Electronic Colloquium on Computational Complexity, Report No. 22 (2023)

Unprovability of strong complexity lower bounds in bounded arithmetic

Jiatu Li[*]
Institute for Interdisciplinary Information Sciences
Tsinghua University

Igor C. Oliveira[†]
Department of Computer Science
University of Warwick

March 11, 2023

Abstract

While there has been progress in establishing the unprovability of complexity statements in lower fragments of bounded arithmetic, understanding the limits of Jeřábek's theory APC_1 [Jeř07a] and of higher levels of Buss's hierarchy S^i_2 [Bus86] has been a more elusive task. Even in the more restricted setting of Cook's theory PV [Coo75], known results often rely on a less natural formalization that encodes a complexity statement using a collection of sentences instead of a single sentence. This is done to reduce the quantifier complexity of the resulting sentences so that standard witnessing results can be invoked.

In this work, we establish unprovability results for *stronger theories* and for *sentences of higher quantifier complexity*. In particular, we unconditionally show that APC_1 cannot prove strong complexity lower bounds separating the third level of the polynomial hierarchy. In more detail, we consider non-uniform average-case separations, and establish that APC_1 cannot prove a sentence stating that

$$\forall n \geq n_0 \; \exists \, f_n \in \Pi_3\text{-}\mathsf{SIZE}[n^d] \text{ that is } (1/n)\text{-far from every } \Sigma_3\text{-}\mathsf{SIZE}[2^{n^\delta}] \text{ circuit.}$$

This is a consequence of a much more general result showing that, for every $i \geq 1$, strong separations for $\Pi_i\text{-}\mathsf{SIZE}[\mathsf{poly}(n)]$ versus $\Sigma_i\text{-}\mathsf{SIZE}[2^{n^{\Omega(1)}}]$ cannot be proved in the theory $\mathsf{T}^i_{\mathsf{PV}}$ consisting of all true $\forall\Sigma^b_{i-1}$-sentences in the language of Cook's theory PV.

Our argument employs a convenient game-theoretic witnessing result that can be applied to sentences of arbitrary quantifier complexity. We combine it with extensions of a technique introduced by Krajíček [Kra11] that was recently employed by Pich and Santhanam [PS21] to establish the unprovability of lower bounds in PV (i.e., the case $i = 1$ above, but under a weaker formalization) and in a fragment of APC_1.

[*]Email: lijt19@mails.tsinghua.edu.cn
[†]Email: igor.oliveira@warwick.ac.uk

ISSN 1433-8092

Contents

1 Introduction

Establishing unconditional lower bounds on the complexity of computations is one of the primary goals of the theory of computational complexity. While the field has seen progress in the setting of restricted computational devices, such as constant-depth Boolean circuits (e.g., [Ajt83, FSS84, Hås86, Raz87, Smo87, Wil14]) and monotone Boolean circuits (e.g., [Raz85, And85, AB87]), proving super-linear circuit size lower bounds against general (unrestricted) circuits (see, e.g., [FGHK16, LY22]) and separating complexity classes remain longstanding challenges.

Several barrier results have been proposed to explain why techniques that have been successful in certain settings cannot lead to stronger results. The most well known of them are relativization [BGS75], natural proofs [RR97], and algebrization [AW09] (see also [FLY22, CHO^{+}22] for recent examples). While knowledge of these results provides a practical way to check if some approaches are likely to fail, each of these barriers is formulated in an ad-hoc way and is limited in scope. For instance, the natural proofs barrier does not consider a standard notion of "proof" and can be circumvented using simple reductions (see, e.g., [AK10, OS18, CJW19, CHO^{+}22]). In general, the aforementioned barriers don't really tell if we simply haven't been clever enough to design a better lower bound technique, or if there is a deeper, more fundamental reason for the difficulty of establishing complexity lower bounds and separations.

This motivates the development of a more principled approach to investigate the difficulty of analysing computations and, perhaps more importantly, the intriguing possibility that strong complexity lower bounds might be unprovable from certain mathematical axioms. In order to implement this plan, the first step is to try to understand which logical theories are able to formalise a significant number of results in algorithms and complexity theory. There has been a long and highly successful line of research showing that certain fragments of Peano Arithmetic collectively known as *Bounded Arithmetic* offer a robust class of theories for the formalization of both basic and advanced results in these areas.

Remark 1.1 (Bounded Arithmetic). Theories of bounded arithmetic aim to capture mathematical proofs that manipulate concepts from a given complexity class (e.g., a proof by induction whose inductive hypothesis can be checked in polynomial time). Notable examples include Cook's theory PV [Coo75], which formalises polynomial-time reasoning, Jeřábek's theory APC_1 [Jeř07a], which formalises probabilistic polynomial-time reasoning, and Buss's theories S^i_2 and T^i_2, which correspond to the levels of the polynomial-time hierarchy [Bus86].

The correspondence between these theories and the complexity classes is reflected in several ways. For instance, certain *witnessing results* show that every provably total function in a given theory $\mathsf{T}_{\mathcal{C}}$ (i.e., when $\forall x\ \exists! y\ \varphi(x, y)$ is provable, for certain formulas φ) is computable within the corresponding complexity class $\mathcal{C}$ (i.e., the function $y = f(x)$ is in $\mathcal{C}$). There are also close relationships between theories of bounded arithmetic and propositional proof systems, e.g., propositional translations between proofs of certain sentences in PV and S^1_2 and polynomial-size proofs in the extended Frege proof system (see, e.g, [Bey09] and references therein).

Weaker theories corresponding to more fine-grained complexity classes such as TC^0 and NC^1 and the mathematical theorems provable in each of them have also received considerable attention. For instance, key properties of the elementary integer arithmetic operations can be established in theory VTC^0 [Jer22], expander graphs can be constructed and analyzed in theory VNC^1 [BKKK20], and several results from linear algebra can be formalised in theory VNC^2 [TC21]. We refer to Cook and Nguyen [CN10] and Krajíček [Kra95, Kra19] for more information about bounded arithmetic and the logical foundations of complexity theory.

Complexity Lower Bounds in Bounded Arithmetic. The study and formalization of complexity lower bounds proofs in bounded arithmetic dates back to Razborov [Raz95b, Raz95a]. We refer to Pich [Pic15a] and to Müller and Pich [MP20] for a comprehensive survey of the area. In particular, the latter paper identifies Jeřábek's theory APC_1 [Jeř07a] for probabilistic reasoning as a suitable theory for the formalization of several existing circuit lower bounds. (Informally, APC_1 is defined as the extension of Cook's theory PV

[Coo75] with the dual weak pigeonhole principle for polynomial-time functions.) For instance, APC_1 can prove super-polynomial lower bounds against bounded-depth circuits and against monotone circuits [MP20], establish the PCP Theorem [Pic15b] (also provable in PV), and formalize randomized matching algorithms [LC11].

Given the expressive power of PV and its extensions, *unconditionally* showing that these theories cannot prove a given result is a non-trivial task. Remarkably, in a recent work, Pich and Santhanam [PS21] employed a technique introduced by Krajíček [Kra11] and further elaborated in [Pic15a] to establish that PV cannot show strong complexity lower bounds separating NP and coNP. More precisely, for each fixed non-deterministic polynomial-time machine M, they showed that PV cannot prove an average-case lower bound for $L(M)$ against co-nondeterministic circuits of size $2^{n^{o(1)}}$.

In the same work, [PS21] showed that this unprovability result extends to a certain fragment of APC_1 (see [PS21] for the details and for additional results). They left open the status of the provability of strong complexity lower bounds in APC_1. This theory has also been identified in other papers (e.g., [CKKO21]) as an important test case for unconditional unprovability results. This is unsurprising, given the number of advanced results from algorithms and complexity that can be formulated and proved in APC_1 and its mild extensions (see [Oja04, CN10, Lê14, Pic14, MP20] for many additional examples).

Witnessing Theorems and Quantifier Complexity. The approach of [PS21] crucially relies on the KPT Witnessing Theorem [KPT91], a result that can be used to extract computational information from a proof of a sentence with a small number of quantifier alternations. This and similar results (e.g., Herbrand's Theorem and Buss's Witnessing Theorem) have been extremely useful tools in unprovability results (see, e.g., [CK07, Kra21, CKKO21]). In order to apply the usual formulation of these witnessing theorems, it is crucial to consider sentences with up to four blocks of quantifiers. In particular, for this reason, the machine M in the aforementioned result from [PS21] is quantified outside of the sentence (i.e., in the meta-theory). A similar challenge is faced in other papers that consider the unprovability of complexity statements in bounded arithmetic (see, e.g., [KO17] and the subsequent papers [BM20, BKO20]).

Our Contributions. We obtain (unconditional) unprovability results for *stronger theories* and for *sentences of higher quantifier complexity*. We can summarize our main contributions as follows.

(i) Building on previous works [Kra11, Pic15a, PS21], we establish the unprovability of strong complexity lower bounds in APC_1 and in more expressive theories of bounded arithmetic. The lower bound sentences showed unprovable refers to separations between the levels of the polynomial hierarchy, where we consider a non-uniform setting and an average-case lower bound against sub-exponential size circuits.

(ii) We consider a more natural (and of higher quantifier complexity) formalization of complexity lower bounds compared with [Kra11, Pic15a, PS21]. To achieve this, we employ a convenient game-theoretic witnessing theorem that allows us to extract computational information from proofs of sentences with an arbitrary number of quantifiers. While extensions of the KPT Witnessing Theorem for formulas with more quantifiers have found applications in bounded arithmetic [Pud92, Pud06, BKT14], to our knowledge this is the first time that such a result is used for the unprovability of complexity bounds.

In the next section, we discuss our results in detail.

1.1 Results

Before formally stating our main unprovability result, we introduce the theories $\mathsf{T}^i_{\mathsf{PV}}$ and their common language (vocabulary) $\mathcal{L}_{\mathsf{PV}}$.

Theory $\mathsf{T}^i_{\mathsf{PV}}$ and Language $\mathcal{L}_{\mathsf{PV}}$. We let $\mathcal{L}_{\mathsf{PV}}$ contain the constant symbols 0 and 1, and a function symbol f for every function in FP, the class of polynomial-time computable functions.[1] In particular, $\mathcal{L}_{\mathsf{PV}}$ contains function symbols for the length function $|x|$, addition $+$, etc. $\mathcal{L}_{\mathsf{PV}}$ contains the equality predicate $=$ as its only relation symbol. Note that one can define any polynomial-time computable predicate through its characteristic function, equality, and the constant symbol 1.

For each integer $i \geq 1$, we let $\mathsf{T}^i_{\mathsf{PV}}$ denote the theory of all true (with respect to the standard model $\mathbb{N}$) $\forall\Sigma^b_{i-1}$-sentences over the language $\mathcal{L}_{\mathsf{PV}}$.[2] In particular, the theory $\mathsf{T}^1_{\mathsf{PV}}$ (which is called T_{PV} in [PS21]) is strictly stronger than Cook's theory PV.[3] We provide some examples of sentences provable in $\mathsf{T}^i_{\mathsf{PV}}$ after stating our main result.

Formalization of Lower Bounds. In order to consider the provability of a strong complexity lower bound separating the i-th level of the (non-uniform) polynomial hierarchy, we introduce a sentence $\mathsf{LB}^i(s_1, s_2, m, n_0)$ stating that, for every input length $n \geq n_0$, there is a Π_i-circuit C of size $\leq s_1(n)$ such that, for every Σ_i-circuit D of size $\leq s_2(n)$, we have

$$\Pr_{x\sim\{0,1\}^n}\Big[C(x) = D(x)\Big] \;\leq\; 1 - \frac{m(n)}{2^n}.$$

Here a Π_i-circuit C (similarly for Σ_i circuits) is simply a standard deterministic Boolean circuit $C(x, z_1, \ldots, z_i)$, where we define that

$$C(x) = 1 \quad \text{if and only if} \quad \forall z_1\, \exists z_2\, \ldots\, Q_i z_i\, C(x, z_1, \ldots, z_i) = 1\;.$$

Formally, let Σ_i-$\mathsf{SIZE}[s(n)]$ and Π_i-$\mathsf{SIZE}[s(n)]$ refer to Σ_i-circuits and Π_i-circuits of size $s(n)$, respectively. Let $\mathsf{LB}^i(s_1, s_2, m, n_0)$ denote the following $\mathcal{L}_{\mathsf{PV}}$-sentence:

$$\begin{aligned}&\forall n \in \mathsf{LogLog} \text{ with } n \geq n_0\ \exists C \in \Pi_i\text{-}\mathsf{SIZE}[s_1(n)]\ \forall D \in \Sigma_i\text{-}\mathsf{SIZE}[s_2(n)]\\ &\exists m = m(n) \text{ distinct } n\text{-bit strings } x^1, \ldots, x^m \text{ s.t. } \mathsf{Error}(C, D, x^i) \text{ for all } i \in [m],\end{aligned}$$

where $\mathsf{Error}(C, D, x)$ means that the circuits C and D do not agree on the input x. For the reader that is not familiar with bounded arithmetic, the notation $n \in \mathsf{LogLog}$ essentially means that all bounded quantifiers refer to objects of length up to $\mathsf{poly}(2^n)$. As in [PS21], this makes the unprovability result stronger. Many existing circuit lower bound proofs can be formalized in APC_1 without ever quantifying over objects of length larger than $\mathsf{poly}(n)$ [MP20].

It's easy to see that $\mathsf{Error}(C, D, x)$ is the disjunction of a Σ^b_i-formula (stating that $C(x) = 0 \wedge D(x) = 1$) and a Π^b_i-formula (stating that $C(x) = 1 \wedge D(x) = 0$). We note that, already for $i = 1$, $\mathsf{LB}^i(s_1, s_2, m, n_0)$

[1] For the reader familiar with bounded arithmetic, we note that in our setup considering polynomial-time functions is equivalent to considering polynomial-time algorithms. See Section 2.2 for more details.

[2] This is a standard class of sentences in bounded arithmetic. Informally, it means that the sentence starts with a block of universal quantifiers, followed by $i-1$ blocks of *bounded* quantifiers, i.e., $\forall x \leq t$ or $\exists x \leq t$ for some term t. The formal definition will be given in Section 2.2. (For the specialist, we note that allowing sharply bounded quantifiers would not change our unprovability results.)

[3] We use PV to refer to its first-order formalization [Coo75, KPT91], also denoted by PV_1 by some authors.

is a $\forall\Sigma^b_4$-sentence. In particular, widely used witnessing results such as the KPT Theorem [KPT91] (see Section 2.6 for a review) cannot be directly applied to it.

Main Unprovability Result. Next, we state our main theorem on the unprovability of complexity lower bounds in $\mathsf{T}^i_{\mathsf{PV}}$ and its corollary for APC_1.

Theorem 1.2 (Main Theorem)**.** *For every $i \geq 1$, $n_0 \in \mathbb{N}$, $\delta \in \mathbb{Q} \cap (0,1)$, and $d \geq 1$,*

$$\mathsf{T}^i_{\mathsf{PV}} \nvdash \mathsf{LB}^i(s_1, s_2, m, n_0)\ ,$$

where $s_1(n) = n^d$, $s_2(n) = 2^{n^\delta}$, and $m = 2^n/n$.

Theorem 1.2 extends the result of [PS21] in two directions. Firstly, it establishes the unprovability of strong complexity lower bounds in theories believed to be much stronger than $\mathsf{T}^1_{\mathsf{PV}}$. Secondly, [PS21] considered a weaker formalization that instead of quantifying over the circuit C (inside the sentence) considers a collection of sentences $\{\mathsf{LB}^1_M\}_M$, one for each uniform non-deterministic machine M (quantified over outside the theory).

Example 1.3 (The Strength of Theory $\mathsf{T}^i_{\mathsf{PV}}$). These theories are quite strong already at small values of i, say $i = 3$. Below we give some examples (see Appendix A for a related discussion).

- (i) Fermat's Little Theorem, which states that if $a^p \not\equiv a \pmod{p}$ then there is $1 < d < p$ such that $d \mid p$, is a true $\forall\Sigma^b_1$-sentence in $\mathcal{L}_{\mathsf{PV}}$ and consequently an axiom of $\mathsf{T}^2_{\mathsf{PV}}$. It is unprovable in $\mathsf{T}^1_{\mathsf{PV}}$ (therefore also unprovable in PV) unless factoring is easy (see, e.g., [KP98, CN10]).
- (ii) The Pigeonhole Principle, which states that for every circuit $C\colon [n+1] \to [n]$ there exists $x \neq y$ such that $C(x) = C(y)$, is also an axiom of $\mathsf{T}^2_{\mathsf{PV}}$. It is not hard to show that even the weaker version of this principle (in which the circuit $C\colon [2n] \to [n]$) is unprovable in $\mathsf{T}^1_{\mathsf{PV}}$ unless there is no (public-key) collision-resistant hash functions (see, e.g., [Kra01, Bus08]).
- (iii) The dual Pigeonhole Principle, which states that for every circuit $C\colon [n] \to [n+1]$ there exists $y \in [n+1]$ such that for all $x \in [n]$ we have $C(x) \neq y$, is in $\mathsf{T}^3_{\mathsf{PV}}$. Even the weak version of this principle (in which the circuit $C\colon [n] \to [2n]$) is unprovable in $\mathsf{T}^1_{\mathsf{PV}}$ unless EMPTY [Kor21] (also known as Range Avoidance [RSW22]) can be solved in polynomial time with $O(1)$ circuit-inversion oracle queries.
- (iv) The induction principle for Σ^p_i-predicates is provable in $\mathsf{T}^{i+2}_{\mathsf{PV}}$, while even the induction principle for NP-predicates is unprovable in $\mathsf{T}^1_{\mathsf{PV}}$ unless the polynomial-time hierarchy collapses [KPT91, Bus95, Zam96].

Since every axiom of APC_1 is implied by a true $\forall\Sigma^b_2$-sentence over the language $\mathcal{L}_{\mathsf{PV}}$ in theory $\mathsf{T}^3_{\mathsf{PV}}$ (see Section 2 for the definition of APC_1), every sentence provable in APC_1 is also provable in $\mathsf{T}^3_{\mathsf{PV}}$. Consequently, we get the following corollary to Theorem 1.2, which shows that APC_1 cannot establish strong complexity lower bounds separating the third level of the (non-uniform) polynomial hierarchy.

Corollary 1.4 (Unprovability of Strong Complexity Lower Bounds in APC_1)**.** *For every $n_0 \in \mathbb{N}$, $\delta \in \mathbb{Q} \cap (0,1)$, and $d \geq 1$,*

$$\mathsf{APC}_1 \nvdash \mathsf{LB}^3(s_1, s_2, m, n_0)\ ,$$

where $s_1(n) = n^d$, $s_2(n) = 2^{n^\delta}$, and $m = 2^n/n$.

Corollary 1.4 establishes the first unconditional result showing the unprovability of strong complexity lower bounds in APC_1. Previously, [PS21] obtained an extension of their result to a fragment of APC_1, but left open the provability of the same collection of sentences in APC_1. Our result is incomparable to theirs in this case, since Corollary 1.4 refers to LB^3 (the third level of the non-uniform polynomial hierarchy) instead of $\{\mathsf{LB}^1_M\}_M$.

Remark 1.5 (**Relevance to the Logical Foundations of Complexity Theory**). The hypothesis that $\mathsf{P} \neq \mathsf{PH}$ (which is equivalent to $\mathsf{P} \neq \mathsf{NP}$) can be interpreted as the statement that polynomial time computations cannot simulate a finite number of bounded quantifier alternations. Our unconditional unprovability result, on the other hand, establishes that $\mathsf{T}^i_{\mathsf{PV}}$, the strongest (sound) theory operating with $\forall\Sigma^b_{i-1}$ axioms over $\mathcal{L}_{\mathsf{PV}}$, cannot strongly separate the i-th level of the polynomial hierarchy.

If the lower bound stated by the LB^i sentence is true, our result indicates the existence of a fundamental limitation of this theory in reasoning about computations at the i-th level of the hierarchy and above. In contrast to previous works, which were restricted to subtheories of APC_1, a significant aspect of Theorem 1.2 is showing that this phenomenon is not caused by a potential weakness of the theory at hand.

1.2 Techniques

In order to prove Theorem 1.2, we formulate a game-theoretic witnessing theorem that can be applied to sentences of high quantifier complexity, such as $\mathsf{LB}^i(s_1, s_2, m, n_0)$. Our general framework is similar to an extension of the KPT Witnessing Theorem considered by Buss, Kołodziejczyk, and Thapen [BKT14].

A Game-Theoretic Witnessing Theorem for General Formulas. For a language (vocabulary) $\mathcal{L}$, let $\varphi(x)$ be a bounded $\mathcal{L}$-formula defined as

$$\begin{aligned}\varphi(x) \triangleq\ & \exists y_1 \leq t_1(x)\, \forall x_1 \leq s_1(x, y_1)\, \exists y_2 \leq t_2(x, y_1, x_1) \ldots \forall x_{k-1} \leq s_{k-1}(x, y_1, x_1, \ldots, y_{k-1}) \\ & \exists y_k \leq t_k(x, y_1, x_1, \ldots, y_{k-1}, x_{k-1})\, \forall x_k \leq s_k(x, y_1, x_1, \ldots, y_k)\, \phi(x, x_1, \ldots, x_k, y_1, \ldots, y_k),\end{aligned}$$

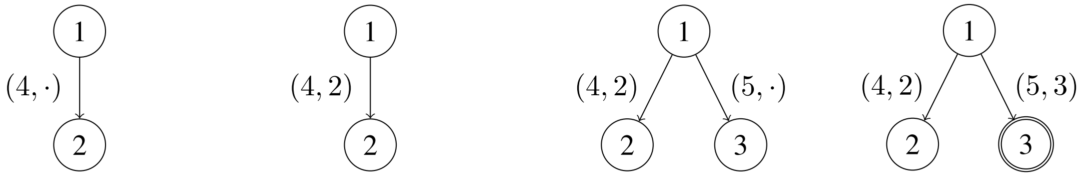
where $\phi(x, \vec{x}, \vec{y})$ is a quantifier-free $\mathcal{L}$-formula. We would like to extract computational information from the provability of $\forall x\, \varphi(x)$ in a theory $\mathcal{T}$. We achieve this by showing that the provability of this sentence is *equivalent* to the existence of a *winning strategy* in a certain *game*. Moreover, the winning strategy will be computable using *terms* of $\mathcal{L}$. Consequently, if the interpretation of each term in a given model $\mathcal{M}$ of $\mathcal{T}$ has limited computational complexity, we obtain a *computationally bounded* winning strategy. For simplicity, we discuss the game only informally below, deferring the formal details to Section 3.

We consider an interactive game between two players, the *truthifier* (associated with existential quantifiers in φ) and the *falsifier* (associated with universal quantifiers in φ). A *board* is defined as a pair $(\mathcal{M}, n_0)$, where $\mathcal{M}$ is a structure over $\mathcal{L}$ such that $\mathcal{M} \vDash \mathcal{T}$, and $n_0 \in \mathcal{M}$ is an element of its domain. The *evaluation game* for the formula $\varphi(x)$ on the board $(\mathcal{M}, n_0)$ is played as follows: in the i-th round of the game ($1 \leq i \leq k$), the truthifier firstly chooses an assignment $m_i \in \mathcal{M}$ for y_i such that $m_i \leq t_i(n_0, m_1, n_1 \ldots, m_{i-1}, n_{i-1})$, then the falsifier chooses an assignment $n_i \in \mathcal{M}$ for x_i such that $n_i \leq s_i(n_0, m_1, n_1, \ldots, m_i)$. The truthifier *wins* if and only if $\phi(x/n_0, \vec{x}/\vec{n}, \vec{y}/\vec{m})$ holds in $\mathcal{M}$. (Note that when playing on a board $(\mathcal{M}, n_0)$ we set x in $\varphi(x)$ to n_0.)

We will also consider a more general game called the *tree exploration game*. In more detail, we allow the truthifier and falsifier to simultaneously play different evaluation games over the same board $(\mathcal{M}, n_0)$. The truthifier has a positional advantage over the falsifier: it can decide where to make the next move, i.e., by either

(i) making the next move in one of the current games; or

(ii) starting a new evaluation game over the board $(\mathcal{M}, n_0)$; or



(a) The truthifier adds a child (2) from the root and a label 4.

(b) The falsifier's response is 2. Node (2) is not a winning node.

(c) The truthifier adds a child (3) from the root and a label 5.

(d) Clearly a winning node is reached regardless of the falsifier's move.

Figure 1: A transcript of the tree exploration game for $\varphi(x) = \exists y \leq 2x\, \forall z < y\, (y \geq x \wedge (z = 1 \vee z \nmid y))$ ("there is a prime number within $[x, 2x]$") on the board $(\mathbb{N}, 3)$. The truthifier wins by reaching node (3).

(iii) playing differently some earlier play, which creates a new game from that position but maintains the existing game plays.

The falsifier must respond to the move of the truthifier in the corresponding evaluation game. Note that the next assignment selected by each player now depends on previous plays in all concurrent games. The truthifier *wins* the tree exploration game if there is a node u in the current partial game tree that is a winning node for the truthifier, that is, the concatenation of the pairs of elements labelling the edges on the root-to-u path forms a winning transcript of the truthifier in the evaluation game of $\varphi(x)$ on the board $(\mathcal{M}, n_0)$. The *tree exploration game of* $\varphi(x)$ is defined as the tree exploration game starting from a partial game tree containing only the root node. See Figure 1 for an example of a transcript of the tree exploration game.

An $\mathcal{L}$*-strategy* of the truthifier in the tree exploration game is described by a sequence of $\mathcal{L}$-terms, where each term describes the next move of the truthifier. Finally, a length-ℓ $\mathcal{L}$-strategy is said to be a *universal winning strategy* if the truthifier wins within ℓ moves against all strategies (not necessarily $\mathcal{L}$-strategies) of the falsifier on any board $(\mathcal{M}, n_0)$. (The "universality" of the strategy comes from the fact that it succeeds over any board $(\mathcal{M}, n_0)$ and against any strategy of the falsifier. Moreover, the location of the next move of the truthifier in the game tree will be independent of the board and of the strategy of the falsifier.)

Recall that a theory $\mathcal{T}$ is said to be a universal theory if every axiom of $\mathcal{T}$ is of the form $\forall \vec{z}\, \psi(\vec{z})$, where $\psi(\vec{z})$ is a formula free of quantifiers. We show that the provability of the sentence $\forall x\, \varphi(x)$ in a universal theory $\mathcal{T}$ with a certain closure property is equivalent to the existence of a universal winning $\mathcal{L}$-strategy of length $O(1)$ for the truthifier in the tree exploration game of $\varphi(x)$.

