing conservativity of classical systems over their intuitionistic version. *Mathematical Structures in Computer Science*, 9(4):323–333, 1999.
- [CJSW21] Lijie Chen, Ce Jin, Rahul Santhanam, and Ryan Williams. Constructive separations and their consequences. In *Symposium on Foundations of Computer Science (FOCS)*, 2021.
- [CK07] Stephen A. Cook and Jan Krajíček. Consequences of the provability of $\text{NP} \subseteq \text{P/poly}$. *Journal of Symbolic Logic*, 72(4):1353–1371, 2007.
- [CKKO21] Marco Carmosino, Valentine Kabanets, Antonina Kolokolova, and Igor C. Oliveira. Learn-uniform circuit lower bounds and provability in bounded arithmetic. In *Symposium on Foundations of Computer Science (FOCS)*, 2021.
- [CU93] Stephen Cook and Alasdair Urquhart. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, 63(2):103–200, 1993.
- [FM98] Fernando Ferreira and António Marques. Extracting algorithms from intuitionistic proofs. *Mathematical Logic Quarterly*, 44:143–160, 1998.
- [GP13] Kaveh Ghasemloo and Ján Pich. A note on natural proofs and intuitionism. *Manuscript*, 2013.
- [GST07] Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. If NP languages are hard on the worst-case, then it is easy to find their hard instances. *Comput. Complex.*, 16(4):412–441, 2007.
- [Har92] Victor Harnik. Provably total functions of intuitionistic bounded arithmetic. *Journal of Symbolic Logic*, 57(2):466–477, 1992.
- [Jeř17] Emil Jeřábek. Proof complexity of intuitionistic implicational formulas. *Annals of Pure and Applied Logic*, 168(1):150–190, 2017.
- [KO17] Jan Krajíček and Igor C. Oliveira. Unprovability of circuit upper bounds in Cook’s theory PV. *Logical Methods in Computer Science*, 13(1), 2017.
- [Koh08] Ulrich Kohlenbach. *Applied Proof Theory - Proof Interpretations and their Use in Mathematics*. Springer Monographs in Mathematics. Springer, 2008.
- [KP89] Jan Krajíček and Pavel Pudlák. Propositional provability and models of weak arithmetic. In *CSL’89: Proceedings of the 3rd Workshop on Computer Science Logic*, pages 193–210. Springer, 1989.
- [Kra97] Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

- [Kra11] Jan Krajíček. On the proof complexity of the Nisan-Wigderson generator based on a hard $\text{NP} \cap \text{coNP}$ function. *Journal of Mathematical Logic*, 11(1), 2011.
- [LO23] Jiayu Li and Igor C. Oliveira. Unprovability of strong complexity lower bounds in bounded arithmetic. In *Symposium on Theory of Computing (STOC)*, 2023.
- [Mon08] Morteza Moniri. On the hierarchy of intuitionistic bounded arithmetic. *Journal of Logic and Computation*, 18(4):625–630, 2008.
- [OS17] Igor C. Oliveira and Rahul Santhanam. Pseudodeterministic constructions in subexponential time. In *Symposium on Theory of Computing (STOC)*, pages 665–677, 2017.
- [Pic15] Ján Pich. Circuit lower bounds in bounded arithmetics. *Annals of Pure and Applied Logic*, 166(1):29–45, 2015.
- [PS21] Ján Pich and Rahul Santhanam. Strong co-nondeterministic lower bounds for NP cannot be proved feasibly. In *Symposium on Theory of Computing (STOC)*, pages 223–233, 2021.
- [Raz95] Alexander A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya: mathematics*, 59(1):205, 1995.
- [RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.
- [SW14] Rahul Santhanam and Ryan Williams. On uniformity and circuit lower bounds. *Computational Complexity*, 23(2):177–205, 2014.

A On the Unprovability of Lower Bounds Against Deterministic Circuits

This section provides an overview of a result from [GP13] showing that, under an appropriate formalization, the unprovability of worst-case lower bounds against *deterministic* Boolean circuits in IS_2^1 yields the unprovability of the same lower bound in S_2^1 . The necessary background on bounded arithmetic can be found in Section 2. In particular, the vocabulary of S_2^1 and IS_2^1 is discussed in Section 2.1.