Theorem 1.6 (Game-Theoretic Witnessing Theorem)**.** *Let $\mathcal{T}$ be a universal bounded theory with vocabulary $\mathcal{L}$ that is closed under if-then-else (see Definition 2.2). Let φ be a bounded $\mathcal{L}$-formula of the form*

$$\begin{aligned}\varphi(x) \;\triangleq\; & \exists y_1 \leq t_1(x)\, \forall x_1 \leq s_1(x, y_1)\, \exists y_2 \leq t_2(x, y_1, x_1) \ldots \forall x_{k-1} \leq s_{k-1}(x, y_1, x_1, \ldots, y_{k-1}) \\ & \exists y_k \leq t_k(x, y_1, x_1, \ldots, y_{k-1}, x_{k-1})\, \forall x_k \leq s_k(x, y_1, x_1, \ldots, y_k)\, \phi(x, x_1, \ldots, x_k, y_1, \ldots, y_k),\end{aligned}$$

where $\phi(x, \vec{x}, \vec{y})$ is a quantifier-free $\mathcal{L}$-formula. Then $\mathcal{T} \vdash \forall x\, \varphi(x)$ if and only if there is a universal winning $\mathcal{L}$-strategy of length $O(1)$ for the truthifier in the corresponding tree exploration game of $\varphi(x)$.

Beyond its applicability to sentences with an arbitrary number of quantifiers, we stress that two keys aspects of Theorem 1.6 are that the winning strategy is computed by $\mathcal{L}$-terms and that the truthifier wins in constantly many rounds. (In practice, in order to use this result to obtain computational information from a proof, one typically fixes a particular strategy of the falsifier, which depends on the context and intended application.)

It is possible to show that Theorem 1.6 is a generalization of the KPT Witnessing Theorem [KPT91]: If the formula $\varphi(x)$ is an $\exists\forall$-formula, the evaluation game for φ has only one round; this means that the tree exploration game for φ is essentially a sequential repetition of the evaluation game (which is equivalent to the Student-Teacher game given by KPT Witnessing Theorem; see Theorem 2.9 and [KPT91, Pic15a]). Indeed, KPT witnessing can also be derived from a less general result that we present in Section 3.2 as a corollary of Theorem 1.6 and that is sufficient for the proof of Theorem 1.2.

We discuss Theorem 1.6 in detail in Section 3 and Appendix B. In contrast to the model-theoretic approach of [BKT14], we establish Theorem 1.6 using techniques from proof theory. As we explain below, in our main application we will actually work with a simplified and more convenient framework that might be of independent interest.

Example 1.7. Why does the provability in a universal theory $\mathcal{T}$ correspond to the tree exploration game instead of the simpler evaluation game? As a conceptual example, one may consider the well-known non-constructive proof of the existence of two irrational numbers x, y such that x^y is rational. By the Law of Excluded Middle (i.e., A or $\neg A$), one can easily argue that either $(x, y) = (\sqrt{2}, \sqrt{2})$ or $(x, y) = ((\sqrt{2})^{\sqrt{2}}, \sqrt{2})$ will be the required pair of irrational numbers. However, we cannot figure out which one of these two possibilities is the correct answer from the structure of this proof. Nevertheless, we can convince any opponent that the original statement is true by a two-round "tree exploration game": we first propose $(x, y) = ((\sqrt{2})^{\sqrt{2}}, \sqrt{2})$ and, in case that the opponent argues that $(\sqrt{2})^{\sqrt{2}}$ is rational, we propose $(\sqrt{2}, \sqrt{2})$ instead. Similarly, the truthifier's strategy extracted from the G3c-proof is not guaranteed to witness the existential quantifiers in one shot; it might need to interact with the falsifier for constantly many rounds to produce a correct answer (and each current move of the truthifier can depend on previous moves of both players).

Unprovability of Strong Complexity Lower Bounds. The proof of Theorem 1.2 extends the approach of [PS21], which explores a technique from [Kra11, Pic15a]. The main challenge for us is that we must consider the significantly more powerful theory $\mathsf{T}^i_{\mathsf{PV}}$ and the (un)provability of a sentence $\mathsf{LB}^i(s_1, s_2, m, n_0)$ with a larger number of quantifier alternations. In particular, while [PS21] considered the provability of a strong complexity lower bound against a fixed machine M, the sentence $\mathsf{LB}^i(s_1, s_2, m, n_0)$ merely states that there exists a strong separation between Π_i circuits vs Σ_i circuits. This introduces an additional technical difficulty that requires us to also revisit and extend the approach of [Kra11, Pic15a].

Suppose, toward a contradiction, that

$$\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(s_1, s_2, m, n_0) \ ,$$

where $s_1(n) = n^d$, $s_2(n) = 2^{n^\delta}$, and $m = 2^n/n$. In other words, we assume that the theory $\mathsf{T}^i_{\mathsf{PV}}$ proves that for every $n \geq n_0$ there is a Π_i-circuit C_n of size $\leq n^d$ such that, for every Σ_i-circuit D_n of size $\leq 2^{n^\delta}$,

$$\Pr_{x \sim \{0,1\}^n} \Big[C_n(x) = D_n(x) \Big] \ \leq \ 1 - \frac{1}{n}.$$

The key idea behind the argument is that the proof of a strong complexity *lower bound* in bounded arithmetic yields a corresponding complexity *upper bound*. We then argue that the lower bound and the upper bound *contradict each other*. From this, the unprovability of the lower bound sentence follows.

In more detail, our high-level strategy is as follows:

(*i*) The provability of the average-case lower bound sentence $\mathsf{LB}^i(s_1, s_2, m, n_0)$ implies the provability in $\mathsf{T}^i_{\mathsf{PV}}$ of a *worst-case* lower bound for Π_i-$\mathsf{SIZE}[n^d]$ vs Σ_i-$\mathsf{SIZE}[2^{n^\delta}]$. The latter is formalized by a sentence $\mathsf{LB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$.

(*ii*) From any $\mathsf{T}^i_{\mathsf{PV}}$-proof of $\mathsf{LB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$, we show how to extract a *complexity upper bound* for an *arbitrary* Π_i-circuit $E_m(x)$ over an input x of length m and of size at most $\mathsf{poly}(m)$. (This is done outside the theory $\mathsf{T}^i_{\mathsf{PV}}$.) More precisely, we show that there is a deterministic circuit B_m with Σ^p_{i-1}-oracle gates and of size $\leq 2^{m^{o(1)}}$ such that

$$\Pr_{x\sim\{0,1\}^m}[E_m(x) = B_m(x)] \geq 1/2 + 2^{-m^{o(1)}}.$$

(*iii*) We invoke a hardness amplification result for the (non-uniform) polynomial hierarchy to conclude that, on any large enough input length n, *every* Π_i-circuit C_n of size $\leq n^d$ agrees with some Σ_i-circuit D_n of size $\leq 2^{n^\delta}$ on more than a $1 - 1/n$ fraction of the inputs. (If this is not the case, we would be able to use hardness amplification to contradict the previous item.)

Since $\mathsf{T}^i_{\mathsf{PV}}$ is a *sound* theory, i.e., every theorem of $\mathsf{T}^i_{\mathsf{PV}}$ is a true sentence, Item (*iii*) is in contradiction with the complexity lower bound stated in $\mathsf{LB}^i(s_1, s_2, m, n_0)$. Consequently, $\mathsf{T}^i_{\mathsf{PV}}$ does not prove this sentence.

Item (*i*) is trivial, since the provability of an average-case lower bound immediately yields the provability of a worst-case lower bound against circuits of the same size. Item (*iii*) requires an extension of a hardness amplification result of Healy, Vadhan, and Viola [HVV06] to higher levels of the polynomial hierarchy. We verify that this is possible in Section 2.5 and Appendix C. The most challenging step of the proof is Item (*ii*), which we discuss next.

General upper bounds from the provability of a complexity lower bound. In Item (*ii*) we aim to extract computational information from a proof of $\mathsf{LB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$ in $\mathsf{T}^i_{\mathsf{PV}}$. For this, we would like to invoke our game-theoretic witnessing theorem (Theorem 1.6). Since this result can only be applied to a *universal theory*, the first step is to introduce a convenient universal theory that is at least as powerful as $\mathsf{T}^i_{\mathsf{PV}}$. Using standard techniques from logic and similarly to [KPT91], we construct a universal theory $\mathsf{UT}^i_{\mathsf{PV}}$ with all the necessary properties (see Theorem 2.18 in Section 2.7). While the axioms of $\mathsf{UT}^i_{\mathsf{PV}}$ are structurally simpler (i.e., universal sentences), the terms of $\mathsf{UT}^i_{\mathsf{PV}}$ no longer correspond to polynomial-time functions. However, a careful construction of $\mathsf{UT}^i_{\mathsf{PV}}$ ensures that its terms (when interpreted over the standard model) correspond to functions in $\mathsf{FP}^{\Sigma^p_{i-1}}$, which will be sufficient for our purposes. In addition to the (syntactic) simplification of the axioms of $\mathsf{T}^i_{\mathsf{PV}}$, a benefit of $\mathsf{UT}^i_{\mathsf{PV}}$ is that the worst-case lower bound sentence $\mathsf{LB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$, whose quantifier complexity grows with i, simplifies to a $\forall\Sigma^b_4$-sentence $\mathsf{ULB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$ in the vocabulary of $\mathsf{UT}^i_{\mathsf{PV}}$.

Since $\mathsf{ULB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$ is also provable in the universal theory $\mathsf{UT}^i_{\mathsf{PV}}$, we can invoke the game-theoretic witnessing theorem with $\mathcal{T} = \mathsf{UT}^i_{\mathsf{PV}}$ and on the formula $\varphi(x)$ corresponding to $\mathsf{ULB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$. (For this overview, think of x as the input length n.) Consequently, there is a universal winning $\mathcal{L}(\mathsf{UT}^i_{\mathsf{PV}})$-strategy for the truthifier (existential player) in the *tree exploration game* of $\varphi(x)$. In particular, for every input length $n \geq n_0$, the truthifier has a winning strategy computed by functions in $\mathsf{FP}^{\Sigma^p_{i-1}}$ that succeeds within $O(1)$ plays in producing a Π_i-circuit C_n of size $\leq n^d$ that cannot be computed (in the worst case) by Σ_i-circuits of size $\leq 2^{n^\delta}$.

The plan for the remainder of the proof is to fix a *particular strategy of the falsifier*, which will depend on the circuit E_m from Item (*ii*) that we would like to approximate, and to show that using the $\mathsf{FP}^{\Sigma^p_{i-1}}$-computable winning strategy of the truthifier we can obtain a good circuit B_m for E_m.

Similarly, in the simpler context of the Student-Teacher game obtained from the KPT Witnessing Theorem and for a worst-case lower bound sentence that refers to a fixed machine M, [Kra11, Pic15a, PS21] showed that an average-case complexity upper bound follows from the provability of a worst-case lower bound. We provide a simple example of how this can be done in Section 4, when we discuss Student-Teacher games with a single round in the context of [PS21]. For games with more than one round, techniques from pseudorandomness and a more elaborated strategy that employs the Nisan-Wigderson generator [NW94] play an important role in the argument from [Kra11, Pic15a, PS21].

In our context, the following difficulties arise:

(1) We need to consider the considerably more complicated tree exploration game played between the truthifier and the falsifier.

(2) The machine M becomes an arbitrary circuit C' that the falsifier proposes as a candidate hard function, and different circuits can be proposed until the winning strategy of the truthifier succeeds in producing a hard circuit C_n.

We are able to avoid a difficult analysis in Item (1) by considering a simpler setting of the tree exploration game that is sufficient for our purposes. In more detail, when considering the strategy for the falsifier based on the circuit E_m that we would like to approximate, the play of the falsifier in the current node of the game tree only depends on the partial play of the *evaluation game* corresponding to the moves of both players in the root-to-node path of the *tree exploration game*. This simpler framework is developed in Section 3.2, and we believe that it might find more applications in the investigation of the logical foundations of algorithms and complexity theory.

Finally, in order to address Item (2), we show that it is possible to modify the use of the Nisan-Wigderson generator in [Kra11, Pic15a] when defining the strategy of the falsifier so that even if the truthifier changes the candidate hard circuit $O(1)$ times when we execute its winning strategy, we are still able to obtain a non-trivial complexity upper bound for E_m. We refer to Section 5 for the technical details.

1.3 Organization

We intended to make the exposition accessible to a broad audience and in particular to someone that might not be so familiar with bounded arithmetic. The remaining sections of the paper are organised as follows:

- Section 2 fixes notation and presents some basic definitions and useful tools in logic and complexity.
- Section 3 formalizes the game-theoretic witnessing theorem (Theorem 1.6). We defer its proof to Appendix B. A simpler version that is sufficient for the proof of Theorem 1.2 is derived in Section 3.2.
- Section 4 provides an exposition of Krajíček's technique [Kra11] (further elaborated in [Pic15a]) and of the main unprovability result from Pich and Santhanam [PS21] in a language that will be more convenient when discussing our proofs.
- Section 5 combines and extends results from the previous sections in order to establish Theorem 1.2.
- Appendix A discusses provability in the theories $\mathsf{T}^i_{\mathsf{PV}}$ and relates their strength to certain computational assumptions.
- Appendix B contains the proofs of the witnessing theorems. A proof of Theorem 1.6 using sequent calculus appears in Appendix B.1. Appendix B.2 provides a self-contained proof of the witnessing theorem presented in Section 3.2 using Herbrandization instead of sequent calculus.
- Appendices C, D, and E contain omitted proofs from Sections 2 and 5.

Acknowledgements. We would like to thank the anonymous STOC reviewers for their helpful comments and for bringing to our attention the witnessing result presented in [BKT14, Section 2.3]. In addition, we thank Ján Pich for answering questions about [PS21], Anupam Das for a discussion on extracting computational content from proofs, and Junhua Yu for comments on the game-theoretic witnessing argument. We also thank Jan Krajíček for providing valuable feedback on different parts of the paper. Finally, we are grateful to Marco Carmosino, Emil Jeřábek, Valentine Kabanets, Rahul Santhanam, Antonina Kolokolova, and Lijie Chen for related discussions. This work received support from the Royal Society University Research Fellowship URF\R1\191059, the EPSRC New Horizons Grant EP/V048201/1, and the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick.

2 Preliminaries

This section fixes notation and presents some basic definitions and useful tools in logic and complexity.

2.1 Complexity theory

Given a function $t\colon \mathbb{N} \to \mathbb{N}$, we generalize the definition of each level of the polynomial hierarchy to machines that run in time $t(n)$ in the natural way. For a fixed $i \geq 1$, we let $\Pi_i\text{-}\mathsf{TIME}[t]$ denote the set of languages L that admit a deterministic machine A running in time $t(n)$ such that, for every $x \in \{0,1\}^n$,

$$x \in L \quad\Longleftrightarrow\quad \forall z_1 \in \{0,1\}^{\leq t(n)}\, \exists z_2 \in \{0,1\}^{\leq t(n)} \ldots\, Q_i z_i \in \{0,1\}^{\leq t(n)}\; A(x, z_1, \ldots, z_i) = 1.$$

The class $\Sigma_i\text{-}\mathsf{TIME}$ is defined in an analogous way. This generalises the classes Σ_i^p and Π_i^p corresponding to the i-th level of the polynomial hierarchy.

We consider (non-uniform) Boolean circuits over a standard set of gates of fan-in at most two, such as $\{\wedge, \vee, \neg\}$. The size of a circuit is the number of gates in the circuit. We adopt this convention only for concreteness, as our results are robust and do not depend on specific details of the circuit model. We let $\mathsf{SIZE}[s]$ denote the set of languages that admit non-uniform Boolean circuits of size $s(n)$.

We also consider circuits and corresponding circuit classes obtained by extending deterministic circuits to circuits with a constant number of alternations. For a fixed $i \geq 1$, we say that a language $L \in \Sigma_i\text{-}\mathsf{SIZE}[s]$ if there is a sequence $\{C_n\}_{n\geq 1}$ of deterministic Boolean circuits C_n of size $s(n)$ such that, for every $x \in \{0,1\}^n$,

$$x \in L \quad\Longleftrightarrow\quad \exists z_1 \in \{0,1\}^{s(n)}\, \forall z_2 \in \{0,1\}^{s(n)} \ldots\, Q_i z_i \in \{0,1\}^{s(n)}\; C_n(x, z_1, \ldots, z_i) = 1.$$

The class $\Pi_i\text{-}\mathsf{SIZE}[s]$ is defined in an analogous way. For convenience, we might refer to $\Sigma_1\text{-}\mathsf{SIZE}[s]$ as $\mathsf{NSIZE}[s]$, i.e., the set of languages computed by non-deterministic circuits of size at most $s(n)$. When we write $C(x) = 1$ for a non-deterministic circuit C and input x, we implicitly refer to its acceptance condition, i.e., that there is an input z such that $C(x, z) = 1$. We adopt the analogous convention for co-nondeterministic circuits and for circuit classes with additional alternations.

We will also consider languages that are computed by circuits with oracle gates. For an oracle $\mathcal{O}$, we let $\mathsf{SIZE}^{\mathcal{O}}[s]$ denote the set of languages computed by circuits of size at most s that can also make use of $\mathcal{O}$-oracle gates.

Finally, for convenience we often abuse notation and associate the size of a Boolean circuit to its bit-length, i.e., its description length under a reasonable encoding.

2.2 Logic and bounded arithmetic

We refer to [Bus97] for an introduction to bounded arithmetic and to the textbooks [Kra95, CN10] for a comprehensive treatment. Below we review the relevant definitions and fix notation.

We use $\mathcal{L}(\mathcal{T})$ to denote the language (vocabulary) of a theory $\mathcal{T}$.

For a structure $\mathcal{M}$ over a language $\mathcal{L}$, we often write $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ to explicitly refer to its domain $\mathcal{D}$ and interpretations $\mathcal{I}$. As usual, the $\mathcal{M}$-interpretation of a function symbol $f \in \mathcal{L}$ will be denoted by $f^{\mathcal{M}}$ (similarly for relations and constants). The $\mathcal{M}$-interpretation of an $\mathcal{L}$-term t is also denoted by $t^{\mathcal{M}}$.

Given a formula ψ, we write $\psi(y)$ to explicitly indicate that y *may* be a free variable in ψ. For a formula $\varphi(x)$ and a term t, we write $\varphi(x/t)$ for substitution of the free variable x with t, or simply $\varphi(t)$ if it is clear from the context. Similarly, we use $s(x)$ to denote a term s that may contain x as a free variable, and $s(x/t)$ to denote the substitution of the free variable x with t, or simply $s(t)$ if it is clear.

The language $\mathcal{L}_{\mathsf{PV}}$. In theoretical computer science one typically considers functions and predicates that operate over binary strings. For the computational models considered in this paper, this is equivalent to operations on integers, by identifying each non-negative integer with its binary representation. For convenience, we adopt the latter perspective when introducing the language (vocabulary) $\mathcal{L}_{\mathsf{PV}}$ of theories $\mathsf{T}^i_{\mathsf{PV}}$.

Let $\mathbb{N}$ denote the set of non-negative integers. For $a \in \mathbb{N}$, we let $|a| = \max\{\lceil \log_2(a+1) \rceil, 1\}$ denote the length of the binary representation of a. For a constant $k \geq 1$, we say that a function $f\colon \mathbb{N}^k \to \mathbb{N}$ is computable in polynomial time if $f(x_1, \ldots, x_k)$ can be computed in time polynomial in $|x_1|, \ldots, |x_k|$. Recall that FP denotes the set of polynomial time functions. While this definition refers to a particular model of computation (Turing machines), Cobham [Cob65] proved that FP can be introduced in a machine independent way as the closure of a set of base functions under composition and limited recursion on notation. We briefly review this construction.[4]

Consider the following class $\mathcal{F}_0$ of base functions:

$$c(x) = 0, \quad s_0(x) = 2 \cdot x, \quad s_1(x) = 2x + 1, \quad \pi^i_\ell(x_1, \ldots, x_\ell) = x_i, \quad x \# y = 2^{|x| \cdot |y|}$$

We say that a function $f(\vec{x}, y)$ is defined from functions $g(\vec{x})$, $h_0(\vec{x}, y, z)$, $h_1(\vec{x}, y, z)$, and $k(\vec{x}, y)$ by *limited recursion on notation* if

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}) \\ f(\vec{x}, s_0(y)) &= h_0(\vec{x}, y, f(\vec{x}, y)) \\ f(\vec{x}, s_1(y)) &= h_1(\vec{x}, y, f(\vec{x}, y)) \\ f(\vec{x}, y) &\leq k(\vec{x}, y) \end{aligned}$$

for every sequence $\vec{x}$ and y of natural numbers. Let $\mathcal{F}$ be the least class of functions that contains $\mathcal{F}_0$ and is closed under composition and limited recursion on notation. Cobham [Cob65] proved that $f \in \mathcal{F}$ if and only if $f \in \mathsf{FP}$.

We let $\mathcal{L}_{\mathsf{PV}}$ contain the constant symbols 0 and 1, and a function symbol f for every function in FP. In particular, $\mathcal{L}_{\mathsf{PV}}$ contains function symbols for the length function $|x|$, $\leq$, $+$, etc.[5]

[4] This is not strictly needed in our presentation. We include it here because it provides more intuition about the language of theories $\mathsf{T}^i_{\mathsf{PV}}$ and the typical choice in bounded arithmetic of defining FP over non-negative integers instead of binary strings.

[5] It is also possible to include in $\mathcal{L}_{\mathsf{PV}}$ a function symbol for every polynomial time *algorithm*, where an algorithm can be described from the base functions and operations allowed in Cobham's characterisation. However, this is inessential in our context. The theories $\mathsf{T}^i_{\mathsf{PV}}$ will contain all true universal sentences, and polynomial time algorithms that compute the same function are provably equivalent in these theories.

We use the standard notation $n \in \mathsf{Log}$ and $n \in \mathsf{LogLog}$ for $\exists N\ n = |N|$ and $\exists N\ n = ||N||$, respectively. We define $\forall n \in \mathsf{Log}$ and $\forall n \in \mathsf{LogLog}$ as $\forall N\ \forall n = |N|$ and $\forall N\ \forall n = ||N||$, respectively.

Bounded formulas and theories $\mathsf{T}^i_{\mathsf{PV}}$. A *bounded quantifier* is a quantifier of the form $Qx \leq t$, where $Q \in \{\exists, \forall\}$ and t is an $\mathcal{L}_{\mathsf{PV}}$-term that does not involve x.[6] An $\mathcal{L}_{\mathsf{PV}}$-formula ψ is *bounded* if every quantifier in ψ is bounded.

We will need to introduce a hierarchy of bounded formulas to define the theories $\mathsf{T}^i_{\mathsf{PV}}$. We let $\Sigma^b_0 = \Pi^b_0$ be the set of quantifier-free $\mathcal{L}_{\mathsf{PV}}$-formulas. We then recursively define sets Σ^b_i and Π^b_i of formulas as follows. For each $i \geq 1$, Σ^b_i and Π^b_i constitute the smallest class of $\mathcal{L}_{\mathsf{PV}}$-formulas such that the following conditions hold:

1. $\Sigma^b_{i-1} \cup \Pi^b_{i-1} \subseteq \Sigma^b_i \cap \Pi^b_i$;
2. both Σ^b_i and Π^b_i are closed under Boolean connectives $\wedge$ and $\vee$;
3. if $\psi(\vec{x}) \equiv \exists y \leq t(\vec{x})\ \varphi(\vec{x}, y)$ is a bounded formula and $\varphi \in \Sigma^b_i$, then $\psi \in \Sigma^b_i$;
4. similarly, if $\psi(\vec{x}) \equiv \forall y \leq t(\vec{x})\ \varphi(\vec{x}, y)$ is a bounded formula and $\varphi \in \Pi^b_i$, then $\psi \in \Pi^b_i$;
5. the negation $\neg\psi$ of a formula ψ from Σ^b_i is in Π^b_i and vice versa.

We note that these classes of sentences are often referred to as *strict* Σ^b_i and Π^b_i formulas in the literature, as we do not include sharply bounded quantifiers between bounded quantifiers.

For convenience, we sometimes describe formulas with the implication symbol $\rightarrow$, implicitly assuming that it is expressed using the Boolean connectives appearing above.

Note that to each $\mathcal{L}_{\mathsf{PV}}$-formula $\phi(x_1, \ldots, x_k)$ we can associate a language $L_\phi \subseteq \{0,1\}^*$ consisting of binary encodings of all tuples $(a_1, \ldots, a_k) \in \mathbb{N}^k$ such that $\mathbb{N} \models \phi(a_1, \ldots, a_k)$. It is known that $\phi \in \Sigma^b_i$ if and only if $L_\phi \in \Sigma^p_i$ [Sto76, Wra76, KH82], where Σ^p_i denotes the i-th level of the polynomial hierarchy.

For $j \geq 0$, we let $\forall\Sigma^b_j$ denote the set of $\mathcal{L}_{\mathsf{PV}}$-sentences of the form $\forall\vec{y}\,\varphi(\vec{y})$, where φ is a Σ^b_j-formula. We sometimes write $\Sigma^b_i(\mathcal{L})$, $\Pi^b_i(\mathcal{L})$, and $\forall\Sigma^b_i(\mathcal{L})$ to emphasize the underlying language $\mathcal{L}$ of a class of formulas.

As expected, the intended model of theories $\mathsf{T}^i_{\mathsf{PV}}$ is $\mathbb{N}$, with the interpretation of each function symbol $f \in \mathcal{L}_{\mathsf{PV}}$ as the corresponding polynomial time function. We will refer to $(\mathbb{N}, 0^{\mathbb{N}}, +^{\mathbb{N}}, \ldots)$ as the *standard model*.

Definition 2.1 (Theories $\mathsf{T}^i_{\mathsf{PV}}$)**.** For each integer $i \geq 1$, we let $\mathsf{T}^i_{\mathsf{PV}}$ denote the theory of all true (with respect to $\mathbb{N}$) $\forall\Sigma^b_{i-1}$ sentences over the language $\mathcal{L}_{\mathsf{PV}}$.

In particular, $\mathsf{T}^1_{\mathsf{PV}}$ is the theory of true universal sentences, and we might refer to $\mathsf{T}^1_{\mathsf{PV}}$ just as T_{PV}.

Note that the definition of $\mathsf{T}^i_{\mathsf{PV}}$ consists of only true Σ^b_i-sentences without sharply bounded quantifiers as axioms. However, as we observe in Appendix A.1, this is inessential in our unprovability results, given that the introduction of sharply bounded quantifiers would not make the theories $\mathsf{T}^i_{\mathsf{PV}}$ any stronger.

In order to simplify the presentation of some results, we introduce the following definition.

Definition 2.2 (Closure under if-then-else)**.** A theory $\mathcal{T}$ is *closed under if-then-else* if for every quantifier-free formula $\varphi(\vec{x})$ and terms $t_1(\vec{x})$ and $t_2(\vec{x})$, there exists a term $t(\vec{x})$ such that

$$\mathcal{T} \vdash \big(t(\vec{x}) = t_1(\vec{x}) \wedge \varphi(\vec{x})\big) \vee \big(t(\vec{x}) = t_2(\vec{x}) \wedge \neg\varphi(\vec{x})\big).$$

[6] Bounded quantifiers can be expressed with the usual quantifiers from first-order logic. For instance, a formula $\psi(y)$ of the form $\forall x \leq t(y)\ \varphi(x,y)$ is equivalent to $\forall x\ (x \leq t(y) \rightarrow \varphi(x,y))$. On the other hand, a formula of the form $\exists x \leq t(y)\ \varphi(x,y)$ is equivalent to $\exists x\ (x \leq t(y) \wedge \varphi(x,y))$.

We note that in such a theory the provability of a disjunction $\psi(x, t_1(x)) \vee \psi(x, t_2(x)) \vee \ldots \vee \psi(x, t_k(x))$ yields the provability of $\psi(x, t(x))$, for a quantifier-free formula $\psi(x, y)$. Typical theories of bounded arithmetic (e.g., S^1_2 and $\mathsf{T}^i_{\mathsf{PV}}$) are closed under if-then-else or admit a suitable extension that is closed under this property.

Theory APC_1. In order to formalize certain probabilistic methods and randomised algorithms, Jeřábek [Jeř04, Jeř07a] (following [Kra01]) introduced the theory APC_1 by extending PV with the *dual Weak Pigeonhole Principle* for PV functions, an axiom scheme postulating that there is no PV function $f : [2^n] \to [(1 + 1/n) \cdot 2^n]$ that is surjective.[7] (The notation APC_1 was proposed by [BKT14].) More formally, we define $\mathsf{dWPHP}(f)$ for a function f (with extra parameters) as the sentence[8]

$$\mathsf{dWPHP}(f) \triangleq \forall n \in \mathsf{Log}\ \forall \vec{z}\ \exists y < (1 + 1/n) \cdot 2^n\ \ \forall x < 2^n\ \ f(\vec{z}, x) \neq y. \tag{1}$$

Let $\mathsf{dWPHP}(\mathsf{PV}) \triangleq \{\mathsf{dWPHP}(f) \mid f \in \mathcal{L}_{\mathsf{PV}}\}$. Then $\mathsf{APC}_1 \triangleq \mathsf{PV} + \mathsf{dWPHP}(\mathsf{PV})$. (For a definition of theory PV, see [Kra95] or the equivalent presentation from [Jeř06].) Jeřábek [Jeř04, Jeř05, Jeř07a] developed a sophisticated (but intuitive) framework for approximate counting in APC_1 built on an elegant formalisation of the Nisan-Wigderson PRG [NW94] in this theory.

By counting the quantifier alternations in Equation (1), it is easy to see that $\mathsf{dWPHP}(f)$ is a $\forall\Sigma^b_2$-sentence in $\mathcal{L}_{\mathsf{PV}}$. As a result, APC_1 is a subtheory of $\mathsf{T}^3_{\mathsf{PV}}$. We note that our unprovability result for APC_1 (Corollary 1.4) is quite robust and works with any non-trivial codomain size in the definition of $\mathsf{dWPHP}(f)$, since this does not increase the quantifier complexity of the corresponding sentences.

2.3 Total search problems and the polynomial hierarchy

In this section, we define complexity classes and circuit classes associated with total search problems in the polynomial hierarchy and explore their basic properties.

Recall that a relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is in TFNP and if there is a polynomial $p(n)$ and a polynomial time machine A such that

- For every $x \in \{0,1\}^*$ there is $y \in \{0,1\}^{\leq p(|x|)}$ such that $R(x, y)$ holds. Moreover, any such y is of length at most $p(|x|)$.
- For every pair (x, y) of strings $x, y \in \{0,1\}^*$, $(x, y) \in R$ if and only if $A(x, y) = 1$.

The next definition is a standard generalisation of this class.

Definition 2.3. For $i \geq 1$, we say that a relation $R \in \mathsf{TF}\Sigma^p_i$ if there is a polynomial $p(n)$ and a polynomial time machine A such that the following conditions hold:

- For every $x \in \{0,1\}^*$ there is $y \in \{0,1\}^{\leq p(|x|)}$ such that $R(x, y)$ holds.
- For every pair (x, y) of strings $x, y \in \{0,1\}^*$,

  $$R(x, y) \iff \forall z_1 \in \{0,1\}^{p(|x|)}\ \exists z_2 \in \{0,1\}^{p(|x|)} \ldots Q_{i-1} z_{i-1} \in \{0,1\}^{p(|x|)}\ A(x, y, z_1, \ldots, z_{i-1}).$$

  In other words, $R \in \Pi^p_{i-1}$.

[7] The size of the codomain (with respect to the size of the domain) affects the power of the dual Weak Pigeonhole Principle. This can be a subtle point, as the equivalence between dual Weak Pigeonhole Principles with different codomain sizes is not known to be provable in PV (see [Jeř07b] for more details).

[8] Note that the additional parameter $\vec{z}$ is crucial in the definition of APC_1. If we remove this parameter in the definition of dWPHP, denoted by dWPHP', the theory $\mathsf{PV} + \mathsf{dWPHP}'(\mathsf{PV})$ will be a (possibly) weaker fragment of APC_1 (see, e.g., [PS21]).

We will need the following simulation result.

Theorem 2.4. *For every $i \geq 1$ and $s(n) \geq n$, $\mathsf{SIZE}^{\Sigma^p_{i-1}}[s(n)] \subseteq \Sigma_i\text{-}\mathsf{SIZE}[\mathsf{poly}(s(n))]$.*

Proof. The proof is similar to the well-known inclusion $\mathsf{P}^{\Sigma^p_{i-1}} \subseteq \Sigma^p_i$ (see, e.g., [Pap94, Chapter 17]), and we omit the details. □

2.4 The Nisan-Wigderson generator

In this section, we review basic properties of the Nisan-Wigderson [NW94] pseudorandom generator and fix notation. For an introduction to this generator and to computational pseudorandomness, see [Vad12].

Definition 2.5. A collection $\mathcal{S} = \{S_1, \ldots, S_k\}$ of sets S_i is said to be an (m, ℓ, a)-*design* if

- for every $i \in [k]$, $S_i \subseteq [m]$;
- for every $i \in [k]$, $|S_i| = \ell$; and
- for every $i \neq j \in [k]$, $|S_i \cap S_j| \leq a$.[9]

The *size* of a design $\mathcal{S}$ is defined as the number of sets in $\mathcal{S}$.

Lemma 2.6 (Explicit designs; see, e.g., [NW94, Vad12])**.** *For every constant $c \geq 2$ and every sufficiently large $n \in \mathbb{N}$, there exists an $(n^c, n^{c/2}, n)$-design $\mathcal{S}_{c,n}$ of size 2^n. Moreover, for every fixed c, there is an algorithm that, given a large enough n and an index $i \in [2^n]$, outputs the i-th set $S_i \in \mathcal{S}_{c,n}$ in time* $\mathsf{poly}(n)$.

Recall that, given an (m, ℓ, a)-design $\mathcal{S}$ of size N and a function $f\colon \{0,1\}^\ell \to \{0,1\}$, the *Nisan-Wigderson generator* (NW generator) maps a *seed* $w \in \{0,1\}^m$ into the N-bit string

$$f(w|_{S_1})f(w|_{S_2})\ldots f(w|_{S_N}),$$

where $w|_{S_i}$ is the string of length ℓ obtained from w be selecting the bits indexed by $S_i \in \mathcal{S}$.

It will be convenient to view the NW generator as a Boolean function and to introduce additional notation. For a large constant $c \geq 1$, given a function $h\colon \{0,1\}^{n^{c/2}} \to \{0,1\}$, we will use the NW generator to define a function $\mathsf{NW}_h\colon \{0,1\}^{n^c} \times \{0,1\}^n \to \{0,1\}$. More precisely,

- The seed length is n^c.
- The corresponding design is described by a $2^n \times n^c$ Boolean matrix A where each row has exactly $n^{c/2}$ entries set to 1, and the 1 entries in distinct rows overlap in at most n columns. As stated in Lemma 2.6, designs with these parameters are known to exist. Given a pair $(i,j) \in [2^n] \times [n^c]$, the (i,j)-entry of the corresponding design matrix can be explicitly computed by circuits of size $\mathsf{poly}(n)$ [NW94].
- For a row index $x \in \{0,1\}^n$ of A, we use $J_x \subseteq [n^c]$ to denote the set of $n^{c/2}$ columns of the x-th row of A set to 1.
- It will often be convenient to consider an n^c-bit string w as a function in $\{0,1\}^{[n^c]}$ that maps $[n^c]$ to $\{0,1\}$. If $S_1, S_2 \subseteq [n^c]$ is a partition of $[n^c]$, $a \in \{0,1\}^{S_1}$, and $u \in \{0,1\}^{S_2}$, we let $w = u \cup a$ denote the corresponding n^c-bit string obtained from a and u.[10]

[9] Designs are also called combinatorial designs by some authors. We will use both terms interchangeably.

[10] This notation is consistent with the standard set-theoretic definition of a function as a set of pairs.

- For $x \in \{0,1\}^n$ and strings $a \in \{0,1\}^{n^c - n^{c/2}}$ and $u \in \{0,1\}^{n^{c/2}}$, we let $r_x(a,u)$ denote the string $w = u \cup a$ of length n^c obtained by viewing $a \in \{0,1\}^{[n^c]\setminus J_x}$ and $u \in \{0,1\}^{J_{x^*}}$.
- By fixing the seed $w \in \{0,1\}^{n^c}$ in the NW generator and the function $h \colon \{0,1\}^{n^{c/2}} \to \{0,1\}$, we obtain a function $\mathsf{NW}_h(w) \colon \{0,1\}^n \to \{0,1\}$ in the natural way. Similarly, we can obtain a family $\{\mathsf{NW}_h(w)\}_{w \in \{0,1\}^{n^c}}$ of functions, one for each possible seed w.

2.5 Hardness amplification in the polynomial hierarchy

In order to relax the average-case complexity parameter in our unprovability results, we need a hardness amplification theorem for the polynomial hierarchy. The result stated below can be seen as the "relativised" version of [HVV06] (see also [PS21, Section 3.3]). For completeness, we sketch their proof and explain how to adapt the result to our purpose in Appendix C.

Theorem 2.7. *There is a constant $\gamma > 0$ and $\ell = \ell(n) = \mathsf{poly}(n)$ such that the following holds for every $i \geq 1$. Let $s_1, s_2 \colon \mathbb{N} \to \mathbb{N}$ be non-decreasing functions, where $s_2(n) = n^{\omega(1)}$, and suppose there is a function $f_n \colon \{0,1\}^n \to \{0,1\}$ computable by Σ_i-$\mathsf{SIZE}[s_1(n)]$ circuits (resp. Π_i-$\mathsf{SIZE}[s_1(n)]$ circuits) such that each Σ^p_{i-1}-oracle circuit A_n of size at most $s_2(n)$ satisfies*

$$\Pr_{x \in \{0,1\}^n}[f_n(x) = A_n(x)] \;\leq\; 1 - \frac{1}{n}.$$

Then there exist a function $h_\ell \colon \{0,1\}^\ell \to \{0,1\}$ computable by Σ_i-$\mathsf{SIZE}[\mathsf{poly}(\ell) \cdot s_1(\ell)]$ circuits (resp. Π_i-$\mathsf{SIZE}[\mathsf{poly}(\ell) \cdot s_1(\ell^\gamma)]$ circuits) such that each Σ^p_{i-1}-oracle circuit B_ℓ of size at most $s_2(\ell^\gamma)^\gamma$ satisfies

$$\Pr_{y \in \{0,1\}^\ell}[h_\ell(y) = B_\ell(y)] \;\leq\; \frac{1}{2} + \frac{1}{s_2(\ell^\gamma)^\gamma}.$$

2.6 Herbrand's Theorem and the KPT Witnessing Theorem

In this section, we review standard witnessing theorems previously used to show unprovability results in bounded arithmetic (see, e.g., [CKKO21, PS21]). In all results, we consider a universal theory $\mathcal{T}$ with vocabulary $\mathcal{L}$.[11] (As a concrete example, one can take $\mathcal{T} = \mathsf{PV}$ and $\mathcal{L} = \mathcal{L}_{\mathsf{PV}}$.)

Two quantifiers ($\forall\exists$). The well-known Herbrand's theorem is the simplest witnessing result and can be applied to $\forall\exists$-sentences (see, e.g., Section 2 of [Koh08]).

Theorem 2.8 (Herbrand's Theorem). *Let $\mathcal{T}$ be a universal theory with vocabulary $\mathcal{L}$. If $\mathcal{T} \vdash \forall x\, \exists y\, \varphi(x,y)$ for a quantifier-free $\mathcal{L}$-formula φ, there exist a constant $\ell \geq 1$ and a sequence $t_1, t_2, \ldots, t_\ell$ of $\mathcal{L}$-terms such that*

$$\mathcal{T} \vdash \forall x\, \big(\varphi(x, t_1(x)) \vee \varphi(x, t_2(x)) \vee \cdots \vee \varphi(x, t_\ell(x))\big).$$

In particular, if $\mathcal{T}$ is closed under if-then-else*, then there is an $\mathcal{L}$-term t such that $\mathcal{T} \vdash \forall x\, \varphi(x, t(x))$.*

[11] Recall that a theory $\mathcal{T}$ is *universal* if all its axioms are universal formulas, i.e., a formula of the form $\forall \vec{x}\, \varphi(\vec{x})$, where φ is free of quantifiers.

Three quantifiers ($\forall\exists\forall$). The KPT Witnessing Theorem [KPT91] extends Herbrand's Theorem by providing witnessing functions for the existential quantifier in a provable $\forall\exists\forall$-sentence.

Theorem 2.9 (KPT Witnessing [KPT91])**.** *Let $\mathcal{T}$ be a universal theory with vocabulary $\mathcal{L}$. Suppose that, for a quantifier-free $\mathcal{L}$-formula φ, $\mathcal{T} \vdash \forall x\, \exists y\, \forall z\ \varphi(x, y, z)$. Then there exist a constant $\ell \geq 1$ and a sequence $t_1, \ldots, t_\ell$ of $\mathcal{L}$-terms such that*

$$\mathcal{T} \vdash \forall x\, \forall \vec{z}\ \big(\varphi(x, t_1(x), z_1) \vee \varphi(x, t_2(x, z_1), z_2) \vee \cdots \vee \varphi(x, t_\ell(x, z_1, \ldots, z_{\ell-1}), z_\ell)\big).$$

KPT witnessing has a well-known computational interpretation as an interactive game between a student and a teacher (see, e.g., [Pic15a]). In the first round, the student is given an arbitrary input x, and computes according to the term $t_1(x)$. This computation provides a candidate object y_1. The teacher then replies with an arbitrary "counterexample" z_1 such that $\neg\varphi(x, y_1, z_1)$ holds, whenever such z_1 exists. Note that the next move of the student takes into account previously presented counterexamples, i.e., the term t_2 depends on both x and z_1. According to Theorem 2.9, the game ends in at most ℓ rounds, and the student is guaranteed to succeed, i.e., to output y such that $\varphi(x, y, z)$ holds for every z.

Example 2.10. An example of the interactive game is the proof of the existence of two irrational numbers x, y such that x^y is rational (see Example 1.7), formalized (in some appropriate theory for real numbers) as:

$$\exists x, y \in \mathbb{R}\ \exists p, q \in \mathbb{Z}\ \forall p', q' \in \mathbb{Z}\ \psi(x, y, p, q, p', q'),\ \text{where}$$
$$\psi(x, y, p, q, p', q') \triangleq x^y = p/q \wedge x \neq p'/q' \wedge y \neq p'/q'$$

The student wants to learn x, y, p, q such that $\psi(x, y, p, q, p', q')$ holds for every p', q', with the help of a teacher that finds counterexamples p', q' making $\psi(x, y, p, q, p', q')$ false. The student's strategy (say, extracted from the proof using KPT witnessing) is that:

- In the first round, try $x = (\sqrt{2})^{\sqrt{2}}, y = \sqrt{2}, p = 2, q = 1$, and ask for a counterexample p', q' from the teacher if it failed.
- Since $y \neq p'/q'$, the student knows that $x = p'/q'$. The student can then propose in the second round that $x = \sqrt{2}, y = \sqrt{2}, p = p', q = q'$.

Four quantifiers ($\forall\exists\forall\exists$). It is also known that one can prove a witnessing theorem for $\forall\exists\forall\exists$-sentences using the standard model-theoretical proof of the KPT witnessing theorem.

Theorem 2.11 (KPT Witnessing for $\forall\exists\forall\exists$-Sentences [KPT91])**.** *Let $\mathcal{T}$ be a universal theory with vocabulary $\mathcal{L}$. Let φ be a quantifier-free $\mathcal{L}$-formula, and suppose that $\mathcal{T} \vdash \forall x\, \exists y\, \forall z\, \exists w\ \varphi(x, y, z, w)$. Then there is an $\ell \geq 1$ and a finite sequence $t_1, \ldots, t_\ell$ of $\mathcal{L}$-terms such that*

$$\mathcal{T} \ \vdash\ \forall x, z_1, \ldots, z_k\ \big(\psi(z, t_1(z), z_1) \vee \psi(x, t_2(x, z_1), z_2) \vee \cdots \vee \psi(x, t_\ell(z_1, \ldots, z_{\ell-1}), z_\ell)\big),$$

where $\psi(x, y, z) \triangleq \exists w\ \varphi(x, y, z, w)$.

Five or more quantifiers? Unlike the case of four quantifiers, there is no obvious direct generalization of the KPT witnessing theorem to five or more quantifiers. The intuitive reason is that there is more than one universal quantifier within the outermost existential quantifier that we would like to witness, so the interaction pattern of the student and the teacher, which can provide counterexamples for all but the outermost universal quantifier, becomes much more complicated. This can be mitigated with the use of Herbrandization, as done in Appendix B.2, but the corresponding witnessing results become significantly more involved.

2.7 A universal theory for $\mathsf{T}^i_{\mathsf{PV}}$

There are two immediate issues when trying to show the unprovability of the lower bound sentence LB^i in $\mathsf{T}^i_{\mathsf{PV}}$. Firstly, LB^i contains more quantifier alternations than a typical witnessing theorem can handle (see Section 3). Secondly, $\mathsf{T}^i_{\mathsf{PV}}$ is not a universal theory if $i > 1$, which violates a common assumption in these results. To address the latter, the first step of our argument is to turn $\mathsf{T}^i_{\mathsf{PV}}$ into a universal theory by introducing Skolem functions. In turn, this will allow us to reduce the quantifier complexity of LB^i so that the techniques developed in Section 3 can be applied (see Section 5.2).

Theory $\mathsf{U}^i_{\mathsf{PV}}$ and Language $\mathcal{L}^i_{\mathsf{PV}}$. Let $i \geq 1$. For each $(\Pi^b_{i-1} \cup \Sigma^b_{i-1})$-formula $\alpha(\vec{z})$ over $\mathcal{L}_{\mathsf{PV}}$, we introduce a function symbol f_α interpreted (in the standard model) as the Boolean-valued function

$$f^{\mathbb{N}}_\alpha(\vec{z}) = \begin{cases} 1 & \text{if } \alpha^{\mathbb{N}}(\vec{z}) \text{ holds;} \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, when $i \geq 2$, for each Σ^b_{i-1}-formula $\beta(\vec{x}, y)$ and term t in $\mathcal{L}_{\mathsf{PV}}$, we introduce a function symbol $g_{\beta,t}$ that is interpreted (in the standard model) as the function[12]

$$g^{\mathbb{N}}_{\beta,t}(\vec{x}) = \begin{cases} \text{smallest } y \in \mathbb{N} \text{ s.t. } \beta^{\mathbb{N}}(\vec{x}, y) & \text{if } \exists y \leq t^{\mathbb{N}}(\vec{x})\ \beta^{\mathbb{N}}(\vec{x}, y); \\ 0 & \text{otherwise.} \end{cases}$$

Denote by $\mathcal{L}^i_{\mathsf{PV}}$ the language of $\mathcal{L}_{\mathsf{PV}}$ supplemented with the new function symbols. Let $\mathsf{U}^i_{\mathsf{PV}}$ be the theory consisting of all universal true sentences (over the standard model) in $\mathcal{L}^i_{\mathsf{PV}}$.

Correctness of the extension $\mathsf{U}^i_{\mathsf{PV}}$. Now we show that $\mathsf{U}^i_{\mathsf{PV}}$ is an extension of $\mathsf{T}^i_{\mathsf{PV}}$, that is, every sentence provable in $\mathsf{T}^i_{\mathsf{PV}}$ is also provable in $\mathsf{U}^i_{\mathsf{PV}}$. We state two useful lemmas (see Appendix D for proofs).

Lemma 2.12 (Defining Axioms of $g_{\beta,t}$)**.** *Let $i \geq 2$, $\beta(\vec{x}, y)$ be any Σ^b_{i-1}-formula in $\mathcal{L}_{\mathsf{PV}}$, and t be any term in $\mathcal{L}_{\mathsf{PV}}$. Then $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{x}\, ((\exists y \leq t(\vec{x})\ f_\beta(\vec{x}, y) = 1) \leftrightarrow f_\beta(\vec{x}, g_{\beta,t}(\vec{x})) = 1)$.*

Lemma 2.13 (Defining Axioms of f_α)**.** *For every $i \geq 1$ and $(\Pi^b_{i-1} \cup \Sigma^b_{i-1})$-formula $\alpha(\vec{z})$ in the language $\mathcal{L}_{\mathsf{PV}}$, $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (\alpha(\vec{z}) \leftrightarrow f_\alpha(\vec{z}) = 1)$.*

Theorem 2.14. *For every $i \geq 1$ and $\mathcal{L}_{\mathsf{PV}}$-sentence φ, if $\mathsf{T}^i_{\mathsf{PV}} \vdash \varphi$, then $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi$.*

Proof. To prove this lemma, it is sufficient to show that for every $\varphi \in \mathsf{T}^i_{\mathsf{PV}}$, $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi$. Let $\varphi = \forall \vec{x}\ \alpha(\vec{x})$ be an axiom of $\mathsf{T}^i_{\mathsf{PV}}$, where $\alpha(\vec{x})$ is a Σ^b_{i-1}-formula. By Lemma 2.13, we only need to show that $\mathsf{U}^i_{\mathsf{PV}}$ proves $\forall \vec{x}\ f_\alpha(\vec{x}) = 1$. This follows directly from the fact that $\forall \vec{x}\ f_\alpha(\vec{x}) = 1$ is a true universal sentence in the standard model. □

Complexity of the function symbols in $\mathcal{L}^i_{\mathsf{PV}}$. As we discussed in Section 1.2, we will extract a KPT-style student-teacher game from the provability of the lower bound sentence in the universal theory $\mathsf{U}^i_{\mathsf{PV}}$. In this step, the complexity of the student is determined by the complexity of the standard interpretations of the function symbols in the language $\mathcal{L}^i_{\mathsf{PV}}$, which consists of both the polynomial-time computable functions (i.e. the symbols in $\mathcal{L}_{\mathsf{PV}}$) and the new function symbols f_α and $g_{\beta,t}$. Now we determine the complexity of the functions f_α and Skolem functions $g_{\beta,t}$.

[12] If the reader is somewhat uncomfortable with the possibility that the smallest y satisfying the condition below might be 0, we stress that this is not going to be an issue in our construction – see, e.g., the statement of Lemma 2.12.

Lemma 2.15. *Let* $i \geq 1$*. For every function symbol* f_α *in* $\mathcal{L}^i_{\mathsf{PV}}$*,* $f^{\mathbb{N}}_\alpha : \mathbb{N} \to \{0,1\}$ *is the characteristic function of a language in* $\Pi^p_{i-1} \cup \Sigma^p_{i-1}$*.*

Proof. Recall that each f_α is introduced for a $(\Pi^b_{i-1} \cup \Sigma^b_{i-1})$-formula $\alpha(\vec{z})$ with language $\mathcal{L}_{\mathsf{PV}}$ such that $f^{\mathbb{N}}_\alpha$ is the characteristic function of $\alpha^{\mathbb{N}}$, i.e., for every $\vec{m} \in \vec{\mathbb{N}}$, $f^{\mathbb{N}}_\alpha(\vec{m}) = 1$ if and only if $\alpha^{\mathbb{N}}(\vec{m})$ holds. Since α is a bounded formula and the initial function symbols and relation symbols, when interpreted in the standard model, are polynomial-time computable, it is not hard to see that $\alpha^{\mathbb{N}} \in \Pi^p_{i-1} \cup \Sigma^p_{i-1}$. □

Lemma 2.16. *Let* $i \geq 2$*. For every function symbol* $g_{\beta,t}$ *in* $\mathcal{L}^i_{\mathsf{PV}}$*,* $g^{\mathbb{N}}_{\beta,t} \in \mathsf{FP}^{\Sigma^p_{i-1}}$*.*

Proof. Recall that $g_{\beta,t}$ is introduced for every Σ^b_{i-1}-formula β and term t in the language $\mathcal{L}_{\mathsf{PV}}$, and that $g^{\mathbb{N}}_{\beta,t}(\vec{x})$ finds the minimum y^* such that $\beta^{\mathbb{N}}(\vec{x}, y^*)$ holds if there is $y \leq t(\vec{x})$ such that $\beta^{\mathbb{N}}(\vec{x}, y)$ holds, or outputs 0 otherwise. Note that using a Σ^p_{i-1} oracle we can decide for $0 \leq l \leq r \leq t(\vec{x})$ whether there exists $y \in [l, r]$ such that $\beta^{\mathbb{N}}(\vec{x}, y)$ holds. So we can perform a binary search over $[0, t(\vec{x})]$ to find the minimum y^* such that $\beta^{\mathbb{N}}(\vec{x}, y^*)$ holds or detect that no such element exists. This is an $\mathsf{FP}^{\Sigma^p_{i-1}}$ computation for every $i \geq 2$. □

Theorem 2.17. *Let* $i \geq 1$*. For every* $\mathcal{L}^i_{\mathsf{PV}}$*-term* $t(x_1, \ldots, x_\ell)$*, we have* $t^{\mathbb{N}}(x_1, \ldots, x_\ell) \in \mathsf{FP}^{\Sigma^p_{i-1}}$*.*

Proof. This directly follows from Lemma 2.15 and Lemma 2.16. □

Theory $\mathsf{U}^i_{\mathsf{PV}}$ has almost all properties needed for the proof of our results, except that it is not necessarily closed under if-then-else (Definition 2.2). This is desirable as it simplifies the statement of our witnessing result and its proof. For this reason, we further modify $\mathsf{U}^i_{\mathsf{PV}}$ to guarantee this property.