Formalization. For concreteness, we consider a constant $\delta > 0$ and focus on the size bound 2^{n^δ} . (The actual size bound is not relevant for the result, provided that it can be captured by a sharply bounded formula in the sense described below.) We use $\text{SAT} = \{\text{SAT}_n\}$ to denote the decision version of the satisfiability problem for de Morgan Boolean formulas represented by n -bit strings. We consider a sentence $\text{SAT-DLB}^{\text{exp}}(2^{n^\delta}, n_0)$ which encodes that, for every input length $n \geq n_0$ and for every deterministic circuit D of size at most 2^{n^δ} , there is an input y such that $\text{SAT}_n(y) \neq D(y)$. The formalization will require the following auxiliary formulas. (The variable f appearing in some of them but not referred to is used to guarantee bounded quantification, as discussed later in this section.)

$\text{ONE}(f, y)$. For variables f (a truth-table with $|f| = 2^n$) and y (an index/input of f), we use the formula $\text{ONE}(f, y)$ to determine if the y -th entry of f is 1.

$\text{SAT}(f, y, z)$. For variables y (encoding a Boolean formula) and z (encoding an assignment), the formula $\text{SAT}(f, y, z)$ denotes that y is satisfied by z .

$\text{DECIDES-SAT}(f, y)$. For variables f and y , the formula $\text{DECIDES-SAT}(f, y)$ denotes $\text{ONE}(f, y) \leftrightarrow \exists z \leq |f| \text{SAT}(y, z)$, i.e., f correctly decides the satisfiability of the formula encoded by y . (The intended interpretation is that z is an integer of magnitude at most 2^n and consequently can be viewed as an n -bit string.)

$\text{SIZE}(f, D, n)$. For a variable D (encoding a deterministic Boolean circuit), $\text{SIZE}(f, D, n)$ denotes the formula that checks if D has n inputs and at most 2^{n^δ} gates.

$\text{ERROR}(f, D, w, y)$. For variables f, D, w (a transcript of D 's computation), and y , $\text{Error}(f, D, w, y)$ denotes the formula stating that w correctly encodes the computation of D on y and $f(y) \neq D(y)$.

$\text{INDEX}(f, W, y, w)$. For variables W (a sequence of transcripts), y , and w , $\text{INDEX}(f, W, y, w)$ denotes that the y -th element of W is w .

Some of these formulas might require the specification of additional sub-formulas in order to capture their intended behavior. Most importantly, we note that all these formulas admit *sharply bounded* descriptions due to the presence of the variable f (we will use $n = \|f\|$) and the explicitly provided candidate transcript w (whose correctness is easy to check). (This claim can be somewhat tedious to check; for additional details, see a similar presentation in [GP13].) Next, we let $\text{SAT-DLB}^{\text{exp}}(2^{n^\delta}, n_0)$ denote the following sentence:

$$\begin{aligned} & \forall f \forall n \forall D \forall W \forall w \exists y \leq |f| \\ & \left(n = \|f\| \wedge n \geq n_0 \wedge \text{DECIDES-SAT}(f, y) \wedge \text{SIZE}(f, D, n) \wedge \text{INDEX}(f, W, y, w) \right) \\ & \quad \rightarrow \\ & \text{ERROR}(f, D, w, y) . \end{aligned}$$

Observe that $\text{SAT-DLB}^{\text{exp}}(2^{n^\delta}, n_0)$ is the universal closure of a sharply bounded formula, which is in particular a $\forall\Sigma_1^b$ -formula. It is not hard to see that it correctly captures (over the standard model \mathbb{N}) the statement that $\text{SAT}_n \notin \text{SIZE}[2^{n^\delta}]$ for all $n \geq n_0$.

Finally, we will make use of the following conservation result.

Theorem A.1 (Avigad [Avi00, Theorem 3.17]). S_2^1 is conservative over IS_2^1 for $\forall\Sigma_1^b$ sentences.

As a consequence of this result and of the quantifier complexity of the formalization, the provability of $\text{SAT-DLB}^{\text{exp}}(2^{n^\delta}, n_0)$ in S_2^1 yields its provability in IS_2^1 . In other words, if the lower bound sentence is unprovable in IS_2^1 , it is also unprovable in S_2^1 :

Theorem A.2. For every rational $\delta > 0$ and $n_0 \in \mathbb{N}$, if $\text{IS}_2^1 \not\vdash \text{SAT-DLB}^{\text{exp}}(2^{n^\delta}, n_0)$ then $S_2^1 \not\vdash \text{SAT-DLB}^{\text{exp}}(2^{n^\delta}, n_0)$.

This result should be contrasted with Theorem 1.1 Item (i), which establishes in particular the unprovability in IS_2^1 of a *co-nondeterministic* size lower bound for SAT. Note that the quantifier complexity of the sentences LB^{exp} and LB^{poly} from Theorem 1.1 does not allow us to invoke Theorem A.2 (nor an extension of this conservation result from [CH99]) to derive an unprovability result for S_2^1 .