Theory $\mathsf{UT}^i_{\mathsf{PV}}$ and Language $\mathcal{L}^i_{\mathsf{UT}}$. Let $i \geq 1$, and consider the language $\mathcal{L}^i_{\mathsf{PV}}$ introduced before. We extend $\mathcal{L}^i_{\mathsf{PV}}$ as follows. For every $k \geq 1$ and function $f : \mathbb{N}^k \to \mathbb{N}$ in $\mathsf{FP}^{\Sigma^p_{i-1}}$, we introduce a new function symbol f_{UT}. Then, we let

$$\mathcal{L}^i_{\mathsf{UT}} \triangleq \mathcal{L}^i_{\mathsf{PV}} \cup \{f_{\mathsf{UT}} \mid f \in \mathsf{FP}^{\Sigma^p_{i-1}}\}.$$

Given $\mathcal{L}^i_{\mathsf{UT}}$, we define $\mathsf{UT}^i_{\mathsf{PV}}$ as the theory of all universal sentences in $\mathcal{L}^i_{\mathsf{UT}}$ that are true in the standard model.

Theorem 2.18 (Main Properties of $\mathsf{UT}^i_{\mathsf{PV}}$)**.** *For every* $i \geq 1$*, the theory* $\mathsf{UT}^i_{\mathsf{PV}}$ *satisfies the following properties:*

(i) $\mathsf{UT}^i_{\mathsf{PV}}$ *is a universal theory.*

(ii) *Every* $\mathcal{L}^i_{\mathsf{PV}}$*-sentence provable in* $\mathsf{U}^i_{\mathsf{PV}}$ *is also provable in* $\mathsf{UT}^i_{\mathsf{PV}}$*.*

(iii) *Every* $\mathcal{L}_{\mathsf{PV}}$*-sentence provable in* $\mathsf{T}^i_{\mathsf{PV}}$ *is also provable in* $\mathsf{UT}^i_{\mathsf{PV}}$*.*

(iv) *Let t be an arbitrary* $\mathcal{L}^i_{\mathsf{UT}}$*-term, and consider its interpretation* $t^{\mathbb{N}} : \mathbb{N}^k \to \mathbb{N}$ *over the standard model. Then* $t^{\mathbb{N}} \in \mathsf{FP}^{\Sigma^p_{i-1}}$*.*

(v) $\mathsf{UT}^i_{\mathsf{PV}}$ *is closed under if-then-else.*

(vi) $\mathsf{UT}^i_{\mathsf{PV}}$ *is sound, i.e., every sentence provable in* $\mathsf{UT}^i_{\mathsf{PV}}$ *is true over* $\mathbb{N}$*.*

The proof of the theorem is deferred to Appendix D.

3 Witnessing Theorems for General Formulas

In this section, we introduce a convenient witnessing theorem that works for sentences of arbitrarily high quantifier complexity. As explained in Section 1, the result is used in the proof that strong complexity lower bounds cannot be established in $\mathsf{T}^i_{\mathsf{PV}}$. While it is possible to obtain a general witnessing result that holds for an arbitrary universal theory, due to our main applications we restrict our attention to theories of bounded arithmetic.

3.1 A game-theoretic witnessing theorem

Let $\mathcal{T}$ be a universal bounded theory over the vocabulary $\mathcal{L}$. Let $\varphi(x)$ be a bounded $\mathcal{L}$-formula defined as

$$\begin{aligned}\varphi(x) \triangleq\; & \exists y_1 \leq t_1(x)\, \forall x_1 \leq s_1(x, y_1)\, \exists y_2 \leq t_2(x, y_1, x_1) \ldots \forall x_{k-1} \leq s_{k-1}(x, y_1, x_1, \ldots, y_{k-1}) \\ & \exists y_k \leq t_k(x, y_1, x_1, \ldots, y_{k-1}, x_{k-1})\, \forall x_k \leq s_k(x, y_1, x_1, \ldots, y_k)\, \phi(x, x_1, \ldots, x_k, y_1, \ldots, y_k),\end{aligned}$$

where $\phi(x, \vec{x}, \vec{y})$ is a quantifier-free $\mathcal{L}$-formula.

The Evaluation Game. We consider an interactive game between two players, the *truthifier* (associated with existential quantifiers in φ) and the *falsifier* (associated with universal quantifiers in φ). A *board* is defined as a pair $(\mathcal{M}, n_0)$, where $\mathcal{M}$ is a structure over $\mathcal{L}$ such that $\mathcal{M} \vDash \mathcal{T}$, and $n_0 \in \mathcal{M}$ is an element of its domain.[13] The *evaluation game* for the formula $\varphi(x)$ on the board $(\mathcal{M}, n_0)$ is played as follows: in the i-th round of the game ($1 \leq i \leq k$), the truthifier firstly chooses an assignment $m_i \in \mathcal{M}$ for y_i such that $m_i \leq t_i(n_0, m_1, n_1 \ldots, m_{i-1}, n_{i-1})$, then the falsifier chooses an assignment $n_i \in \mathcal{M}$ for x_i such that $n_i \leq s_i(n_0, m_1, n_1, \ldots, m_i)$. The truthifier *wins* if and only if $\phi(x/n_0, \vec{x}/\vec{n}, \vec{y}/\vec{m})$ holds in $\mathcal{M}$.

The *transcript* of a game given strategies $\tau^{\mathtt{t}}$ for the truthifier and $\tau^{\mathtt{f}}$ for the falsifier, denoted by $\langle \tau^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle$, is a pair $(\vec{n}, \vec{m})$ of sequences that records the moves for both players.[14] A *partial transcript* is a prefix of a transcript. A partial transcript is *valid* if all elements m_i and n_i respect the corresponding upper bounds (in $\mathcal{M}$) given by functions t_i and s_i. A strategy $\tau^{\mathtt{t}}$ for the truthifier is said to *beat* a strategy $\tau^{\mathtt{f}}$ for the falsifier (w.r.t. a given board and formula) if the truthifier wins in the transcript $\langle \tau^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle$. A *strategy* for a player is defined in the natural way, i.e., as a function that produces the next assignment given a partial transcript of the game. Equivalently, since we will consider games with only a fixed number of rounds, one can describe a strategy as a finite sequence of functions of the form $f\colon \mathcal{M}^i \to \mathcal{M}$, for appropriate values $i \leq 2k$.

We will consider games that are played in a more general setting. Roughly speaking, we allow the truthifier and falsifier to simultaneously play different evaluation games over the same board $(\mathcal{M}, n_0)$. The truthifier has a positional advantage over the falsifier: it can decide where to make the next move, i.e., by either making the next move in one of the current games or starting a new game over the board $(\mathcal{M}, n_0)$ or playing differently some earlier play, which creates a new game from there but maintains the existing game plays. The falsifier must respond to that move in the corresponding game. *Crucially, the next assignment selected by each player now depends on previous plays in all games.* The formal details are provided next.

The Tree Exploration Game. A *partial game tree* $T = (V, E, \gamma)$ (where (V, E) is a directed rooted tree and $\gamma\colon E \to \mathcal{M} \times \mathcal{M}$) of the evaluation game for φ on the board $(\mathcal{M}, n_0)$ is defined as a finite rooted tree where

[13] For a concrete example, think of $\mathcal{M} = (\mathbb{N}, \leq^{\mathbb{N}}, +^{\mathbb{N}}, \times^{\mathbb{N}}, \ldots)$.

[14] For convenience, we might also write the transcript as $(m_1, n_1, \ldots, m_k, n_k)$. The moves of each player will always be clear in each context.

each edge $e \in E(T)$ is labeled with a pair (m, n) of elements of $\mathcal{M}$ and such that, for every node $u \in V(T)$, the concatenation of each pair of elements labelling the edges on the root-to-u path is a prefix of a valid transcript of the evaluation game of $\varphi(x)$ on the board $(\mathcal{M}, n_0)$. More precisely, if the pairs labelling the edges from the root to u are $(m_1, n_1), (m_2, n_2), \dots, (m_i, n_i)$, then $(m_1, n_1, m_2, n_2, \dots, m_i, n_i)$ is a valid partial transcript of the evaluation game, i.e., for all $j \in [i]$, $\mathcal{M} \vDash m_j \leq t_j(n_0, m_1, n_1, \dots, m_{j-1}, n_{j-1})$ and $\mathcal{M} \vDash n_j \leq s_j(n_0, m_1, n_1, \dots, m_j)$. Note that if $\mathcal{M}$ is the standard model then a partial game tree of the evaluation game is a finite upper part of the (exponential size) complete game tree of the evaluation game.

Let T be a partial game tree of φ and $(\mathcal{M}, n_0)$ be a board. The *tree exploration game* starting from T on $(\mathcal{M}, n_0)$ is played as follows. In each *round*, first the truthifier chooses a node u from T (not necessarily a leaf) and an element $m \in \mathcal{M}$, then the falsifier chooses an element $n \in \mathcal{M}$. This creates a child of u and a corresponding directed edge labeled by (m, n). Note that when playing each round of the tree exploration game both players should guarantee that the new partial game tree is always a valid partial game tree, i.e., m and n should satisfy the corresponding inequalities. The *size* of a partial game tree T is given by $|T(V)|$.

The truthifier *wins* the tree exploration game if there is a node in the current partial game tree that is a winning node for the truthifier, that is, the concatenation of the pairs of elements labelling the edges on the root-to-u path forms a winning transcript of the truthifier in the evaluation game of $\varphi(x)$ on the board $(\mathcal{M}, n_0)$. The *tree exploration game of* $\varphi(x)$ is defined as the tree exploration game starting from a partial game tree containing only the root node. We refer to Figure 1 for an example of the tree exploration game.

Recall that $\mathcal{L}$ is the vocabulary of the universal (bounded) theory $\mathcal{T}$. The main result established in this section shows the existence of a "computationally bounded" winning strategy for the truthifier from a $\mathcal{T}$-proof of φ, i.e., the strategy can be computed by $\mathcal{L}$-terms. In addition, the strategy is universal, in the sense that it is specified by $\mathcal{L}$-terms that are independent of the board $(\mathcal{M}, n_0)$. Finally, the location of each play of the truthifier in the partial game tree is fixed in advance and does not depend on the strategy of the falsifier nor on the board $(\mathcal{M}, n_0)$. (The elements selected by the truthifier depend on the previous plays of the truthifier and falsifier.) This means that the trees in the sequence of partial game trees are fixed in advance.

$\mathcal{L}$-Strategies for the Tree Exploration Game. An $\mathcal{L}$-*quasi-strategy* of the truthifier of *length* $\ell \in \mathbb{N}$ and initial tree size d is a sequence $\tau = \langle p_1, r_1, p_2, r_2, \dots, p_\ell, r_\ell \rangle$, where each p_i is an $\mathcal{L}$-term and each $r_i \in \mathbb{N}$ is such that $1 \leq r_i < d + i$. Let $(\mathcal{M}, n_0)$ be a board and T be a partial game tree on this board with $V(T) = \{1, 2, \dots, d\}$. The strategy for the tree exploration game starting from the partial game tree T *induced* by τ proceeds as follows:

- In the i-th move, the truthifier introduces a node numbered $d + i$ as a child of the node r_i and chooses the element $v_i \triangleq p_i^{\mathcal{M}}(n_0, \Gamma) \in \mathcal{M}$, where Γ is the sequence of $\mathcal{M}$-elements chosen by the players in previous rounds (i.e., $v_1, \dots, v_{i-1}$ and the falsifier's moves $w_1, \dots, w_{i-1}$).

Note that an arbitrary $\mathcal{L}$-*quasi-strategy* might induce an invalid move $v_i = p_i^{\mathcal{M}}(n_0, \Gamma)$ that violates the desired upper bound on v_i, depending on the moves of the falsifier. We say that an $\mathcal{L}$-quasi-strategy of the truthifier is an $\mathcal{L}$-*strategy* if for every board $(\mathcal{M}, n_0)$ the resulting partial game trees are valid for every valid strategy of the falsifier.

Finally, a length-ℓ $\mathcal{L}$-strategy is said to be a *universal winning strategy* if the truthifier wins within ℓ moves against all valid strategies (not necessarily $\mathcal{L}$-strategies) of the falsifier on any board $(\mathcal{M}, n_0)$. Note that the falsifier's strategy is a function of the board $(\mathcal{M}, n_0)$, partial game tree $T = (V, E, \gamma)$ (which includes all moves from previous rounds), and the move of the truthifier in the current round.

Theorem 3.1 (Game-Theoretic Witnessing Theorem). *Let $\mathcal{T}$ be a universal bounded theory with vocabulary*

$\mathcal{L}$ that is closed under if-then-else. Let φ be a bounded $\mathcal{L}$-formula of the form

$$\begin{aligned}\varphi(x) \triangleq\ & \exists y_1 \leq t_1(x)\, \forall x_1 \leq s_1(x, y_1)\, \exists y_2 \leq t_2(x, y_1, x_1) \ldots \forall x_{k-1} \leq s_{k-1}(x, y_1, x_1, \ldots, y_{k-1}) \\ & \exists y_k \leq t_k(x, y_1, x_1, \ldots, y_{k-1}, x_{k-1})\, \forall x_k \leq s_k(x, y_1, x_1, \ldots, y_k)\, \phi(x, x_1, \ldots, x_k, y_1, \ldots, y_k),\end{aligned}$$

where $\phi(x, \vec{x}, \vec{y})$ is a quantifier-free $\mathcal{L}$-formula. Then $\mathcal{T} \vdash \forall x\ \varphi(x)$ if and only if there is a universal winning $\mathcal{L}$-strategy of length $O(1)$ for the truthifier in the corresponding tree exploration game of $\varphi(x)$.

We defer the proof of Theorem 3.1 to Appendix B.

3.2 A special case: Falsifiers with oblivious strategies

In this section, we present a special case of game-theoretic witnessing (Theorem 3.1) that involves *sequential* invocations of the *evaluation game* played against an *oblivious falsifier*. This version is sufficient to show the unprovability of strong circuit lower bounds in bounded arithmetic (Section 5.2).

We assume familiarity with the notation from Section 3.1. In particular, let $\mathcal{T}$, $\mathcal{L}$, and $\varphi(x)$ be defined as in Section 3.1. The main difference is that here we consider the evaluation game (as opposed to the tree exploration game) in the presence of *ancillary information for the truthifier*, as explained next.

Strategies with Ancillary Information. Let $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ be a model for $\mathcal{T}$. An $\mathcal{L}$-strategy for the truthifier *with ancillary information* in the evaluation game of $\varphi(x)$ is a sequence $\tau^{\mathtt{t}} = (p_1, p_2, \ldots, p_k)$ of k $\mathcal{L}$-terms, where $p_i \triangleq p_i(n_0, m_1, n_1, \ldots, m_{i-1}, n_{i-1}, \vec{a})$ means given the *ancillary information* $\vec{a}$ (constantly many elements from $\mathcal{D}$), $n_0 \in \mathcal{D}$, and moves $m_1, n_1, \ldots, m_{i-1}, n_{i-1} \in \mathcal{D}$, the truthifier chooses $m_i = p_i^{\mathcal{M}}(n_0, m_1, n_1, \ldots, m_{i-1}, n_{i-1}, \vec{a})$ as the current move. For every $\vec{a} \in \vec{\mathcal{D}}$, the strategy induced by $\tau^{\mathtt{t}}$ given $\vec{a}$ as ancillary information is denoted by $\tau^{\mathtt{t}}[\vec{a}]$. In particular, if the $\mathcal{L}$-strategy has no ancillary information, the induced strategy is denoted by $\tau^{\mathtt{t}}[\varnothing]$. Similarly to Section 3.1, the *transcript* of a game given strategies $\tau^{\mathtt{t}}$ for the truthifier (possibly with ancillary information) and $\tau^{\mathtt{f}}$ for the falsifier, denoted by $\langle \tau^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle$, is a pair $(\vec{n}, \vec{m})$ of sequences that records the moves of both players.

Theorem 3.2 (Winning strategies against oblivious falsifiers)**.** *Let $\mathcal{T}$ be a universal theory over the language $\mathcal{L}$ that is closed under if-then-else. Let $\varphi(x)$ be the formula*

$$\begin{aligned}\varphi(x) \triangleq\ & \exists y_1 \leq t_1(x)\, \forall x_1 \leq s_1(x, y_1)\, \exists y_2 \leq t_2(x, y_1, x_1) \ldots \forall x_{k-1} \leq s_{k-1}(x, y_1, x_1, \ldots, y_{k-1}) \\ & \exists y_k \leq t_k(x, y_1, x_1, \ldots, y_{k-1}, x_{k-1})\, \forall x_k \leq s_k(x, y_1, x_1, \ldots, y_k)\, \phi(x, x_1, \ldots, x_k, y_1, \ldots, y_k),\end{aligned}$$

where $\phi(x, \vec{x}, \vec{y})$ is a quantifier-free $\mathcal{L}$-formula. If $\mathcal{T} \vdash \forall x\ \varphi(x)$, then there is a constant $\ell \in \mathbb{N}$ and $\mathcal{L}$-strategies $\tau_1^{\mathtt{t}}, \tau_2^{\mathtt{t}}, \ldots, \tau_\ell^{\mathtt{t}}$ (with ancillary information) such that, for any board $(\mathcal{M}, n_0)$ and evaluation game of $\varphi(x)$ on $(\mathcal{M}, n_0)$, for every strategy $\tau^{\mathtt{f}}$ of the falsifier:

- *either $\hat{\tau}_1^{\mathtt{t}} \triangleq \tau_1^{\mathtt{t}}[\varnothing]$ beats $\tau^{\mathtt{f}}$,*
- *or $\hat{\tau}_2^{\mathtt{t}} \triangleq \tau_2^{\mathtt{t}}[\langle \hat{\tau}_1^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle]$ beats $\tau^{\mathtt{f}}$,*
- *or $\hat{\tau}_3^{\mathtt{t}} \triangleq \tau_3^{\mathtt{t}}[\langle \hat{\tau}_1^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle, \langle \hat{\tau}_2^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle]$ beats $\tau^{\mathtt{f}}$,*
- *$\ldots$,*
- *or $\hat{\tau}_\ell^{\mathtt{t}} \triangleq \tau_\ell^{\mathtt{t}}[\langle \hat{\tau}_1^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle, \langle \hat{\tau}_2^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle, \ldots, \langle \hat{\tau}_{\ell-1}^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle]$ beats $\tau^{\mathtt{f}}$.*

Before we establish this result, a few comments are in order. First, notice that the moves of the falsifier only depend on the previous moves in the *current game*. On the other hand, the truthifier gets as ancillary information the transcripts of all previous games, and succeeds in beating the strategy of the falsifier after (sequentially) playing at most $\ell = O(1)$ games. Intuitively, the falsifier is *oblivious*, since its moves in the current game do not depend on the moves from any previously completed or different game played in parallel, as in the tree exploration game described in Section 3.1. Consequently, when extracting computational information from proofs (where one defines appropriate strategies for the falsifier and considers the behaviour of the truthifier), Theorem 3.2 is more limited than Theorem 3.1.

Proof of Theorem 3.2. Intuitively, as explained above, the meaning of the theorem is that the truthifier has a winning strategy (with ancillary information) in ℓ sequential plays of the evaluation game when the falsifier's strategy is fixed. We will obtain such a strategy from a strategy for the truthifier that succeeds in the tree exploration game. This is not entirely obvious, since there is a mismatch between the games: the next play of the truthifier in the tree exploration game depends on all previous plays in the game tree, while in the evaluation game there is no game tree and they play a sequence of evaluation games.

Let $\mathcal{T}, \mathcal{L}$, and $\varphi(x)$ be defined as above. By Theorem 3.1, there exists an $\ell = O(1)$ length $\mathcal{L}$-term winning strategy τ^{tree} for the tree exploration game of $\varphi(x)$. Let $(\mathcal{M}, n_0)$ be a board. Consider the tree exploration game when the falsifier plays with a *fixed strategy of the evaluation game*, that is, there exist functions $f_1(x, y_1), f_2(x, y_1, y_2), \dots, f_k(x, y_1, y_2, \dots, y_k)$ such that:

- In the i-th step, if the truthifier adds a node v to the partial game tree as a child of u and chooses m as the label and $(m_1, n_1), (m_2, n_2), \dots, (m_d, n_d)$ are the labels on the length-d path from the root to u, then the falsifier's move is $f_{d+1}(n_0, m_1, m_2, \dots, m_d, m)$.

We say that a falsifier's strategy of this form in the tree exploration game is *oblivious*, i.e., the next move of the falsifier only considers moves in the corresponding root-to-node path. Since τ^{tree} is a winning strategy for the tree exploration game, it beats all strategies of the falsifier, including oblivious strategies.

We would like to simulate τ^{tree}, a strategy for the tree exploration game, in the context of Theorem 3.2, where the evaluation game is played sequentially and the truthifier has ancillary information. The main idea is to play each *round* in the *tree exploration game* as a new *game* in the *evaluation game* that simulates the current root-to-node path. This guarantees when translating strategies that all the necessary information from the tree exploration game appears in the transcript of previous plays (ancillary information) during the next evaluation game. (If the root-to-node path at the end of a round in the tree exploration game is only a partial play of the corresponding evaluation game, the truthifier simply outputs $0^{\mathcal{M}}$ in the current evaluation game until a new game can be started.) In other words, when the truthifier adds a node v as the child of u, it can "replay" the path from the root to v using the moves $m_1, m_2, \dots, m_d$ on the path, and the oblivious falsifier will choose the moves $n_1, n_2, \dots, n_d$ as response. Therefore the truthifier can simulate the winning strategy for the tree exploration game by sequentially playing the evaluation game ℓ times and beating the falsifier in at least one of the games.

We now describe in more detail the translation of an $\mathcal{L}$-term universal winning strategy in the tree exploration game into $\mathcal{L}$-strategies (with ancillary information) for the evaluation game. Consider the strategy $\tau^{\mathsf{tree}} = \langle p_1, r_1, p_2, r_2, \dots, p_\ell, r_\ell \rangle$, where ℓ is a constant. Recall that the location of each play of the truthifier in the tree exploration game is fixed, and that $r_1, \dots, r_\ell$ describe the nodes to which a new child is added in each play. For each $i \in [\ell]$, we define an $\mathcal{L}$-term strategy for the evaluation game τ_i^{eval} as follows:

- Let r_i be the node of the game tree that is extended during the i-th play of the tree exploration game. Suppose this node is at the d_i-th level of the tree, and let $p_{i_1}, \dots, p_{i_{d_i}}$ be the $\mathcal{L}$-terms corresponding to the moves of the truthifier in the root-to-r_i path, including the current move.

- Define the following $\mathcal{L}$-strategy τ_i^{eval} of the evaluation game: (1) parse the ancillary information as a sequence Γ of transcripts derived from playing strategies $\tau_1^{\mathsf{eval}}, \ldots, \tau_{i-1}^{\mathsf{eval}}$ with the ancillary information described in the statement of the theorem; (2) in the j-th step (during the i-th evaluation game), where $j \in [k]$, if $j \leq d_i$ play according to p_{i_j} using that all plays from previous rounds of the tree exploration game are available in the transcript Γ. Otherwise, choose $0^{\mathcal{M}}$ (i.e., the j-th term defining the strategy is the constant term 0).

From the discussion above, the correctness of the translation is clear: if the strategies $\tau_1^{\mathsf{eval}}, \tau_2^{\mathsf{eval}}, \ldots, \tau_\ell^{\mathsf{eval}}$ cannot beat a fixed falsifier strategy $\tau^{\mathtt{f}}$ in ℓ sequential plays of the evaluation game, we can use the oblivious strategy defined by $\tau^{\mathtt{f}}$ in the tree exploration game to show that the truthifier does not win the tree exploration game withing ℓ moves. □

Remark 3.3. Instead of viewing Theorem 3.2 as a special case of the game-theoretic witnessing theorem that employs the tree exploration game (Theorem 3.1), we can also establish the result in a more direct way using a technique known as the *no-counterexample interpretation*. We present a self-contained proof in Appendix B.2.

4 Warm-up: Krajícek's Technique and the Pich-Santhanam Result

In this section, we provide a detailed exposition of the unprovability result from Santhanam and Pich [PS21], which relies on a technique introduced by Krajícek [Kra11] and further investigated by Pich [Pic15a]. Their result (intuitively) means that strong average-case circuit lower bounds against co-nondeterministic circuits are not provable in T_{PV}. Concretely, for every $L \in \mathsf{NTIME}[2^{n^{o(1)}}]$, $\delta \in (0,1) \cap \mathbb{Q}$, and $n_0 \in \mathbb{N}$, T_{PV} cannot prove that:

> For every $n > n_0$ and every co-nondeterminisetic circuit $C : \{0,1\}^n \to \{0,1\}^1$ of size 2^{n^δ}, $C(x) = L(x)$ on at most $\frac{1}{2} + \frac{1}{2^{n^\delta}}$ fraction of $x \in \{0,1\}^n$.

Since our unprovability results are obtained by extending the original ideas of Pich and Santhanam [PS21] and Krajícek [Kra11] in combination with our new witnessing theorem, this section might be particularly helpful for a reader that is unfamiliar with these methods.

4.1 Formalization of complexity lower bounds

While the unprovability result of [PS21] is robust to some details of the formalization, we will make a few comments here about the way it is done. First, we can represent any natural number $a \in \mathbb{N}$ by an $\mathcal{L}(\mathsf{PV})$-term, e.g., $a = 1 + 1 + \ldots + 1$, where $+ \colon \mathbb{N}^2 \to \mathbb{N}$ is the $\mathcal{L}(\mathsf{PV})$ function symbol for addition, and 1 is a constant symbol in $\mathcal{L}(\mathsf{PV})$.[15] From this, we can introduce representations for other finite objects. For instance, a natural number can represent the code of a Turing machine M, while a pair of natural numbers can represent a rational number $\delta \in \mathbb{Q}$. In some cases, we will quantify over all such objects in the meta-language, e.g., if M is a Turing machine (in the usual sense), then we can consider a $\mathcal{L}(\mathsf{PV})$-sentence ϕ_M that refers to machine M via its representation as a natural number.

For a nondeterministic Turing machine M, a constant $n_0 \in \mathbb{N}$, and functions $s, m \colon \mathbb{N} \to \mathbb{N}$, we write $\mathsf{LB}(M, s, m, n_0)$ to denote an $\mathcal{L}(\mathsf{PV})$-sentence stating that, for every input length $n \geq n_0$ and for every co-nondeterminstic circuit $D_n(x, z)$ of size $\leq s(n)$, there are at least $m = m(n)$ distinct input strings $x^1, \ldots, x^m \in \{0,1\}^n$ such that $M(x^i) \neq D_n(x^i)$ for each $1 \leq i \leq m$.[16] A bit more formally, this

[15] Of course, one can consider more efficient encodings (e.g., dyadic notation), but this will not make a difference in our argument.

[16] Here and throughout the exposition, we use that $M(x) = 1$ if and only if there exists y such that $M(x, y) = 1$ (as M computes *nondeterministically*), while $D_n(x) = 1$ if and only if for every z we have $D_n(x, z) = 1$ (since D is a *co-nondeterministic* circuit).

sentence can be expressed in $\mathcal{L}(\mathsf{PV})$ in the following way, where we assume that M on input length n runs in time $\leq t(n)$ for some efficiently computable time bound $t(n) \leq N(n) = 2^n$ and that $s(n)$ and $m(n)$ are efficiently computable and bounded by $N = 2^n$:

$$\begin{aligned}\mathsf{LB}(M, s, m, n_0) \quad \triangleq \quad & \forall v\ \forall N = |v|\ \forall n = |N| \text{ such that } n \geq n_0 \quad (\text{in other words, } n \in \mathsf{LogLog}) \\ & \forall \text{ co-nondet. circuit } D_n \text{ of size } \leq s(n) \\ & \exists m = m(n) \text{ distinct } n\text{-bit strings } x^1, \ldots, x^m \text{ s.t. } \mathsf{Error}_{M,D_n}(x^i) \text{ for all } i \in [m],\end{aligned}$$

where we let $\mathsf{Error}_{M,D_n}(x)$ denote the following $\mathcal{L}(\mathsf{PV})$-formula:

$$\mathsf{Error}_{M,D_n}(x) \ \equiv \ \Big[\exists y\ \exists z\ M(x,y) = 1 \ \wedge \ D_n(x,z) = 0\Big] \ \vee \ \Big[\forall y'\ M(x,y') = 0 \ \wedge \ \forall z'\ D_n(x,z') = 1\Big],$$

with the length of y, y' and z, z' bounded by the running time of M and the size of D_n, respectively.

The definition above can be made formal by the use of explicit $\mathcal{L}(\mathsf{PV})$-function symbols that evaluate circuits and machines on a given input and that perform other necessary checks, e.g., deciding when a given object represents a circuit of size at most $s(n)$. All this can be done without increasing the quantifier complexity of the resulting sentence, since $n \in \mathsf{LogLog}$ and polynomial-time computations over $N = 2^n$-bit strings are feasible. For the same reason, the quantification over $i \in [m]$ does not increase quantifier complexity, using that $m(n) \leq N$. Indeed, in the sentence it is enough to existentially quantify over $m(n)$ strings x^i and over $m(n)$ strings y^i, z^i followed by a universal quantification over $m(n)$ strings y'^i, z'^i, and the remaining error conditions can be expressed using a single $\mathcal{L}(\mathsf{PV})$-function symbol that gets as input the encoding of each collection of strings (formally, each family of m strings is a single object, and the strings are decoded from it). Overall, we get that $\mathsf{LB}(M, s, m, n_0)$ is a $\forall\Sigma^b_2$-$\mathcal{L}(\mathsf{PV})$ sentence.

Theorem 4.1 ($\mathsf{T_{PV}}$ doesn't prove strong a.e. average-case co-nondeterministic lower bounds for NP)**.** *For every $n_0 \in \mathbb{N}$ and $\delta \in \mathbb{Q} \cap (0,1)$, if M is a nondeterministic machine whose running time is bounded by some constructive function $t(n) = 2^{n^{o(1)}}$, then*

$$\mathsf{T_{PV}} \nvdash \mathsf{LB}(M, s, m, n_0),$$

where $s(n) = 2^{n^\delta}$ and $m(n) = 2^n/2 - 2^n/2^{n^\delta}$.

In particular, for every language $L \in \mathsf{NP}$ and $\delta > 0$ it is consistent with $\mathsf{T_{PV}}$ that there are infinitely many input lengths n and a co-nondeterministic circuit D_n of size $\leq 2^{n^\delta}$ such that

$$\Pr_{x \sim \{0,1\}^n}[L(x) = D_n(x)] \geq 1/2 + 2^{-n^\delta}.$$

A strengthening of Theorem 4.1 is discussed in Section 4.3.

4.2 Proof of Theorem 4.1

Let n_0, δ, M, $t(n)$, $s(n)$, and $m(n)$ be as in the statement of Theorem 4.1. Arguing as in [PS21], we assume towards a contradiction that

$$\mathsf{T_{PV}} \vdash \mathsf{LB}(M, s, m, n_0).$$

Let $L \subseteq \{0,1\}^*$ be the language defined by M. We argue as follows.

(i) From the provability of this almost-everywhere average-case lower bound against co-nondeterministic circuits, it follows by the soundness of T_{PV} that (in the standard model) for every sequence $\{E_n\}_{n\geq 1}$ of *deterministic* circuits E_n of size $\leq 2^{n^\delta}$, if $n \geq n_0$ then

$$\Pr_{x\sim\{0,1\}^n}[L(x) = E_n(x)] \leq 1/2 + 2^{-n^\delta}.$$

(ii) From the provability of the sentence $\mathsf{LB}(M, s, m, n_0)$ it trivially follows that T_{PV} proves a sentence $\mathsf{LB}_{\mathsf{wst}}(M, s, n_0)$ which states a *worst-case* lower bound for M against co-nondeterministic circuits of the same size. We then show that the provability of $\mathsf{LB}_{\mathsf{wst}}(M, s, n_0)$ in T_{PV} implies that, in the standard model, for every fixed $k \geq 1$ and for every large enough n, there is a deterministic circuit A defined over n^k input variables and of size $2^{O(n)}$ such that

$$\Pr_{w\sim\{0,1\}^{n^k}}[L(w) = A(w)] \geq 1/2 + 2^{-O(n)}.$$

Taking $k > 1/\delta$ contradicts Item (i) above.

Note that the only remaining step is to show that:

> ($\star$) The provability of a worst-case lower bound against *co-nondeterministic* circuits allows us to non-trivially approximate L using *deterministic* circuits of bounded size.

Before proceeding with the proof of this result, we describe the aforementioned worst-case lower bound sentence in a convenient way.

$$\begin{aligned}\mathsf{LB}_{\mathsf{wst}}(M, s, n_0) \quad \equiv \quad & \forall n \in \mathsf{LogLog} \text{ with } n \geq n_0,\ \forall \text{ co-nondet. circuit } D \text{ of size } \leq s(n)\\ & \exists x \in \{0,1\}^n\ \exists y \in \{0,1\}^{t(n)}\ \exists z \in \{0,1\}^{s(n)} \text{ such that } \mathsf{Error}(x, y, z),\end{aligned}$$

where here $\mathsf{Error}(x, y, z)$ denotes the following $\mathcal{L}(\mathsf{PV})$-formula:

$$\mathsf{Error}(x, y, z) \;\equiv\; \Big[M(x, y) = 1 \,\wedge\, D(x, z) = 0\Big] \;\vee\; \Big[\forall y'\, M(x, y') = 0 \,\wedge\, \forall z'\, D(x, z') = 1\Big], \tag{2}$$

where the lengths of y' and z' are bounded as before. Observe that $\mathsf{LB}_{\mathsf{wst}}(M, s, n_0)$ is also a $\forall\Sigma^b_2$-$\mathcal{L}(\mathsf{PV})$ sentence.

It is easy to see that, under any reasonable formalization, if $m(n) \geq 1$ then T_{PV} derives the worst-case lower bound sentence $\mathsf{LB}_{\mathsf{wst}}(M, s, n_0)$ from the average-case lower bound sentence $\mathsf{LB}(M, s, m, n_0)$. Consequently, it is sufficient for us to prove the following lemma, which formalizes statement ($\star$).

Lemma 4.2 (Non-trivial correlation from the provability of a worst-case lower bound)**.** *Let $n_0 \in \mathbb{N}$, $\delta \in \mathbb{Q} \cap (0, 1)$, M be a nondeterministic machine whose running time is bounded by some constructive function $t(n) = 2^{n^{o(1)}}$, and $s(n) = 2^{n^\delta}$. If*

$$\mathsf{T}_{\mathsf{PV}} \vdash \mathsf{LB}_{\mathsf{wst}}(M, s, n_0),$$

then for every $k \geq 1$ and sufficiently large n, there is a deterministic circuit $B : \{0,1\}^{n^k} \to \{0,1\}$ of size $2^{O(n)}$ such that

$$\Pr_{w\sim\{0,1\}^{n^k}}[L(w) = B(w)] \geq 1/2 + 2^{-O(n)},$$

where L is the language decided by M.

Lemma 4.2 and its proof have appeared in [Kra11, Pic15a]. We provide next a detailed exposition of this technique.

4.2.1 A simpler case: $\ell = 1$ in the KPT student-teacher protocol

Note that we can apply Theorem 2.9 to sentence $\mathsf{LB}_{\mathsf{wst}}$ and theory T_{PV}, since T_{PV} is a universal theory and it is not difficult to see that $\mathsf{LB}_{\mathsf{wst}}$ can be written in the form required by Theorem 2.9. Since $\mathcal{L}(\mathsf{PV})$-terms correspond to polynomial-time computable functions, the corresponding student computes in time polynomial in the length of its input. Using that $n \in \mathsf{LogLog}$ in sentence $\mathsf{LB}_{\mathsf{wst}}$, we obtain uniform algorithms $f_1, \ldots, f_\ell$ that compute in time $2^{O(n)}$ and satisfy the conclusion of Theorem 2.9. Note that we cannot control the constant ℓ. In this section, we discuss the simpler case when we get $\ell = 1$ in the application of Theorem 2.9 to $\mathsf{T}_{\mathsf{PV}} \vdash \mathsf{LB}_{\mathsf{wst}}(M, s, n_0)$.

Omitting auxiliary variables in the input to f_1 and highlighting the relevant parameters,[17] $f_1(n, D)$ receives n and an *arbitrary* co-nondeterministic circuit D of size $\leq s(n) = 2^{n^\delta}$, and outputs a triple (x, y, z) such that $\mathsf{Error}(x, y, z)$ holds. Assuming that $n \geq n_0$ and $\ell = 1$ in the KPT Witnessing, it follows that this triple witnesses that $D(x) \neq M(x)$ over the standard model $\mathbb{N}$. In other words, $x \in \{0,1\}^n$, $y \in \{0,1\}^{t(n)}$ for $t(n) = 2^{n^{o(1)}}$, $z \in \{0,1\}^{s(n)}$, and the following holds:

$$\Big[M(x,y) = 1 \wedge D(x,z) = 0\Big] \vee \Big[\forall y'\, M(x,y') = 0 \wedge \forall z'\, D(x,z') = 1\Big].$$

To prove Lemma 4.2 when $\ell = 1$, let $k \geq 1$, and assume that n is sufficiently large. We will use f_1 to construct a deterministic circuit B defined over n^k input variables and of size $2^{O(n)}$ such that

$$\Pr_{w \sim \{0,1\}^{n^k}}[L(w) = B(w)] \geq 1/2 + 2^{-O(n)},$$

where L is the language computed by the nondeterministic machine M. As a key point, note that we can invoke $f_1(n, D)$ on any co-nondeterministic circuit $D(x, \cdot)$ over n input variables and of size $\leq 2^{n^\delta}$. In order to construct the deterministic circuit B, we will also use that $f_1(n, D) = (x, y, z)$ computes in time $2^{O(n)}$ and therefore can be simulated by a circuit of size $2^{O(n)}$. For the $\ell = 1$ case, we will not need the output strings y and z during the construction of A. From now on, we simplify notation and write $f_1(D) = x$ to denote the relevant input and output of f_1 for this case.

Let $C_{n^k}(w, z)$ be a Boolean circuit that computes as M on inputs of length n^k, where z corresponds to the nondeterministic input. Since M runs in time at most $2^{n^{o(1)}}$, C_{n^k} has size at most 2^{n^δ} when n is sufficiently large. We partition its first input as $w = x \| w'$, where $|x| = n$ and $|w'| = n^k - n$. Now for a fixed string $w' \in \{0,1\}^{n^k - n}$, consider the circuit $D_{w'}(x, z) \triangleq \neg C_{n^k}(x \| w', z)$. Viewing $D_{w'}$ as a co-nondeterministic circuit, we get that

$$D_{w'}(x) = 1 \iff \forall z\, D_{w'}(x, z) = 1 \iff \forall z\, C_{n^k}(x \| w', z) = 0 \iff x \| w' \notin L(M). \tag{3}$$

Intuitively, if we "learn" the output bit of $D_{w'}(x)$ for some pair (w', x), we also "learn" if the input string $x \| w'$ is in $L(M)$. As a consequence, the collection $\{D_{w'}(x)\}_{w'}$ of co-nondeterministic circuits $D_{w'}$ (defined over n input bits) captures the computation of L on inputs of length n^k.

As each $D_{w'}$ has size at most 2^{n^δ}, we can invoke f_1 on them. Since $f_1(D_{w'})$ finds mistakes with respect to the nondeterministic computation of M, we know that $D_{w'}(x^*) \neq M(x^*)$ for $x^* \triangleq f_1(D_{w'})$. Since there are only 2^n possibilities for the output of f_1, the following holds.

[17] The condition $n \in \mathsf{LogLog}$ means that n is the length of N, while N is the length of some universally quantified variable v. For this reason, formally, v is an input to f_1, and not n. However, over $\mathbb{N}$ we will run f_1 on a fixed input v, such as 1^N, where $N = 2^n$. For this reason, n is the parameter that controls the length of the remaining inputs.

Fact 4.3. *There is a string $x^* \in \{0,1\}^n$ such that*

$$\Pr_{w' \in \{0,1\}^{n^k - n}} [f_1(D_{w'}) = x^*] \geq 2^{-n}.$$

Following our informal discussion from above, from the knowledge that $f_1(D_{w'}) = x^*$ (and of a bit encoding if $x^* \in L(M)$) we "learn" how to compute $L(M)$ on the input $w = x^* \| w'$. Since this will happen with non-trivial probability over a random choice of w' and x^*, this can be used to non-trivially approximate $L(M)$ over input length n^k.

Formally, let $b^* \in \{0,1\}$ be 1 if and only if $M(x^*) = 1$, i.e., if $x^* \in L(M)$. The string x^* and the bit b^* will be stored as non-uniform advice in the deterministic circuit B that we show to be correlated with $L(M)$ on input length n^k. First, consider the following randomized circuit B':

```
   Input  : The input $w \in \{0,1\}^{n^k}$ and a random $r \in \{0,1\}$
   Advice: $x^* \in \{0,1\}^n$ and $b^* \in \{0,1\}$
 1 Let $w = x \| w'$ and compute $f_1(D_{w'})$;
   // Note that we can construct a description of $D_{w'}$ from $w$ and $C_{n^k}$.
 2 If $f_1(D_{w'}) \neq x^*$, return $r$;
 3 If $x \neq x^*$, return $r$;
 4 Otherwise, return $b^*$;
```

Algorithm 1: Randomized Circuit B' for $L(M)$

Note that B' can be computed with $2^{O(n)}$ gates, since f_1 runs in time $2^{O(n)}$. Next, we show that the randomized circuit B' non-trivially correlates with $L(M)$ on inputs of length n^k. After that, fixing the random bit b in B' yields the desired deterministic circuit A.

Fact 4.4. *If B' reaches Line 4 on an input string w, then $B'(w, b) = M(w)$, i.e., B' correctly decides $L(M)$ on input w.*

Proof. Under the assumption that B' reaches Line 4 on an input string w, it follows that $w = x^* \| w'$ and $f_1(D_{w'}) = x^*$. Moreover, observe that the random bit b does not affect the output of B' in this case. We have

$$\begin{aligned} B'(w,r) = 1 &\iff b^* = 1 && \\ &\iff M(x^*) = 1 && \text{(by the definition of } b^*\text{)} \\ &\iff D_{w'}(x^*) = 0 && \text{(using that } f_1 \text{ finds a mistake)} \\ &\iff w = x^* \| w' \in L(M) && \text{(by Equation 3)} \\ &\iff M(w) = 1. && \text{(since } M \text{ computes } L(M)\text{)} \end{aligned}$$

In other words, $B'(w,r) = M(w)$. □

In addition, by Fact 4.3,

$$p \triangleq \Pr_{r,w}[B' \text{ reaches Line 4}] = \Pr_{x,w'}[x = x^* \wedge f_1(D_{w'}) = x^*] \geq 2^{-n} \cdot 2^{-n}.$$

On the other hand, when B' does not reach Line 4 it outputs a random bit that is independent of the input string w. Therefore, using Fact 4.4 and the lower bound on p,

$$\Pr_{r,w}[B'(w,r) = M(w)] \geq p \cdot 1 + (1-p) \cdot 1/2 = 1/2 + p/2 \geq 1/2 + 2^{-2n+1} = 1/2 + 2^{-O(n)}.$$

Fixing the random bit r in the best way maintains this advantage and completes the proof of Lemma 4.2 when $\ell = 1$.

Remark 4.5. The same argument can be used to approximate *any* nondeterministic circuit of size $2^{n^{k\delta}}$ defined over n^k bits by a deterministic circuit of size $2^{O(n)}$, instead of just for $L(M) \cap \{0,1\}^{n^k}$. In other words, by connecting $D_{w'}$ to the computation of the appropriate co-nondeterminisetic circuit, "learning" output bits of $D_{w'}$ via f_1 translates into a non-trivial approximation *(*using exactly the same strategy*)*. This will also hold when analysing the case $\ell > 1$. In particular, from $\mathsf{T}_{\mathsf{PV}} \vdash \mathsf{LB}_{\mathsf{wst}}(M, s, n_0)$ we are able to non-trivially approximate any language in $\mathsf{NSIZE}[2^{n^{o(1)}}]$ and not just $L(M)$.

4.2.2 The case $\ell = 2$ via the Nisan-Wigderson generator

In this section, we consider the case where the disjunction obtained from KPT Witnessing (Theorem 2.9) has size $\ell = 2$. This essentially covers all difficulties in the general case. Before handling $\ell = 2$, it is instructive to highlight some key points of the proof when $\ell = 1$:

(i) We implicitly relied on the ability of *certifying* when an input x is a mistake. More precisely, when $\ell = 1$, if $f_1(D_{w'}) = (x, y, z)$, we have the *guarantee* that $D_{w'}(x) \neq M(x)$. This is because there is a *single* round in the corresponding Student-Teacher protocol.

(ii) By an averaging argument, we fixed a good string $x^* \in \{0,1\}^n$ (Fact 4.3), which eventually allowed us to compute $M(x^* w')$ on a non-trivial fraction of w', by storing x^* and the corresponding bit $b^* = M(x^*)$.

(iii) This was accomplished by considering a family $\{D_{w'}(x)\}_{w'}$ of co-nondeterministic circuits over n-bit inputs that compute according to a circuit defined over input length n^k that is related to the language we would like to approximate.

(iv) On an input $w \in \{0,1\}^{n^k}$ with $w = x \| w'$ for which the witnessing provided by $f_1(D_{w'})$ was inconsistent with the actual input part x (we can easily detect this), we output a random bit.

Note that this approach no longer works when $\ell > 1$: the first term obtained from KPT Witnessing might not succeed in finding a mistake. For this reason, we cannot assume in Item (i) that if $f_1(D_{w'}) = (x, y, z)$ then $D_{w'}(x) \neq M(x)$.

Let f_1 be the first term in the KPT disjunction when $\ell > 1$. Note that we can still fix a popular *candidate* mistake $x^* \in \{0,1\}^n$, as in Fact 4.3. Recall that $f_1(D_{w'}) = (x_{w'}, y_{w'}, z_{w'})$ (we did not have to use $y_{w'}$ and $z_{w'}$ in the argument for $\ell = 1$). We can check whether $x_{w'} = x^*$, as before, and we would like to use $y_{w'}$ and $z_{w'}$ together with some hard-coded information to decide if $x^* = x_{w'}$ is indeed a mistake for $D_{w'}$. While both $y_{w'}, z_{w'} \in \{0,1\}^{\leq 2^n}$, there are $2^{\Omega(n^k)}$ possible strings w'. Unfortunately, it is unclear how to store enough information in the non-uniform circuit B' to certify that a mistake has been found by f_1 while maintaining a circuit size bound of $2^{O(n)}$.

To reduce the amount of advice needed in B' and address this difficulty, the solution [Kra11, Pic15a] is to employ a more sophisticated family $\{D_{w'}\}_{w'}$ of circuits constructed via the Nisan-Wigderson generator [NW94].

For a nondeterministic machine M that decides a language $L(M)$, we use the notation $\{\mathsf{NW}_{\overline{L(M)}}(w)\}_w$ to denote the collection of functions obtained from the Nisan-Wigderson generator when instantiated with the Boolean function h that corresponds to the negation of $L(M)$ over inputs of length $n^{c/2}$.

Fact 4.6. *Let M be a nondeterministic machine that runs in time $2^{m^{o(1)}}$ on inputs of length m. For any* *constant $c \geq 1$ and every large enough n, each function in $\{\mathsf{NW}_{\overline{L(M)}}(w)\}_w$ can be computed by a co-nondeterministic circuit $D_w(x)$ of size at most 2^{n^δ}.*

The case $\ell = 1$ via the NW generator. Before handling the case $\ell = 2$, we sketch the proof of the case $\ell = 1$ using the collection $\{D_w\}_{w \in \{0,1\}^{n^c}}$ obtained from the nondeterministic machine M and the NW generator, with parameters as above.

Consider the function $f_1(D_w) = (x, y, z)$ obtained by applying Theorem 2.9, and assume that $\ell = 1$. Again, we will not inspect y and z when $\ell = 1$. Recall that $f_1(D_w)$ computes in time $2^{O(n)}$. We show how to decide $L(M)$ on inputs of length $n^{c/2}$ by a deterministic circuit of size $2^{O(n)}$ that agrees with $L(M)$ with probability $\geq 1/2 + 2^{-O(n)}$ over a uniformly random input string.

Similarly to Fact 4.3, by a standard averaging argument we can establish the following fact.

Fact 4.7. *There is a string $x^* \in \{0,1\}^n$ such that*

$$\Pr_{w \in \{0,1\}^{n^c}}[f_1(D_w) = x^*] \geq 2^{-n}.$$

Recall that J_{x^*} denotes the subset of $[n^c]$ of size $n^{c/2}$ corresponding to the x^*-row of the design in our NW generator; for $a \in \{0,1\}^{n^c - n^{c/2}}$ and $u \in \{0,1\}^{n^c}$, $r_x(a, u)$ denotes the "concatenated" string $a \cup u$ obtained by viewing $a \in \{0,1\}^{[n^c] \setminus J_x}$ and $u \in \{0,1\}^{J_x}$. By another averaging argument, we get the following consequence.

Fact 4.8. *There is a string $a \in \{0,1\}^{[n^c] \setminus J_{x^*}}$ of length $n^c - n^{c/2}$ such that*

$$\Pr_{\substack{u \sim \{0,1\}^{J_{x^*}} \\ w \triangleq u \cup a}}[f_1(D_w) = x^*] \geq 2^{-n}.$$

We can view $D_w = \mathsf{NW}_{\overline{L(M)}}(w)$ as a co-nondeterministic circuit for computing $\overline{L(M)}$ over inputs of length $n^{c/2}$ derived from the seed w:

$$D_w(x) = 1 \quad \Longleftrightarrow \quad w|_{J_x} \in \overline{L(M)}.$$

Given the previous discussion, we are interested in seeds $w \in \{0,1\}^{n^{2c}}$ of the form $w = a \cup u$, where $a \in \{0,1\}^{[n^c] \setminus J_{x^*}}$ is fixed, $u \in \{0,1\}^{J_{x^*}}$, and $f_1(D_w) = x^*$. We know that a non-trivial fraction of strings u will satisfy this condition. Since f_1 witnesses mistakes with respect to $L(M)$ over inputs of length n (note that D_w is a conondeterministic circuit over n-bit inputs), whenever $f_1(D_w) = x^*$ we are guaranteed that

$$D_w(x^*) = 1 \quad \Longleftrightarrow \quad M(x^*) = 0,$$

which implies that $M(x^*) = 0$ if and only if $w|_{J_{x^*}} \notin L(M)$. Now x^* is fixed, so the equality $M(x^*) = 0$ does not depend on other conditions. For instance, if $M(x^*) = 0$, we can conclude that on any input string $u \in \{0,1\}^{n^{c/2}}$, if for $w = a \cup u$ we have $f_1(D_w) = x^*$, then $u = w|_{J_{x^*}}$ is not in $L(M)$. Consequently, this allows us to correctly compute $L(M)$ on any such input $u \sim \{0,1\}^{n^{c/2}}$, which constitute a non-trivial fraction of inputs. Moreover, we can check whether an input u satisfies $f_1(D_w) = x^*$ using a deterministic circuit of size $2^{O(n)}$.

Formally, consider the fixed strings $x^* \in \{0,1\}^n$ and $a \in \{0,1\}^{[n^c] \setminus J_{x^*}}$ from above, and let $b^* \triangleq M(x^*) \in \{0,1\}$. We hardcode x^*, a, and b^* in the randomised circuit $B(u)$ described below:

```
Input : The input $u \in \{0,1\}^{n^{c/2}}$ and a random $r \in \{0,1\}$
Advice: $x^* \in \{0,1\}^n$ and $b^* \in \{0,1\}$
Let $w = r_{x^*}(a, u)$;
Let $x = f_1(D_w)$;
If $x \neq x^*$, output the random bit $r$;
Otherwise, return $b^*$;
```

Algorithm 2: Randomised Circuit B for $L(M)$

Given the aforementioned discussion, it is easy to see that

$$\Pr_{u,r}[B(u,r) = M(u)] \geq p \cdot 1 + (1-p) \cdot 1/2 = 1/2 + p/2 \geq 1/2 + 2^{-n+1},$$

where p is the probability in the LHS of Fact 4.8. Consequently, by an averaging argument over the random bit r, there is a deterministic circuit of size $2^{O(n)}$ that computes $L(M)$ on inputs of length $n^{c/2}$ with the same advantage.

The case $\ell = 2$ via the NW generator. Recall that

$$\mathsf{Error}(x,y,z) \;\equiv\; \Big[M(x,y) = 1 \,\wedge\, D(x,z) = 0\Big] \;\vee\; \Big[\forall y'\; M(x,y') = 0 \,\wedge\, \forall z'\; D(x,z') = 1\Big].$$

We now have a function $f_1(D) = (x,y,z)$ that attempts to produce a triple (x,y,z) satisfying $\mathsf{Error}(x,y,z)$, and a function $f_2(D,y',z')$ which given a pair y', z' for which

$$\Big[M(x,y) = 0 \,\vee\, D(x,z) = 1\Big] \;\wedge\; \Big[M(x,y') = 1 \,\vee\, D(x,z') = 0\Big] \tag{4}$$

is able to produce an input x' such that $D(x') \neq M(x')$.

Again, we consider the family $\{D_w\}_{w \in \{0,1\}^{n^c}}$ of conondeterministic circuits D_w of size $\leq 2^{n^\delta}$ that compute $\mathsf{NW}_{\overline{L(M)}}(w)\colon \{0,1\}^n \to \{0,1\}$ for a fixed seed w, with parameters as described above. In particular, this generator is instantiated with respect to the Boolean function h corresponding to $\overline{L(M)}$ over inputs of length $n^{c/2}$, for a fixed but arbitrarily large constant $c \geq 1$.

By an averaging argument, the following claim holds.

Fact 4.9. *There is a string $x_1 \in \{0,1\}^n$ such that*

$$\Pr_{w \in \{0,1\}^{n^c}}[f_1(D_w) = x_1] \geq 2^{-n}.$$

Fix this x_1. We define the sets $S^{\mathsf{mist}}_{x_1} \subseteq S_{x_1} \subseteq \{0,1\}^{n^c}$ as follows:

$$\begin{aligned} S_{x_1} &\triangleq \Big\{w \in \{0,1\}^{n^c} \mid f_1(D_w) = x_1\Big\}, \\ S^{\mathsf{mist}}_{x_1} &\triangleq \Big\{w \in S_{x_1} \mid D_w(x_1) \neq M(x_1)\Big\}, \end{aligned}$$

and consider the density of $S^{\mathsf{mist}}_{x_1}$ with respect to its superset S_{x_1}.

Case 1. $|S^{\mathsf{mist}}_{x_1}| > (2/3) \cdot |S_{x_1}|$. We can essentially proceed as in the case of $\ell = 1$, with the exception that one needs to be careful when invoking an analogue of Fact 4.8. This is because fixing a string $a \in \{0,1\}^{[n^c] \setminus J_{x_1}}$ might keep the density of S_{x_1} at least 2^{-n} but could significantly decrease the relative density of the set $S^{\mathsf{mist}}_{x_1}$ after the restriction.

To handle this, we introduce the following notation. For $m \geq 1$, a set $S \subseteq \{0,1\}^{[m]}$, and a string $a \in \{0,1\}^I$, where $I \subseteq [m]$, we define the *restriction of S with respect to a* as the set

$$S\restriction_a \triangleq \{w \in S \mid w|_I = a\}.$$

Under the assumption that $|S^{\mathsf{mist}}_{x_1}| > (2/3) \cdot |S_{x_1}|$, it is possible to show by a counting argument (see, e.g., Lemma E.1) that there exists a string $a \in \{0,1\}^{[n^c] \setminus J_{x_1}}$ such that

$$p \triangleq \frac{|S_{x_1}\restriction_a|}{2^{n^{c/2}}} \geq \frac{1}{n} \cdot 2^{-n} \quad \text{and} \quad \frac{|S^{\mathsf{mist}}_{x_1}\restriction_a|}{|S_{x_1}\restriction_a|} \geq \frac{2}{3} - \frac{1}{n}. \tag{5}$$

While it is not clear how to decide in size $2^{O(n)}$ if a string $w \in S^{\mathsf{mist}}_{x_1}\restriction_a$, we can check whether $w \in S_{x_1}\restriction_a$. Since $S^{\mathsf{mist}}_{x_1}$ is dense in $S_{x_1}\restriction_a$, this is enough to adapt the original strategy used for $\ell = 1$.

Formally, fix strings $x_1 \in \{0,1\}^n$ and $a \in \{0,1\}^{[n^c] \setminus J_{x_1}}$ as above, and let $b_1 \triangleq M(x_1) \in \{0,1\}$. We hardcode x_1, a, and b_1 in the randomised circuit $B_1(u,r)$ described below.

```
Input  : The input $u \in \{0,1\}^{n^{c/2}}$ and a random $r \in \{0,1\}$
Advice: $x_1 \in \{0,1\}^n$, $a \in \{0,1\}^{n^c - n^{c/2}}$, and $b_1 \in \{0,1\}$
Let $w = r_{x_1}(a,u)$;
Let $x = f_1(D_w)$;
If $x \neq x_1$, output the random bit $r$;
Otherwise, output the fixed bit $b_1 = M(x_1)$;
```

Algorithm 3: Randomized Circuit B_1 for $L(M)$ when $|S^{\mathsf{mist}}_{x_1}| > (2/3) \cdot |S_{x_1}|$.

Clearly, B_1 is computed by a randomised circuit of size $2^{O(n)}$. To analyse its success probability, first note that if u is such that $w = r_{x_1}(a,u) \notin S_{x_1}\restriction_a$, then $B_1(u) = M(u)$ with probability $1/2$. On the other hand, for those u such that $w = r_{x_1}(a,u) \in S_{x_1}\restriction_a$, at least a $2/3 - 1/n$ fraction of them are in $S^{\mathsf{mist}}_{x_1}\restriction_a$, in which case $B_1(u)$ is correct. Since $S_{x_1}\restriction_a$ has density at least $1/n \cdot 2^{-n}$, it follows that

$$\Pr_{u,r}[B_1(u,r) = M(u)] = (1-p) \cdot \frac{1}{2} + p \cdot \left(\frac{2}{3} - \frac{1}{n}\right) = \frac{1}{2} + p \cdot \left(\frac{1}{6} - \frac{1}{n}\right) = \frac{1}{2} + \Omega\left(\frac{2^{-n}}{n}\right),$$

which is $1/2 + 2^{-O(n)}$. Fixing the random bit r in the best way yields the desired deterministic circuit.

Case 2. $|S^{\mathsf{mist}}_{x_1}| < (2/3) \cdot |S_{x_1}|$. In this case, the mistakes of at least a $1/3$ fraction of the circuits D_w for $w \in S_{x_1}$ must be witnessed by f_2. To make sure the output of $f_2(D_w, y', z')$ is indeed a string x_2 for which $M(x_2) \neq D_w(x_2)$, we must provide a pair y', z' such that

$$\Big[M(x_1, y_1) = 0 \vee D_w(x_1, z_1) = 1\Big] \wedge \Big[M(x_1, y') = 1 \vee D_w(x_1, z') = 0\Big], \tag{6}$$

where $f_1(D_w) = (x_1, y_1, z_1)$. We consider the Teacher that to each $w \in S_{x_1} \setminus S^{\mathsf{mist}}_{x_1}$ and corresponding (y_1, z_1) assign the lexicographic first pair (y'_w, z'_w) for which Equation (6) holds. Note that such a pair always exists, since in this case for $x_1 = f_1(D_w)$ we have $M(x_1) = D_w(x_1)$.

By an averaging argument, the following claim holds.

Fact 4.10. *Under this fixed Teacher, there is a string $x_2 \in \{0,1\}^n$ such that the set*

$$S_{x_1,x_2} \triangleq \{w \in S_{x_1} \mid D_w(x_1) = M(x_1) \wedge f_2(D_w, y'_w, z'_w) = x_2\}$$

has density at least $(1/3) \cdot 2^{-2n}$ in $\{0,1\}^{n^c}$.

Note that, by construction, if $w \in S_{x_1,x_2}$ then for $x_2 = f_2(D_w, y'_w, z'_w)$ we have $D_w(x_2) \neq M(x_2)$. Note that $x_1 \neq x_2$ because otherwise we have $S_{x_1,x_2} = \varnothing$.[18] Moreover, the set S_{x_1,x_2} has enough density for our purposes. However, for this to be useful we must verify that a given circuit D_w satisfies $w \in S_{x_1,x_2}$ using a deterministic circuit of size $2^{O(n)}$.

By another averaging argument, we have the following result.

Fact 4.11. *There is a string $a \in \{0,1\}^{[n^c]\setminus J_{x_2}}$ such that*

$$\frac{|S_{x_1,x_2}\!\restriction_a|}{2^{n^{c/2}}} \;\geq\; \frac{1}{3} \cdot 2^{-2n}.$$

Fix this string $a \in \{0,1\}^{[n^c]\setminus J_{x_2}}$ together with the strings x_1 and x_2. We will assume that the following computation is possible in order to complete the proof, returning to it later on:

(∇) There is a deterministic circuit $E(w)$ of size $2^{O(n)}$ as follows: Given a $w \in S_{x_1}$ of the form $a \cup u$ such that $D_w(x_1) = M(x_1)$, it outputs the lexicographic first pair (y'_w, z'_w) for which Equation (6) holds, where $(x_1, y_1, z_1) = f_1(D_w)$.[19]

Consider strings $x_1, x_2 \in \{0,1\}^n$ and $a \in \{0,1\}^{[n^c]\setminus J_{x_2}}$ as above, and let $b_2 \triangleq M(x_2) \in \{0,1\}$. We hardcode this information in the randomised circuit $B_2(u)$ described below, which includes the circuit $E(w)$ from (∇) as a subroutine:

```
Input  : The input $u \in \{0,1\}^{n^{c/2}}$ and a random $r \in \{0,1\}$
Advice: $x_1, x_2 \in \{0,1\}^n$, $a \in \{0,1\}^{n^c - n^{c/2}}$, and $b_2 \in \{0,1\}$
Let $w = r_{x_2}(a, u)$;
Let $(x, y_1, z_1) = f_1(D_w)$;
If $x \neq x_1$, output the random bit $r$;
Let $(y'_w, z'_w) = E(w)$;
If the tuple $(x_1, y_1, z_1, y'_w, z'_w)$ satisfies Equation (6) and $f_2(D_w, y'_w, z'_w) = x_2$, output $b_2$;
Otherwise output the random bit $r$.
```

Algorithm 4: Randomized Circuit B_2 for $L(M)$ when $|S^{\mathsf{mist}}_{x_1}| \leq (2/3) \cdot |S_{x_1}|$.

Note that, under assumption (∇), B_2 can be computed by a randomised circuit of size $2^{O(n)}$. Moreover, it follows from our discussion and from the density of $S_{x_1,x_2}\!\restriction_a$ that

$$\Pr_{u,r}[B_2(u,r) = M(u)] \;\geq\; \frac{1}{2} + \Omega\big(2^{-2n}\big)\,.$$

[18] Assume it is not the case, there is a $w \in S_{x_1,x_2}$ such that $D_w(x_2) \neq M(x_2)$. However, we know that $D_w(x_1) = M(x_1)$ by the definition of S_{x_1,x_2}, which is impossible when $x_1 = x_2$.

[19] Note that in this case $f_2(D_w, y'_w, z'_w)$ outputs a mistake of D_w, since $\mathsf{Error}(x_1, y_1, z_1)$ does not hold and correct witnesses for this are provided.

This yields a deterministic circuit with the same advantage.[20]

Proof of (∇). We will now use the main property of the combinatorial design behind the Nisan-Wigderson generator: the sets J_{x_1} and J_{x_2} overlap in at most n coordinates. This will allow us to hardcode all relevant pairs (y'_w, z'_w) using circuit size $2^{O(n)}$.

To implement (∇), we are given a string $w = a \cup u$, where $a \in \{0,1\}^{[n^c]\setminus J_{x_2}}$ is fixed and $u \in \{0,1\}^{J_{x_2}}$, such that the following conditions hold:

- Let $(x, y_1, z_1) = f_1(D_w)$, then $x = x_1$.
- $D_w(x_1) = M(x_1)$.

Our goal is to output the lexicographic first pair (y'_w, z'_w) such that:

$$\Big[M(x_1, y_1) = 0 \vee D_w(x_1, z_1) = 1\Big] \wedge \Big[M(x_1, y') = 1 \vee D_w(x_1, z') = 0\Big].$$

Note that such pair must exist since we assume that $D_w(x_1) = M(x_1)$.

Recall that $D_w(x_1) = \mathsf{NW}_{\overline{L(M)}}(w, x_1)$. The crucial observation that leads to the use of NW generator is that the desired pair (y'_w, z'_w) only depends on $w|_{J_{x_1}}$, which contains at most n bits of the input $u \in \{0,1\}^{n^{c/2}}$. This is because $w = a \cup u$ is a concatenation of a fixed $a \in \{0,1\}^{[n^c]\setminus J_{x_2}}$ and u viewed as $u \in \{0,1\}^{J_{x_2}}$, which means that

$$w|_{J_{x_1}} = (a \cup u)|_{J_{x_1}} = a|_{J_{x_1}} \cup u|_{J_{x_1}},$$

where $a|_{J_{x_1}}$ is fixed and $u|_{J_{x_1}}$ only consists of the indices within $J_{x_1} \cap J_{x_2}$ of size at most n.

As $E(w)$ depends on at most n bits of the input $u \in \{0,1\}^{n^c}$, we can implement it as a circuit that store all the answers for all 2^n possibilities, which requires at most $\mathsf{poly}(2^n) = 2^{O(n)}$ gates. Concretely, the circuit works as follows: Given $w \in \{0,1\}^{n^c}$, we firstly obtain $u \in \{0,1\}^{J_{x_2}}$ such that $w = a \cup u$; let $u' = u|_{J_{x_1}}$ be of length at most n, we look up the table to find the answer corresponding to u'. □

Remark 4.12. As in Remark 4.5, we note that the argument can be easily adapted to approximate any Boolean function g defined over n^k bits computable by a nondeterministic circuit of size $2^{n^{k\delta}}$ using a deterministic circuit of size $2^{O(n)}$, instead of for just $L(M) \cap \{0,1\}^{n^k}$. The provability of a circuit lower bound for a single language $L(M)$ provides non-trivial circuits for any such g.

Based on this, we can also prove that under the same assumption (i.e., the provability of worst-case circuit lower bound in $\mathsf{T}^i_{\mathsf{PV}}$), for every constant $\epsilon \in (0,1)$, $s = s(m) = 2^{m^{o(1)}}$, and sufficiently large m, any Boolean function $g : \{0,1\}^m \to \{0,1\}$ that can be computable by a nondeterminisetic circuit of size s can also be approximated by a co-nondeterministic circuit D of size 2^{m^ϵ}, that is:

$$\Pr_{x\sim\{0,1\}^m}\Big[C(x) = D(x)\Big] \geq \frac{1}{2} + \frac{1}{2^{m^\epsilon}}.$$

This can be done by setting $k = \lceil 20/\epsilon \rceil$, padding dammy inputs to $g : \{0,1\}^m \to \{0,1\}$ to obtain $g' : \{0,1\}^{m'} \to \{0,1\}$, where $m' = \lceil m^{1/k} \rceil^k \leq 2m$ for sufficiently large m, and applying the observation above to g' with

[20] Note that we cannot really guarantee that $D_w(x_1) = M(x_1)$ when invoking (∇), since this cannot be easily decided in deterministic size $2^{O(n)}$. This means that more inputs u than those leading to strings $w \in S_{x_1,x_2}\restriction_a$ might reach Line 5 and be assigned output value b_2. Nevertheless, B_2 will be correct on any such input u, by virtue of the two checks performed in Line 5. Put another way, the argument "covers" the inputs u leading to strings $w \in S_{x_1,x_2}\restriction_a$.

$n = \lceil m^{1/k} \rceil$.

4.2.3 Sketch of the general case

We now sketch how the argument presented in Section 4.2.2 can be generalised to the case that the Student-Teacher protocol runs for $\ell \geq 3$ rounds. Recall that sets $S_{x_1}, S_{x_1}^{\mathsf{mist}}, S_{x_1,x_2}$ in Section 4.2.2 are defined as

$$\begin{aligned} S_{x_1} &\triangleq \left\{ w \in \{0,1\}^{n^c} \mid f_1(D_w) = x_1 \right\} \\ S_{x_1}^{\mathsf{mist}} &\triangleq \left\{ w \in S_{x_1} \mid D_w(x_1) \neq M(x_1) \right\} \\ S_{x_1,x_2} &\triangleq \{ w \in S_{x_1} \setminus S_{x_1}^{\mathsf{mist}} \mid f_2(D_w, y'_w, z'_w) = x_2 \} \end{aligned}$$

In the general case, we will define a sequence of $x_1, x_2, \ldots, x_\ell \in \{0,1\}^n$ as well as the sets

$$S_1, S_1^{\mathsf{mist}} \subseteq S_1, S_2 \subseteq S_1 \setminus S_1^{\mathsf{mist}}, S_2^{\mathsf{mist}} \subseteq S_2, \ldots, S_\ell \subseteq S_{\ell-1} \setminus S_{\ell-1}^{\mathsf{mist}}, S_\ell^{\mathsf{mist}} \subseteq S_\ell.$$

For instance, if $\ell = 3$, we proceed as follows.

(i) We initially argue as in Section 4.2.2 with $\ell = 2$. In Case 1 (i.e., $|S_{x_1}^{\mathsf{mist}}| > (2/3) \cdot |S_{x_1}|$), we can simply apply the aforementioned circuit B_1 to approximate $L(M)$. However, we can no longer conclude in its Case 2 (i.e., $|S_{x_1}^{\mathsf{mist}}| < (2/3) \cdot |S_{x_1}|$) that x_2 is a mistake of D_w for every $w \in S_{x_1,x_2}$. To address this, we define the set

$$S_{x_1,x_2}^{\mathsf{mist}} \triangleq \{ w \in S_{x_1,x_2} \mid D_w(x_2) \neq M(x_2) \} \subseteq S_{x_1,x_2}$$

and consider its density in S_{x_1,x_2}.

(ii) If $|S_{x_1,x_2}^{\mathsf{mist}}| / |S_{x_1,x_2}| \geq 2/3$, we know that for at least a $2/3$ fraction of $w \in S_{x_1,x_2}$, x_2 is a mistake of D_w. As in Case 1 of Section 4.2.2, we apply Lemma E.1 (instead of a direct counting argument in Fact 4.11) to find a "good" $a \in \{0,1\}^{[n^c] \setminus J_{x_2}}$ such that $S_{x_1,x_2} \restriction_a / 2^{n^{c/2}} \geq \Omega(2^{-2n})$ and the density of $S_{x_1,x_2}^{\mathsf{mist}} \restriction_a$ in $S_{x_1,x_2} \restriction_a$ is at least $2/3 - 1/100$. By plugging in this a into the circuit B_2, we will achieve agreement $\geq 1/2 + \Omega(2^{-2n})$ with $L(M)$.

(iii) Otherwise, we assume that $|S_{x_1,x_2}^{\mathsf{mist}}| / |S_{x_1,x_2}| \geq 2/3$. Let $(x_2, y_2, z_2) = f_2(D_w, y'_w, z'_w)$. Similar to y'_w and z'_w, for every $w \in S_{x_1,x_2} \setminus S_{x_1,x_2}^{\mathsf{mist}}$, we define (y''_w, z''_w) as the lexicographic first pair such that

$$\Big[M(x_2, y_2) = 0 \vee D_w(x_2, z_2) = 1 \Big] \wedge \Big[M(x_2, y') = 1 \vee D_w(x_1, z') = 0 \Big],$$

that is, (y''_w, z''_w) is the output of the canonical Teacher in the second round of the Student-Teacher protocol. Since S_{x_1,x_2} has density at least $\Omega(2^{-2n})$, we can find a string $x_3 \in \{0,1\}^n$ such that the following set

$$S_{x_1,x_2,x_3} \triangleq \{ w \in S_{x_1,x_2} \setminus S_{x_1,x_2}^{\mathsf{mist}} \mid f_3(D_w, y'_w, z'_w, y''_w, z''_w) = x_3 \},$$

has density at least $\Omega(2^{-3n})$. Since x_3 must be a mistake of D_w when $\ell = 3$ and $w \in S_{x_1,x_2,x_3}$, and this set is sufficiently dense, we can obtain a deterministic circuit of size $2^{O(n)}$ that achieves agreement $\geq 1/2 + \Omega(2^{-3n})$ with $L(M)$.

The argument can be generalised in the natural way, which allows us to obtain a circuit of size $2^{O(n)}$ that approximates $L(M)$ with advantage $\geq 1/2 + \Omega(2^{-\ell n})$ in the case of a disjunction of length ℓ in the application of the KPT Witnessing (see Theorem 2.9). This deterministic circuit computes $L(M)$ on inputs of length $n^{c/2}$, where c is an arbitrary constant.

Remark 4.13. Note that the approach breaks down in theories where the number of rounds in the Student-Teacher game obtained from Theorem 2.9 is polynomial in the relevant parameter, as in the case of Buss's theory S^1_2 (see, e.g., [Kra92]). In the latter case, one can get up to $\ell = \mathsf{poly}(2^n)$ rounds in the corresponding witnessing theorem, and the advantage of the resulting deterministic circuit under a naive extension of the presented proof becomes trivial.

4.3 Extensions of the technique and unprovability of weaker lower bounds

As noted in [PS21], one can use hardness amplification to weaken the average-case hardness in the unprovability result (Theorem 4.1). By an adaptation of the proof of Theorem 4.1 via Remarks 4.5 and 4.12 and an application of Theorem 2.7, we can obtain the following unprovability result.

Theorem 4.14. *For every $n_0 \in \mathbb{N}$ and $\delta \in \mathbb{Q} \cap (0,1)$, if M is a nondeterministic machine whose running time is bounded by some constructive function $t(n) = 2^{n^{o(1)}}$, then*[21]

$$\mathsf{T}_{\mathsf{PV}} \nvdash \mathsf{LB}(M, s, m, n_0),$$

where $s(n) = 2^{n^\delta}$ and $m(n) = 2^n/n$.

As a consequence, for every language $L \in \mathsf{NTIME}[2^{n^{o(1)}}]$ and $\delta > 0$ it is consistent with T_{PV} that there are infinitely many input lengths n and a co-nondeterministic circuit D_n of size $\leq 2^{n^\delta}$ such that

$$\Pr_{x \sim \{0,1\}^n}[L(x) = D_n(x)] \geq 1 - 1/n.$$

Proof of Theorem 4.14. Let n_0, δ, M, $s(n) = 2^{n^\delta}$, and $m(n) = 2^n/n$ be as above. Assume towards a contradiction that

$$\mathsf{T}_{\mathsf{PV}} \vdash \mathsf{LB}(M, s, m, n_0).$$

Let $L \triangleq L(M)$ be the language defined by M. We argue as follows.

(i) Under the provability of an almost-everywhere average-case lower bound against conondeterministic circuits, it follows by the soundness of T_{PV} that (in the standard model) for every sequence $\{E_n\}_{n\geq 1}$ of *deterministic* circuits E_n of size $\leq 2^{n^\delta}$, if $n \geq n_0$ then

$$\Pr_{x \sim \{0,1\}^n}[L(x) = E_n(x)] \leq 1 - 1/n.$$

(ii) From the provability of $\mathsf{LB}(M, s, m, n_0)$, it follows that T_{PV} proves the sentence $\mathsf{LB}_{\mathsf{wst}}(M, s, n_0)$ which states a *worst-case* lower bound for M against conondeterministic circuits of the same size. By adapting the argument presented in Section 4.2 (see Remarks 4.5 and 4.12), the provability of $\mathsf{LB}_{\mathsf{wst}}(M, s, n_0)$ in T_{PV} implies that, in the standard model, for *every* sequence $\{g_n\}_{n\geq 1}$ of functions

[21] The original statement in [PS21] is slightly weaker: they require the nondeterministic machine M to be in polynomial-time instead of $t(n)$ time. We obtain such quantitative improvement by explicitly computing the complexity overhead of the hardness amplification in [HVV06] (see Theorem 2.7).

in $\mathsf{NSIZE}[2^{n^{o(1)}}]$, $\varepsilon > 0$, and large enough n, there is a deterministic circuit C' defined over n input variables and of size 2^{n^ε} such that

$$\Pr_{x \sim \{0,1\}^n}[g_n(x) = C'(x)] \geq 1/2 + 2^{-n^\varepsilon}. \tag{7}$$

(iii) Let $\{f_n\}_{n\geq 1}$ be the sequence of functions in $\mathsf{NTIME}[2^{n^{o(1)}}]$ obtained from L, i.e., $f(x) = 1$ if and only if $x \in L$. Note that this sequence satisfies the hypothesis of Theorem 2.7 for $s_1(n) = 2^{n^{o(1)}}$ and $s_2(n) = 2^{n^\delta}$ for sufficiently large n. Let $\{h_m\}_{m\geq 1}$ be the sequence of functions in $\mathsf{NSIZE}[2^{m^{o(1)}}]$ obtained by an application of this result, we know that for sufficiently large n and any deterministic circuit C of size $(2^{m^{\gamma\delta}})^\gamma$, it holds that

$$\Pr_{x \sim \{0,1\}^m}[h_m(x) = C(x)] \leq 1/2 + 2^{-\gamma m^{\gamma\delta}}.$$

Now the hardness of h_m according to Theorem 2.7 contradicts the upper bound provided in Equation (7), if we take $\varepsilon = (1/2) \cdot \delta \cdot \gamma$ and consider large enough input lengths.

This shows that $\mathsf{T}_{\mathsf{PV}} \nvdash \mathsf{LB}(M, s, m, n_0)$, as desired. □

5 Unprovability of Strong Complexity Lower Bounds in Bounded Arithmetic

In this section, we establish the unprovability of strong Σ^p_i-vs-Π^p_i-style lower bounds in bounded arithmetic. Our result generalises a previous unprovability result from [PS21] in two directions: (1) it holds for stronger theories $\mathsf{T}^i_{\mathsf{PV}}$ instead of only $\mathsf{T}^1_{\mathsf{PV}}$; and (2) the lower bound sentence in our unprovability result is more natural in the sense that the hard problem is quantified within the theory, instead of in the meta-theory.

Due to the complexity of the argument, we will first show in Section 5.1 how to generalise the unprovability result in [PS21] to $\mathsf{T}^i_{\mathsf{PV}}$. Then in Section 5.2 we combine this extension with the new game-theoretic witnessing theorem and with other ideas to obtain our main result, which has both features mentioned above.

5.1 Unprovability of lower bounds in expressive theories

For $i \geq 1$, recall that $\mathsf{T}^i_{\mathsf{PV}}$ is the theory consisting of all true (in the standard model) $\forall\Sigma^b_{i-1}(\mathsf{PV})$ sentences. For instance, $\mathsf{T}^1_{\mathsf{PV}}$ is the universal true theory of PV. We want to generalize the unprovability of strong nondeterministic circuit lower bounds in $\mathsf{T}^1_{\mathsf{PV}}$ to $\mathsf{T}^i_{\mathsf{PV}}$ for all $i \geq 1$, stated as follows.[22]

Theorem 5.1. *Fix $i \geq 1$. Let $t(n) = 2^{n^{o(1)}}$ be a constructive time bound, and M be a Π_i-$\mathsf{TIME}[t(n)]$ machine and $\mathsf{LB}^i(M, s, m, n_0)$ be the $\mathcal{L}_{\mathsf{PV}}$-sentence: for all $n \in \mathsf{LogLog}$ with $n > n_0$ and $C \in \Sigma_i$-$\mathsf{SIZE}[s(n)]$, there exist m distinct inputs $x_1, \ldots, x_m$ such that $M(x_j) \neq C(x_j)$ for all $j \in [m]$. Then*

$$\mathsf{T}^i_{\mathsf{PV}} \nvdash \mathsf{LB}^i(M, s, m, n_0)$$

for $s(n) = 2^{n^\delta}$, $m(n) = 2^n/2 - 2^n/2^{n^\delta}$, and $\delta \in \mathbb{Q} \cap (0, 1)$.

[22] While in Section 4 we considered a lower bound for a nondeterministic machine against co-nondeterministic circuits, it will be more convenient for us in this section to phrase the statement as Π_i-machines against Σ_i-circuits. Note that this is inconsequential, as the results are equivalent via complementation.

To obtain the unprovability of strong complexity lower bounds, we rely on a witnessing theorem that extracts computational information from a proof of the lower bound sentence $\mathsf{LB}^i(M, s, m, n_0)$. We discuss the quantifier complexity of (the worst-case complexity analogue of) the $\mathsf{LB}^i(M, s, m, n_0)$ sentence in Section 5.1.1. As its formalization results in a $\forall\Sigma^b_{i+1}(\mathsf{PV})$ sentence, note that when $i > 1$ we can no longer directly apply the KPT Witnessing Theorem, as in Section 4. (In addition, for $i > 1$ the theory $\mathsf{T}^i_{\mathsf{PV}}$ is not universal, which is needed when applying this result.) A key aspect of our argument is to introduce an appropriate universal theory with the right abstractions and term complexity (see Section 2.7).

5.1.1 Witnessing for Π_i vs Σ_i lower bounds

Let $\mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0)$ be the following *worst-case* lower bound sentence in the language $\mathcal{L}_{\mathsf{PV}}$:

> *For all $n \in \mathsf{LogLog}$ with $n > n_0$ and circuit $D \in \Sigma_i\text{-}\mathsf{SIZE}[s(n)]$, there exists an input x of length n, such that $D(x) \neq M(x)$.*

More formally, we have

$$\begin{aligned}\mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0) \quad \triangleq \quad & \forall n \in \mathsf{LogLog} \text{ with } n > n_0, \forall \text{ circuit } D \in \Sigma_i\text{-}\mathsf{SIZE}[s(n)] \\ & \exists x \in \{0,1\}^n \text{ such that } \mathsf{Error}(D, x),\end{aligned}$$

where $\mathsf{Error}(D, x)$ is a sentence stating that $M(x) \neq D(x)$. Since M is a Π_i-machine and D is a Σ_i-circuit, in the language $\mathcal{L}_{\mathsf{PV}}$, the sentence $\phi_1(D, x) \triangleq (M(x) = 1 \wedge D(x) = 0)$ is in Π^b_i and the sentence $\phi_2(D, x) \triangleq (M(x) = 0 \wedge D(x) = 1)$ is in Σ^b_i.

Fact 5.2. *Let $m(n) \geq 1$. If $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(M, s, m, n_0)$ then $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0)$.*

Proof. This is immediate for any reasonable formalization of the sentence $\mathsf{LB}^i(M, s, m, n_0)$, since it states an average-case lower bound (at least $m(n) \geq 1$ mistakes) while $\mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0)$ states a worst-case lower bound (i.e. at least one mistake). □

Assume that

$$\phi_1(D, x) \triangleq \forall y \in \{0,1\}^{O(s(n))}\ \phi'_1(D, x, y),$$
$$\phi_2(D, x) \triangleq \exists z \in \{0,1\}^{O(s(n))}\ \phi'_2(D, x, z),$$

for some Σ^b_{i-1}-formula ϕ'_1 and Π^b_{i-1}-formula ϕ'_2, respectively. Note that the lengths of the strings y and z are bounded by $O(s(n))$ since we obtain from them parts of the computation of the circuit D (of size $s(n)$) and of the machine M (with running time $2^{n^{o(1)}} < s(n)$). Then $\mathsf{Error}(D, x) \triangleq \phi_1(D, x) \vee \phi_2(D, x)$ is logically equivalent to the formula

$$\mathsf{Error}'(D, x) \triangleq \exists z \in \{0,1\}^{O(s(n))}\ \forall y \in \{0,1\}^{O(s(n))}\ (\phi'_1(D, x, y) \vee \phi'_2(D, x, z)).$$

Next, consider the universal theories $\mathsf{U}^i_{\mathsf{PV}}$ and $\mathsf{UT}^i_{\mathsf{PV}}$ introduced in Section 2.7.

Lemma 5.3. *Let $\mathsf{ULB}^i_{\mathsf{wst}}(M, s, n_0)$ be a Π^b_3-sentence in $\mathcal{L}(\mathsf{U}^i_{\mathsf{PV}})$ defined as follows:*

$$\begin{aligned}\mathsf{ULB}^i_{\mathsf{wst}}(M, s, n_0) \triangleq & \forall n \in \mathsf{LogLog} \textit{ with } n \geq n_0,\ \forall \textit{ circuit } D \in \Sigma_i\text{-}\mathsf{SIZE}[s(n)] \\ & \exists x \in \{0,1\}^n\ \exists z \in \{0,1\}^{O(s(n))} \\ & \forall y \in \{0,1\}^{O(s(n))}\ (f_{\phi'_1}(D, x, y) = 1 \vee f_{\phi'_2}(D, x, z) = 1).\end{aligned}$$

Then $\mathsf{U}^i_{\mathsf{PV}}$ proves $\mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0) \leftrightarrow \mathsf{ULB}^i_{\mathsf{wst}}(M, s, n_0)$. Moreover, $\mathsf{UT}^i_{\mathsf{PV}}$ proves $\mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0) \leftrightarrow \mathsf{ULB}^i_{\mathsf{wst}}(M, s, n_0)$.

Proof. By the discussion above, $\mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0)$ is logically equivalent to

$$\begin{aligned}
&\forall n \in \mathsf{LogLog} \text{ with } n \geq n_0,\ \forall \text{ circuit } D \in \Sigma_i\text{-}\mathsf{SIZE}[s(n)] \\
&\exists x \in \{0,1\}^n\ \exists z \in \{0,1\}^{O(s(n))} \\
&\forall y \in \{0,1\}^{O(s(n))}\ (\phi'_1(D, x, y) = 1 \vee \phi'_2(D, x, z) = 1),
\end{aligned}$$

which is further equivalent to $\mathsf{ULB}^i_{\mathsf{wst}}(M, s, n_0)$ in $\mathsf{U}^i_{\mathsf{PV}}$ by Lemma 2.13. The provability of the same sentence in $\mathsf{UT}^i_{\mathsf{PV}}$ follows from Theorem 2.18. □

Note that the $\mathcal{L}(\mathsf{U}^i_{\mathsf{PV}})$-sentence $\mathsf{ULB}^i_{\mathsf{wst}}(M, s, n_0)$ has low quantifier complexity. By exploring the connection between $\mathsf{T}^i_{\mathsf{PV}}$ and the universal theory $\mathsf{UT}^i_{\mathsf{PV}}$, we can show the following witnessing result.

Lemma 5.4 (Witnessing lemma for $\mathsf{LB}(M, s, m, n_0)$)**.** *Let $i \geq 1$, M be a Π_i-$\mathsf{TIME}[2^{n^{o(1)}}]$ machine, $\delta \in (0,1)$, $n_0 \geq 1$, $s(n) = 2^{n^\delta}$, and $m(n) = 2^n/2 - 2^n/2^{n^\delta}$. If $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(M, s, m, n_0)$, then there exist $\ell \in \mathbb{N}$ and ℓ algorithms $A_1, A_2, \ldots, A_\ell$ such that:*

- *Every A_i is computable in $\mathsf{FP}^{\Sigma^p_{i-1}}$ over inputs of length of order $N = 2^n$.*
- *For every $i \in [\ell]$, the input of A_i consists of 1^N, 1^n for $n = \log N$, an n-input circuit $D \in \Sigma_i$-$\mathsf{SIZE}[s(n)]$, and $i - 1$ strings $y_1, \ldots, y_{i-1}$; the output of A_i is a pair $(x_i, z_i) \in \{0,1\}^n \times \{0,1\}^{O(s(n))}$.*[23]
- *Let $h : (n, D, x) \mapsto y$ be the following function. Given n, a string $x \in \{0,1\}^n$, and a circuit $D \in \Sigma_i$-$\mathsf{SIZE}[s(n)]$, output a y such that $\neg\phi'_1(D, x, y)$ if such $y \in \{0,1\}^{O(s(n))}$ exists, or 0 otherwise.*
- *For all $n > n_0$ and circuit $D \in \Sigma_i$-$\mathsf{SIZE}[s(n)]$, let*

  $$\begin{array}{rcl}
  (x_1, z_1) \triangleq A_1(1^n, D) & & y_1 \triangleq h(n, D, x_1) \\
  (x_2, z_2) \triangleq A_2(1^n, D, y_1) & & y_2 \triangleq h(n, D, x_2) \\
  \vdots & & \vdots \\
  (x_\ell, z_\ell) \triangleq A_\ell(1^n, D, y_1, \ldots, y_{i-1}) & & y_\ell \triangleq h(n, D, x_\ell).
  \end{array}$$

  Then there is an index $v \in [\ell]$ such that $D(x_v) \neq M(x_v)$.

Proof. Let $i, M, \delta, n_0, s(n), m(n)$ be defined as above. Assume that $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(M, s, m, n_0)$. Then, by Fact 5.2, it follows that $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i_{\mathsf{wst}}(M, s, m, n_0)$. Using Theorem 2.14 and Lemma 5.3, we get that $\mathsf{UT}^i_{\mathsf{PV}} \vdash \mathsf{ULB}^i_{\mathsf{wst}}(M, s, m, n_0)$.

Since $\mathsf{ULB}^i_{\mathsf{wst}}(M, s, m, n_0)$ is a $\forall\Sigma^b_2$-sentence and $\mathsf{UT}^i_{\mathsf{PV}}$ is a universal theory, we can invoke the KPT witnessing theorem (Theorem 2.9) to obtain constantly many $\mathcal{L}^i_{\mathsf{PV}}$-terms $A_1, A_2, \ldots, A_\ell$ witnessing the existential quantifier given counter-examples to the innermost universal quantifier. By Theorem 2.17 and using that $n \in \mathsf{LogLog}$, each A_i is computable in $\mathsf{FP}^{\Sigma^p_{i-1}}$ over an input of order $N = 2^n$. Furthermore, since the function h is a valid counter-example oracle for the innermost universal quantifier, it is easy to check that the conclusion of the lemma follows from the guarantee provided by KPT witnessing. □

[23] Formally, since $n \in \mathsf{LogLog}$ in our formalization, each A_i has access to an input α of length $|\alpha| = N = 2^n$. For convenience of notation, when discussing $A_1, \ldots, A_\ell$ we often omit the input 1^N and concentrate on n, which is the key parameter.

5.1.2 Proof of Theorem 5.1

Theorem (Theorem 5.1, restated)**.** *Fix $i \geq 1$. Let M be a Π_i-$\mathsf{TIME}[2^{n^{o(1)}}]$ machine and $\mathsf{LB}^i(M, s, m, n_0)$ be the $\mathcal{L}_{\mathsf{PV}}$-sentence: for all $n \in \mathsf{LogLog}$ with $n > n_0$ and Σ_i-$\mathsf{SIZE}[s(n)]$-circuit C, there exist m distinct inputs $x_1, \ldots, x_m$ such that $M(x_j) \neq C(x_j)$ for all $j \in [m]$. Then*

$$\mathsf{T}^i_{\mathsf{PV}} \nvdash \mathsf{LB}^i(M, s, m, n_0)$$

for $s(n) = 2^{n^\delta}$, $m(n) = 2^n/2 - 2^n/2^{n^\delta}$, and $\delta \in \mathbb{Q} \cap (0, 1)$.

Proof. Suppose that $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(M, s, m, n_0)$ for some $M \in \Pi_i$-$\mathsf{TIME}[2^{n^{o(1)}}]$, $n_0 \in \mathbb{N}$, $s(n) = 2^{n^\delta}$, $m(n) = 2^n/2 - 2^n/2^{n^\delta}$, and $\delta \in \mathbb{Q} \cap (0, 1)$, there exist an $\ell \in \mathbb{N}$ and ℓ algorithms $A_1, A_2, \ldots, A_\ell$ as described by Lemma 5.4. Similar to [PS21], we will utilize the algorithms A_i to show that M can be non-trivially approximated by Σ_i-$\mathsf{SIZE}[2^{n^\delta}]$ circuits for some $n > n_0$, leading to a contradiction to the soundness of $\mathsf{T}^i_{\mathsf{PV}}$.

Let c be a constant to be determined later, and $\mathsf{NW}_f(w, x)$ be the Nisan-Wigderson generator with seed length $|w| = n^c$, output length 2^n, "hard" function $f : \{0, 1\}^{n^{c/2}} \to \{0, 1\}$ (therefore each subset in the combinatorial design has size $n^{c/2}$), $|x| = n$, and any two distinct subsets in the combinatorial design with intersection of size at most n. For every seed $w \in \{0, 1\}^{n^c}$, let $D_w : \{0, 1\}^n \to \{0, 1\}$ be a Σ_i-circuit computing $D_w(x) \triangleq \mathsf{NW}_{\overline{M}}(w, x)$, which is of size at most $2^{n^{o(c)}} \leq 2^{n^\delta}$ for sufficient large n. We will find some $w \in \{0, 1\}^{n^c}$ and use D_w as C in Lemma 5.4 to obtain a Σ_i-$\mathsf{SIZE}[2^{O(n)}]$ circuit B approximating M on input length $n^{c/2}$, i.e., $\Pr_{u \in \{0,1\}^{n^{c/2}}}[B(u) = M(u)] \geq \frac{1}{2} + 2^{-O(n)}$. Then by choosing $c > 2/\delta$ and sufficiently large n, we can prove the theorem.

Case 1. Recall that in Lemma 5.4, A_1 takes 1^n and an n-input circuit $D \in \Sigma_i$-$\mathsf{SIZE}[s(n)]$ as input and output a pair $(x, y) \in \{0, 1\}^n \times \{0, 1\}^{O(s(n))}$. By an averaging argument, there is an $x_1 \in \{0, 1\}^n$ such that for a uniformly random $w \in \{0, 1\}^{n^c}$, with probability at least 2^{-n}, $A_1(1^n, D_w)$ outputs $(x_1, \cdot)$. Fix this x_1 and let

$$S_1 \triangleq \left\{ w \in \{0, 1\}^{n^c} \;\middle|\; A_1(1^n, D_w) = (x_1, \cdot) \right\},$$
$$S_1^{\mathsf{mist}} \triangleq \left\{ w \in S_1 \;\middle|\; D_w(x_1) \neq M(x_1) \right\}.$$

By the definition of x_1, we know that $|S_{x_1}|/2^{n^c} \geq 2^{-n}$.

In Case 1 we assume that $|S_1^{\mathsf{mist}}| > (2/3) \cdot |S_1|$, handling the other scenario in a subsequent case. For any $w \in \{0, 1\}^{n^c}$, we know that $D_w(x_1) = \mathsf{NW}_{\overline{M}}(w, x_1) = \overline{M}(w|_{J_{x_1}})$, where J_{x_1} is the subset of indices corresponding to the x_1-th row of the combinatorial design. By Lemma E.1, there is an assignment $a \in \{0, 1\}^{[n^c] \setminus J_{x_1}}$ for the indices outside of J_{x_1} such that $|S_1 \restriction_a | / 2^{n^{c/2}} \geq 2^{-O(n)}$ and $|S_1^{\mathsf{mist}} \restriction_a | / |S_1 \restriction_a | \geq 3/5$. Fix an $a \in \{0, 1\}^{n^c \setminus J_{x_1}}$ as above. Let $b_1 \triangleq M(x_1) \in \{0, 1\}$. We define a randomized circuit $B_1 : \{0, 1\}^{n^{c/2}} \times \{0, 1\} \to \{0, 1\}$, where the second input is regarded as a random bit, as follows (see Algorithm 5 and recall the notation from Section 2.4).

We first analyse the complexity of B_1. Let $m = n^{c/2} = |u|$ be the input length. Since A_1 is computable in $\mathsf{FP}^{\Sigma^p_{i-1}}$, it is easy to see that $B_1 \in \mathsf{SIZE}^{\Sigma^p_{i-1}}[2^{O(n)}]$. By Theorem 2.4, we get that $B_1 \in \Sigma_i$-$\mathsf{SIZE}[2^{O(n)}]$. So we only need to show that for an random bit $r \in \{0, 1\}$, $B(x, r)$ approximates $M(x)$ well.

```
Input  : The input $u \in \{0,1\}^{n^{c/2}}$ for $M$ and a bit $r \in \{0,1\}$
Advice: $x_1 \in \{0,1\}^n$, $a \in \{0,1\}^{[n^c] \setminus J_{x_1}}$, and $b_1 = M(x_1)$
Let $w = r_{x_1}(a,u)$ and $(x,\cdot) = A_1(1^n, D_w)$;
If $x \neq x_1$, return $r$;
Otherwise, return $b_1$.
```

Algorithm 5: Randomized Circuit B_1 for M

For any input $u \in \{0,1\}^{n^{c/2}}$ such that $u \in S_1 \upharpoonright_a$, we know that

$$\begin{aligned} B(u,r) = M(u) &\iff M(x_1) = M(u) && (x = x_1 \text{ by the definition of } S_1,\ B(u,r) = b_1 = M(x_1)) \\ &\iff M(x_1) \neq D_w(x_1) && (D_w(x_1) = \mathsf{NW}_{\overline{M}}(w,x_1) = \overline{M}(w|_{J_{x_1}}) = \overline{M}(u)) \\ &\iff u \in S_1^{\mathsf{mist}} \upharpoonright_a . \end{aligned}$$

This means that $B(u,r)$ and M agree on at least $3/5$ of the inputs $u \in S_1 \upharpoonright_a$. In the other case, the circuit B outputs the random bit r, therefore for some fixed bit $r^* \in \{0,1\}$, $B_1(u,r^*)$ and $M(u)$ agree on at least $1/2$ of the inputs $u \notin S_1 \upharpoonright_a$. Since $|S_1 \upharpoonright_a| / 2^{n^{c/2}} \geq 2^{-O(n)}$, we obtain that

$$\Pr_{u \in \{0,1\}^{n^{c/2}}} \Big[B_1(u,r^*) = M(u) \Big] \geq \frac{3}{5} \cdot \frac{|S_1 \upharpoonright_a|}{2^{n^{c/2}}} + \frac{1}{2} \cdot \left(1 - \frac{|S_1 \upharpoonright_a|}{2^{n^{c/2}}} \right) = \frac{1}{2} + 2^{-O(n)}.$$

Case 2. Assume that $|S_1^{\mathsf{mist}}| \leq (2/3) \cdot |S_1|$ instead. For every $w \in S_1$, we define $y_1(w) \triangleq h(n, D_w, x_1)$ to be the output of the counter-example oracle h in Lemma 5.4. Again by an averaging argument, there is an $x_2 \in \{0,1\}^n$ such that for a uniformly random $w \in S_1 \setminus S_1^{\mathsf{mist}}$, with probablity at least 2^{-n}, $A_2(1^n, D_w, y_1(w)) = (x_2, \cdot)$. Fix this x_2. Let S_2 an S_2^{mist} be the sets defined as follows:

$$\begin{aligned} S_2 &\triangleq \Big\{ w \in S_1 \setminus S_1^{\mathsf{mist}} \,\Big|\, A_2(1^n, D_w, y_1(w)) = (x_2, \cdot) \Big\}, \\ S_2^{\mathsf{mist}} &\triangleq \{ w \in S_2 \mid D_w(x_2) \neq M(x_2) \}. \end{aligned}$$

By the definition of x_2 we know that $|S_2| / 2^{n^c} \geq (1/3) \cdot 2^{-O(n)} = 2^{-O(n)}$.

In this case, we further assume that $|S_2^{\mathsf{mist}}| > (2/3) \cdot |S_2|$. By construction, for any $w \in \{0,1\}^{n^c}$, $D_w(x_2) = \overline{M}(w|_{J_{x_2}})$. By Lemma E.1, there is an assignment $a \in \{0,1\}^{[n^c] \setminus J_{x_2}}$ for the indices outside of J_{x_2} such that $|S_2 \upharpoonright_a| / 2^{n^{c/2}} \geq 2^{-O(n)}$ and $|S_2^{\mathsf{mist}} \upharpoonright_a| / |S_2 \upharpoonright_a| \geq 3/5$. Fix this a. We will need the following subroutine to complete this case.

(∇) Given $w \in S_1$ of the form $a \cup u$ ($u \in \{0,1\}^{J_{x_2}}$), there is a deterministic circuit $E(w)$ of size at most $2^{O(n)}$ that outputs $(y_1(w), e_1(w))$, where $y_1(w) = h(n, D_w, x_1)$ and $e_1(w) \in \{0,1\}$ such that $e_1(w) = 1$ if and only if $w \in S_1^{\mathsf{mist}}$.

Note that the circuit $E(w)$ is used to simulate the counter-example oracle h in the first round of the KPT-style game. Let $b_2 \triangleq M(x_2)$. We construct a randomized circuit B_2 as follows (see Algorithm 6), discussing the claim (∇) later in the proof.

Let $m = n^{c/2} = |u|$ be the input length. We first show that $B_2 : \{0,1\}^m \times \{0,1\} \to \{0,1\}$ can be implemented by a Σ_i-circuit with size $2^{O(n)}$. Both A_1 and A_2 are $\mathsf{FP}^{\Sigma_{i-1}^p}$ algorithms with input length $\mathsf{poly}(2^n)$, so both of them can be implemented by $\mathsf{SIZE}^{\Sigma_{i-1}^p}[2^{O(n)}]$ circuits. By (∇) we also know that $E(w)$

```
Input  : The input $u \in \{0,1\}^{n^{c/2}}$ for $M$ and random bit $r \in \{0,1\}$
Advice: $x_1, x_2 \in \{0,1\}^n$, $a \in \{0,1\}^{[n^c] \setminus J_{x_2}}$ as discussed, $b_2 = M(x_2)$, and $\Gamma$ to support the
        subroutine $(\nabla)$
Let $w = r_{x_2}(a, u)$ and $(\hat{x}_1, \hat{z}_1) = A_1(1^n, D_w)$;
If $\hat{x}_1 \neq x_1$, then return the random bit $r$; // after this step, $w \in S_1$
Let $(y_1(w), e_1(w)) = E(w)$ by $(\nabla)$;
If $e_1(w) = 1$, then return the random bit $r$; // after this step, $w \in S_1 \setminus S_1^{\mathsf{mist}}$
Let $(\hat{x}_2, \hat{z}_2) = A_2(1^n, D_w, y_1(w))$;
If $\hat{x}_2 \neq x_2$, then return the random bit $r$;
Otherwise, return $b_2$. // reaching this line if and only if $w \in S_2$
```

Algorithm 6: Randomized circuit B_2 for M

can be implemented by a $2^{O(n)}$-size circuit. As a result, $B_2 \in \mathsf{SIZE}^{\Sigma_{i-1}^p}[2^{O(n)}] \subseteq \Sigma_i\text{-}\mathsf{SIZE}[2^{O(n)}]$, where the last inclusion follows from Theorem 2.4.

Now we prove the correctness of the algorithm B_2. By construction, it is easy to verify that the algorithm reaches the last line if and only if $w = r_{x_2}(a, u) \in S_2$. Therefore B_2 will output a random bit when $w \notin S_2$ (i.e., $u \notin S_2 \upharpoonright_a$) and output b_2 when $w \in S_2$ (i.e., $u \in S_2 \upharpoonright_a$). In the former case, B_2 agrees with M on $1/2$ of the inputs for an $r^* \in \{0,1\}$, which will be hard-wired into the circuit. In the latter case, with probability at least $3/5$ over $u \in \{0,1\}^{n^{c/2}}$, $u \in S_2^{\mathsf{mist}} \upharpoonright_a$, which further means that

$$M(u) = M(w|_{J_{x_2}}) = \overline{D_w}(x_2) = M(x_2) = b_2 = B_2(u, r).$$

Since $|S_2 \upharpoonright_a| / 2^{n^{c/2}} \geq 2^{-O(n)}$, we can conclude that $B_2(u, r^*)$ agrees with $M(u)$ on $\frac{1}{2} + 2^{-O(n)}$ of the inputs $u \in \{0,1\}^{n^{c/2}}$.

Case $j \geq 2$. Using the technique for Case 2, we can in fact deal with all the remaining cases. Let $j \in \{2, 3, \dots, \ell\}$. We define the following notations recursively:

(i) $y_{j-1}(w) \triangleq h(n, D_w, x_{j-1})$.

(ii) $x_j \in \{0,1\}^n$ be the lexicographically first string such that for a uniformly random string $w \in S_{j-1} \setminus S_{j-1}^{\mathsf{mist}}$, with probability at least 2^{-n}, $A_j(1^n, D_w, y_1(w), \dots, y_{j-1}(w)) = (x_j, \cdot)$.

(iii) Define S_j and S_j^{mist} as the sets

$$S_j \triangleq \left\{ w \in S_{j-1} \setminus S_{j-1}^{\mathsf{mist}} \;\middle|\; A_j(1^n, D_w, y_1(w), \dots, y_{j-1}(w)) = (x_j, \cdot) \right\}$$
$$S_j^{\mathsf{mist}} \triangleq \{ w \in S_j \mid D_w(x_j) \neq M(x_j) \}.$$

In Case $j \geq 2$ we assume that (1) $|S_j^{\mathsf{mist}}| > (2/3) \cdot |S_j|$, and (2) for every $i \in \{1, 2, \dots, j-1\}$, $|S_i^{\mathsf{mist}}| / |S_i| \leq 2/3$. Crucially, by the definition of each of these sets and the conclusion of Lemma 5.4, we get that by reaching $j = \ell$ we necessarily have $S_\ell = S_\ell^{\mathsf{mist}}$, so the case analysis is complete.

The following lemma will be needed later in the proof.

Lemma 5.5. *For every $1 \leq i < j$, we have $x_i \neq x_j$.*

Proof. Suppose that $x_i = x_j$ for $i < j$. First, it follows from the construction that $S_j \cap S_i^{\mathsf{mist}} = \emptyset$. Therefore, for any $w \in S_j^{\mathsf{mist}} \subseteq S_j$, we have $M(x_i) = D_w(x_i)$. On the other hand, by definition, for any $w \in S_j^{\mathsf{mist}}$, we have $M(x_j) \neq D_w(x_j)$. Note that the assumption of Case $j \geq 2$ implies that S_j^{mist} is nonempty. Take any $w^* \in S_j^{\mathsf{mist}}$. Under the hypothesis that $x_i = x_j$, the previous claims yield that both $M(x_i) = D_{w^*}(x_i)$ and $M(x_i) \neq D_{w^*}(x_i)$, which is contradictory. □

Under assumptions (1) and (2), one can prove by induction that $|S_j|/2^{n^c} = 2^{-O(n)}$, therefore by Lemma E.1, there is an assignment $a \in \{0,1\}^{[n^c]\setminus J_{x_j}}$ such that $|S_j \restriction_a|/2^{n^{c/2}} \geq 2^{-O(n)}$ and $|S_j^{\mathsf{mist}} \restriction_a|/|S_j \restriction_a| \geq 3/5$. Similarly to (∇) in Case 2, we need the following computation (∇_j^i) for every $i \in \{1, 2, \dots, j-1\}$.

(∇_j^i) Given $w \in S_i$ of the form $a \cup u$ ($u \in \{0,1\}^{J_{x_j}}$), there is a deterministic circuit $E_i(w)$ of size at most $2^{O(n)}$ that outputs $(y_i(w), e_i(w))$, where $y_i(w) = h(n, D_w, x_i)$ and $e_i(w) \in \{0,1\}$ such that $e_i(w) = 1$ if and only if $w \in S_i^{\mathsf{mist}}$.

Note that (∇_2^1) is simply (∇) in Case 2. With these subroutines we can construct a randomized circuit B_j that approximates M well as follows (see Algorithm 7).

```
Input  : The input $u \in \{0,1\}^{n^{c/2}}$ for $M$ and random bit $r \in \{0,1\}$
Advice: $x_1, \dots, x_j \in \{0,1\}^n$, $a \in \{0,1\}^{[n^c]\setminus J_{x_j}}$ as discussed above, $b_j = M(x_j)$, and $\Gamma$ to support
         the subroutines $(\nabla_j^i)$
Let $w = r_{x_j}(a, u)$;
for $i = 1, 2, \dots, j$ do
    Let $(\hat{x}_i, \hat{z}_i) = A_i(1^n, D_w, y_1, \dots, y_{i-1})$;
    If $\hat{x}_i \neq x_i$, then return the random bit $r$;
    // reaching this line iff $w \in S_i$
    if $i < j$ then
        Let $(y_i(w), e_i(w)) = E_i(w)$ by $(\nabla_j^i)$;
        If $e_1(w) = 1$, then return the random bit $r$;
        // otherwise, $x \in S_i \setminus S_i^{\mathsf{mist}}$
    end
end
// reaching this line iff $w \in S_j$
return $b_j$;
```

Algorithm 7: Randomized circuit B_j for M

Now we analyze the complexity and correctness of the algorithm B_j.

(Complexity). Let $m = n^{c/2}$ be the input length. Since every A_i is computable in $\mathsf{FP}^{\Sigma_{i-1}^p}$ with input length $\mathsf{poly}(2^n, s(n)) = 2^{O(n)}$, and every E_i is computable by a $2^{O(n)}$-size deterministic circuit, we know that $B_j \in \mathsf{SIZE}^{\Sigma_{i-1}^p}[2^{O(n)}] \subseteq \Sigma_i\text{-}\mathsf{SIZE}[2^{O(n)}]$ (see Theorem 2.4).

(Correctness). The correctness of B_j is proved similarly to Case 2. As noted in the comments appearing in the pseudo-code, for any $w = r_{x_j}(a, u)$ with $u \in \{0,1\}^{n^{c/2}}$, we can prove by induction that the algorithm reaches the end of the i-th iteration within the for-loop if and only if $w \in S_i \setminus S_i^{\mathsf{mist}}$ for every $i \in \{1, 2, \dots, j-1\}$. Furthermore, on such inputs the algorithm reaches the last line if and only if $w \in S_j$. This means that, for an appropriate fixed bit $r^* \in \{0,1\}$, on inputs of this form the algorithm agrees with M on at least $1/2$ of $w \notin S_j$ and on at least $3/5$ of $w \in S_j$. Since $|S_j \restriction_a|/2^{n^{c/2}} = 2^{-O(n)}$, $B_j(\cdot, r^*)$ achieves an advantage of $2^{-O(n)}$ with $M(\cdot)$.

Implementation of (∇). To finish the proof, we need to upper bound the circuit complexity of the computation $E_i(w)$ in ∇^i_j for $1 \leq i < j \leq \ell$, which is used to simulate the counter-example oracle h and to check if $w \in S^{\mathsf{mist}}_i$, for w of the appropriate form. Let $j \in \{2, 3, \ldots, \ell\}$ and $i \in \{1, 2, \ldots, j-1\}$. Thanks to Lemma 5.5, we know that $x_i \neq x_j$. Recall that $D_w(x) \triangleq \mathsf{NW}_{\overline{M}}(w, x)$, where $w \triangleq r_{x_j}(a, u)$. Since any two distinct subsets in the combinatorial design of the Nisan-Wigderson generator have intersection size at most n, we get that $|J_{x_i} \cap J_{x_j}| \leq n$. Notice that for $u \in \{0,1\}^{n^{c/2}}$, $h(n, D_w, x_i)$ for $w = r_{x_j}(a, u)$ only depends on $w|_{J_{x_i}}$, which contains at most n bits of u. As a result, we can hard-wire the answers of all 2^n cases and construct a $2^{O(n)}$ circuit to compute $y_i(w) = h(n, D_w, x_i)$. The value $e_i(w)$ can be computed in a similar way. Overall, we obtain that $E_i(w) \triangleq (y_i(w), e_i(w))$ can be computed on all relevant inputs by a (non-uniform) deterministic circuit of size $2^{O(n)}$.

Wrapping things up. By the case analysis above, for every $c \geq 2$ and every sufficiently large n, there always exists a Σ_i-$\mathsf{SIZE}[2^{O(n)}]$ circuit B with input length $m = n^{c/2}$ such that

$$\Pr_{u \in \{0,1\}^m}[B(u) = M(u)] \geq \frac{1}{2} + 2^{-O(n)}.$$

By taking $c \geq 2/\delta$ we get, that for infinitely many values of n, $L(M) \cap \{0,1\}^n$ can be approximated with advantage at least 2^{-n^δ} by Σ_i-$\mathsf{SIZE}[2^{n^\delta}]$ circuits. This leads to a contradiction, since under $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(M, s, m, n_0)$ and from the soundness of $\mathsf{T}^i_{\mathsf{PV}}$ we obtain that, for sufficiently large n, M cannot be approximated with advantage 2^{-n^δ} by Σ_i-$\mathsf{SIZE}[2^{n^\delta}]$ circuits. □

As pointed out in Remark 4.5 and Remark 4.12, we note that assuming the provability of *worst-case* circuit lower bound, the approximation of the machine $M \in \Pi_i$-$\mathsf{TIME}[2^{n^{o(1)}}]$ by deterministic Σ^p_{i-1}-oracle circuits of small size also works for any sequence $\{f_n\}_{n \geq 1}$ of functions computable in Π_i-$\mathsf{SIZE}[2^{n^{o(1)}}]$. Concretely, instead of defining $D_w(x) \triangleq \mathsf{NW}_{\overline{M}}(w, x)$, we define $D_w(x) \triangleq \mathsf{NW}_{f'}(w, x)$ for $f'(x) \triangleq \neg f(x)$, and proceed the argument as above. By padding dammy input bits as in Remark 4.12, we obtain the following corollary.

Corollary 5.6. *Fix $i \geq 1$. Let M be a Π_i-$\mathsf{TIME}[2^{n^{o(1)}}]$ machine and $\mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0)$ be the worst-case lower bound sentence defined as above. Assume that for some $\delta \in (0,1) \cap \mathbb{Q}$ and $s(n) = 2^{n^\delta}$, $\mathsf{T}^i_{\mathsf{PV}}$ proves $\mathsf{LB}^i_w(M, s, n_0)$. Then for every constant $\epsilon \in (0,1)$, every sufficiently large n, and circuit $C \in \Pi_i$-$\mathsf{SIZE}[2^{n^\delta}]$, there is a Σ^p_{i-1}-oracle circuit D of size 2^{n^ϵ} such that*

$$\Pr_{x \sim \{0,1\}^n}\Big[C(x) = D(x)\Big] \geq \frac{1}{2} + \frac{1}{2^{n^\epsilon}}.$$

5.1.3 Relaxing the average-case complexity parameter

Recall that in Section 4.3, we showed how to obtain the unprovability of circuit lower bounds with a weaker average-case complexity parameter via hardness amplification. This will also be the case here, since the hardness amplification in [HVV06] generalises to all levels in the polynomial hierarchy (see Theorem 2.7).

Theorem 5.7. *Fix $i \geq 1$. Let M be a Π_i-$\mathsf{TIME}[t(n)]$ machine for some constructive function $t(n) = 2^{n^{o(1)}}$ and $\mathsf{LB}^i(M, s, m, n_0)$ be defined as in Theorem 5.1. Then for every constant $\delta \in \mathbb{Q} \cap (0, 1)$, $s(n) \triangleq 2^{n^\delta}$, $m(n) \triangleq 2^n/n$, and $n_0 \in \mathbb{N}$, $\mathsf{T}^i_{\mathsf{PV}} \nvdash \mathsf{LB}^i(M, s, m, n_0)$.*

Proof. Towards a contradiction, we assume that $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(M, s, m, n_0)$ and argue as follows.

(i) Under the provability of the almost-everywhere average-case lower bound $\mathsf{LB}(M, s, m, n_0)$, we obtain from the soundness of $\mathsf{T}^i_{\mathsf{PV}}$ that (in the standard model) for every sequence $\{E_n\}_{n\geq 1}$ of Σ^p_{i-1}-oracle circuits of size $\leq 2^{n^\delta}$ and $n \geq n_0$, we have

$$\Pr_{x\sim\{0,1\}^n}[M(x) = E_n(x)] \leq 1 - \frac{1}{n}.$$

(ii) From the provability of $\mathsf{LB}(M, s, m, n_0)$, under reasonable formalization, we can also show that $\mathsf{LB}^i_{\mathsf{wst}}(M, s, n_0)$ is provable in $\mathsf{T}^i_{\mathsf{PV}}$. We then get from Corollary 5.6 that for every $\epsilon \in (0, 1)$, every sufficiently large n, and circuit $C \in \Pi_i$-$\mathsf{SIZE}[2^{n^\delta}]$, there is a Σ^p_{i-1}-oracle circuit D of size 2^{n^ϵ} such that

$$\Pr_{x\sim\{0,1\}^n}[C(x) = D(x)] \geq \frac{1}{2} + \frac{1}{2^{n^\epsilon}}. \tag{8}$$

(iii) Assume that n is sufficiently large and $f_n : \{0,1\}^n \to \{0,1\}$ is defined as $f(x) = M(x)$. Note that this function satisfies the hypothesis of Theorem 2.7 for $s_1(n) = 2^{n^{o(1)}}$ and $s_2(n) = 2^{n^\delta}$, hence we can obtain a function $h_\ell : \{0,1\}^\ell \to \{0,1\}$ for some $\ell = O(n^2)$ such that for *every* Σ^p_{i-1}-oracle circuit D of size $2^{\gamma\ell^{\gamma\delta}}$, it holds that

$$\Pr_{x\in\{0,1\}^\ell}[h_\ell(x) = D(x)] \leq \frac{1}{2} + \frac{1}{2^{\gamma\ell^{\gamma\delta}}}.$$

By setting $\epsilon = (1/2) \cdot \delta \cdot \gamma$, this violates the upper bound in Equation (8) when n is sufficiently large.

As a result, we know that $\mathsf{T}^i_{\mathsf{PV}} \nvdash \mathsf{LB}^i(M, s, m, n_0)$ for every $i \geq 1$. □

5.2 Unprovability of lower bound sentences of higher quantifier complexity

In this section, we extend the unprovability results to sentences of higher quantifier complexity that formalize separations between non-uniform circuit classes. Recall that Σ_i-$\mathsf{SIZE}[s(n)]$ and Π_i-$\mathsf{SIZE}[s(n)]$ refer to Σ_i-circuits and Π_i-circuits of size $s(n)$, respectively. Let $\mathsf{LB}^i(s_1, s_2, m, n_0)$ denote the following $\mathcal{L}_{\mathsf{PV}}$-sentence:

$$\begin{aligned}&\forall n \in \mathsf{LogLog} \text{ with } n \geq n_0\ \exists C \in \Pi_i\text{-}\mathsf{SIZE}[s_1(n)]\ \forall D \in \Sigma_i\text{-}\mathsf{SIZE}[s_2(n)]\\ &\exists m = m(n) \text{ distinct } n\text{-bit strings } x^1, \ldots, x^m \text{ s.t. } \mathsf{Error}(C, D, x^i) \text{ for all } i \in [m],\end{aligned}$$

where $\mathsf{Error}(C, D, x)$ means that the circuits C and D do not agree on the input x. It's easy to see that $\mathsf{Error}(C, D, x)$ is a disjunction of a Σ^b_i-formula and a Π^b_i-formula. Observe that, already for $i = 1$, $\mathsf{LB}^i(s_1, s_2, m, n_0)$ is a $\forall\Sigma^b_4$-sentence.

Theorem 5.8. *For every $i \geq 1$, $n_0 \in \mathbb{N}$, $\delta \in \mathbb{Q} \cap (0, 1)$ and $d \geq 1$, $\mathsf{T}^i_{\mathsf{PV}} \nvdash \mathsf{LB}^i(s_1, s_2, m, n_0)$, where $s_1(n) = n^d$, $s_2(n) = 2^{n^\delta}$ and $m(n) = 2^n/2 - 2^n/2^{n^\delta}$.*

5.2.1 Witnessing lemma for lower bound sentences

Similar to the technique we used in the previous section, we need to apply the witnessing theorem to the lower bound sentences. We define the worst-case version of this lower bound to be the following formula $\mathsf{LB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$.

$$\begin{aligned}&\forall n \in \mathsf{LogLog} \text{ with } n \geq n_0\ \exists C \in \Pi_i\text{-}\mathsf{SIZE}[s_1(n)]\ \forall D \in \Sigma_i\text{-}\mathsf{SIZE}[s_2(n)]\\ &\exists x \in \{0,1\}^n \text{ s.t. } \mathsf{Error}(C, D, x).\end{aligned}$$

Let $\phi_1(C, D, x) \triangleq (C(x) = 1 \wedge D(x) = 0)$ be a Π^b_i-formula and $\phi_2(C, D, x) \triangleq (C(x) = 0 \wedge D(x) = 1)$ be a Σ^b_i-formula. Note that $\mathsf{Error}(C, D, x) \triangleq \phi_1(C, D, x) \vee \phi_2(C, D, x)$. Assume that

$$\begin{aligned}\phi_1(C, D, x) &\triangleq \forall y \in \{0,1\}^{O(s(n))} \phi'_1(C, D, x, y),\\ \phi_2(C, D, x) &\triangleq \exists z \in \{0,1\}^{O(s(n))} \phi'_2(C, D, x, z),\end{aligned}$$

where ϕ'_1 is a Σ^b_{i-1}-formula and ϕ'_2 is a Π^b_{i-1}-formula. Note that the lengths of y and z are bounded by $O(s(n))$ since they are parts of the computation of the circuits C and D.

Lemma 5.9. *Let $\mathsf{ULB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$ be a $\forall\Sigma^b_4$-sentence in $\mathcal{L}(\mathsf{U}^i_{\mathsf{PV}})$ defined as follows:*

$$\begin{aligned}\mathsf{ULB}^i_{\mathsf{wst}}(s_1, s_2, n_0) \triangleq\ & \forall n \in \mathsf{LogLog} \textit{ with } n \geq n_0,\ \exists \textit{ circuit } C \in \Pi_i\text{-}\mathsf{SIZE}[s_1(n)]\\ & \forall \textit{ circuit } D \in \Sigma_i\text{-}\mathsf{SIZE}[s_2(n)],\\ & \exists x \in \{0,1\}^n\ \exists z \in \{0,1\}^{O(s(n))}\ \forall y \in \{0,1\}^{O(s(n))}\\ & \left(f_{\phi'_1}(C, D, x, y) = 1 \vee f_{\phi'_2}(C, D, x, z) = 1\right).\end{aligned}$$

Then $\mathsf{U}^i_{\mathsf{PV}}$ proves $\mathsf{LB}^i_{\mathsf{wst}}(s_1, s_2, n_0) \leftrightarrow \mathsf{ULB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$. Moreover, $\mathsf{UT}^i_{\mathsf{PV}}$ proves $\mathsf{LB}^i_{\mathsf{wst}}(s_1, s_2, n_0) \leftrightarrow \mathsf{ULB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$.

Proof. The provability in $\mathsf{U}^i_{\mathsf{PV}}$ follows from the provability of the defining axioms for f_α (see Lemma 2.13). In turn, the provability in $\mathsf{UT}^i_{\mathsf{PV}}$ follows from Theorem 2.18. □

Lemma 5.10. *Assume that $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(s_1, s_2, m, n_0)$. There is an integer $\ell \in \mathbb{N}$ and $\mathsf{FP}^{\Sigma^p_{i-1}}$ algorithms $P_1, Q_1, P_2, Q_2, \ldots, P_\ell, Q_\ell$ such that the following condition holds.*[24]

Let $n > n_0$, g be a function that maps a Π_i-$\mathsf{SIZE}[s_1(n)]$-circuit to a Σ_i-$\mathsf{SIZE}[s_2(n)]$ circuit, and $D_C \triangleq g(C)$. Let $h : (n, C, D, x) \mapsto y$ be the function such that y is the lexicographic first string in $\{0,1\}^{O(s(n))}$ such that $\neg\phi'_1(C, D, x, y)$ holds or 0 if such a string does not exist. Let

$$\begin{array}{ccc} P_1(1^n) = C_1 & Q_1(1^n, D_{C_1}) = (x_1, z_1) & h(n, C_1, D_{C_1}, x_1) = y_1 \\ P_2(1^n, D_{C_1}, y_1) = C_2 & Q_2(1^n, D_{C_1}, D_{C_2}, y_1) = (x_2, z_2) & h(n, C_2, D_{C_2}, x_2) = y_2 \\ \vdots & \vdots & \vdots \\ P_\ell(1^n, D_{C_{1\ldots\ell-1}}, y_{1\ldots\ell-1}) = C_\ell & Q_\ell(1^n, D_{C_{1\ldots\ell}}, y_{1\ldots\ell-1}) = (x_\ell, z_\ell) & h(n, C_\ell, D_{C_\ell}, x_\ell) = y_\ell\ . \end{array}$$

Then there is $k \in [\ell]$ such that $\mathsf{Error}(C_k, D_{C_k}, x_k)$ holds.

[24] As in the statement of Lemma 5.4, the input length of these algorithms is of order $N = 2^n$, since in our formalisation $n \in \mathsf{LogLog}$. In order to be succinct, we simply write 1^n as one of the inputs, since n is the key parameter for us.

Proof. Suppose that $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(s_1, s_2, m, n_0)$. Then we also have $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$, which further means by Theorem 2.14, Lemma 5.9, and Theorem 2.18 that $\mathsf{UT}^i_{\mathsf{PV}} \vdash \mathsf{ULB}^i_{\mathsf{wst}}(s_1, s_2, n_0)$. Recall that $\mathsf{UT}^i_{\mathsf{PV}}$ is a universal theory closed under if-then-else (see Theorem 2.18). By Theorem 3.2, there are an $\ell \in \mathbb{N}$ and a sequence of ℓ $\mathcal{L}$-strategies $\tau^{\mathsf{t}}_1, \tau^{\mathsf{t}}_2, \ldots, \tau^{\mathsf{t}}_\ell$ for the truthifier such that for any fixed strategy τ^{f} of the falsifier, at least one of the strategies beats τ^{f} in ℓ sequential plays of the evaluation game with $\tau^{\mathsf{t}}_1, \tau^{\mathsf{t}}_2, \ldots, \tau^{\mathsf{t}}_\ell$ vs τ^{f}.

In particular, consider the following strategy for the falsifier: if the truthifier chooses C in the first round of the game, the falsifier will choose D_C; then if the truthifier chooses x, z in the second round, the falsifier will choose $h(n, C, D_C, x)$. It is easy to see that the claim in the lemma is precisely the winning property of $\tau^{\mathsf{t}}_1, \ldots, \tau^{\mathsf{t}}_\ell$ against this particular strategy for the falsifier, given the corresponding auxiliary information. □

5.2.2 Proof of Theorem 5.8

Theorem (Reminder of Theorem 5.8)**.** *For every $i \geq 1$, $n_0 \in \mathbb{N}$, $\delta \in \mathbb{Q} \cap (0, 1)$ and $d \geq 1$, $\mathsf{T}^i_{\mathsf{PV}} \nvdash \mathsf{LB}^i(s_1, s_2, m, n_0)$, where $s_1(n) = n^d$, $s_2(n) = 2^{n^\delta}$ and $m(n) = 2^n/2 - 2^n/2^{n^\delta}$.*

Proof. Assume that $\mathsf{T}^i_{\mathsf{PV}} \vdash \mathsf{LB}^i(s_1, s_2, m, n_0)$. We will derive a contradiction to the soundness of $\mathsf{T}^i_{\mathsf{PV}}$ by showing that for sufficiently large n and all Π_i-circuits $M : \{0,1\}^{n^{c/2}} \to \{0,1\}$ of size $s_1(n^{c/2})$, there is a Σ_i-circuit $B : \{0,1\}^{n^{c/2}} \to \{0,1\}$ of size at most $s_2(n^{c/2})$ that agrees with M on all but at most $m(n^{c/2})$ inputs, for some constant $c \in \mathbb{N}$ which will be determined later.

Let $\mathsf{NW}_f(w, x)$ be the Nisan-Wigderson generator with: $f : \{0,1\}^{n^{c/2}} \to \{0,1\}$, seed length $|w| = n^c$, $|x| = n + n^d$, and any two distinct subsets in the combinatorial design of intersection of size at most $O(n^d)$. Designs with these parameters are known to exist (see Section 2.4).

By Lemma 5.10, we have $\ell \in \mathbb{N}$ and $\mathsf{FP}^{\Sigma^p_{i-1}}$ machines $P_1, P_2, \ldots, P_\ell, Q_1, \ldots, Q_\ell$ as described. Let $M : \{0,1\}^{n^{c/2}} \to \{0,1\}$ be a Π_i-circuit of size $s_1(n^{c/2})$ as described above. Let $D_{w,C} : \{0,1\}^n \to \{0,1\}$ be a Σ_i-circuit of size at most $s_2(n)$ computing $D_{w,C}(x) \triangleq \mathsf{NW}_{\overline{M}}(w, x \| C)$ for $w \in \{0,1\}^{n^c}$ and $C \in \{0,1\}^{n^d}$.[25] We would like to find some suitable w and apply Lemma 5.10 with $g : C \mapsto D_{w,C}$ to obtain a circuit $B \in \mathsf{SIZE}^{\Sigma^p_{i-1}}[2^{O(n^d)}] \subseteq \Sigma_i\text{-}\mathsf{SIZE}[2^{O(n^d)}]$ approximating M, i.e. $\Pr_u[B(u) = M(u)] \geq \frac{1}{2} + 2^{-O(n^d)}$. By choosing c as a constant much larger than d, we can prove the theorem.

Case 1. Let $C_1 \triangleq P_1(1^n)$. By an averaging argument, there is an $x_1 \in \{0,1\}^n$ such that for a uniformly random $w \in \{0,1\}^{n^c}$, with probability at least 2^{-n}, the first coordinate of $Q_1(1^n, D_{w,C_1})$ is x_1. Fix this x_1 and let

$$S_1 \triangleq \Big\{ w \in \{0,1\}^{n^c} \mid Q_1(1^n, D_{w,C_1}) = (x_1, \cdot) \Big\},$$
$$S_1^{\mathsf{mist}} \triangleq \Big\{ w \in S_1 \mid D_{w,C_1}(x_1) \neq C_1(x_1) \Big\}.$$

By the definition of x_1 we get that $|S_1|/2^{n^c} \geq 2^{-n}$.

In this case we assume that $|S_1^{\mathsf{mist}}| \geq (2/3) \cdot |S_1|$, dealing with the other situation in a subsequent case analysis. For any w, we know that $D_{w,C_1}(x_1) = \mathsf{NW}_{\overline{M}}(w, x_1 \| C_1) = \overline{M}(w|_{J_{x_1\|C_1}})$, where $J_{x_1\|C_1}$ is the subset of indices corresponding to the $(x_1\|C_1)$-th row of the combinatorial design. By Lemma E.1, there is an assignment $a \in \{0,1\}^{[n^c] \setminus J_{x_1\|C_1}}$ for the indices outside of $J_{x_1\|C_1}$ such that $|S_1 \restriction_a|/2^{n^{c/2}} \geq 2^{-O(n)}$ and $|S_1^{\mathsf{mist}} \restriction_a|/|S_1 \restriction_a| \geq 3/5$.

```
Input : The input $u \in \{0,1\}^{n^{c/2}}$ for $M$ and random bit $r \in \{0,1\}$
Advice: $x_1 \in \{0,1\}^n$, $C_1 = P_1(1^n)$, $a \in \{0,1\}^{[n^c] \setminus J_{x_1\|C_1}}$ as discussed, and $b_1 = C_1(x_1)$
Let $w = r_{x_1\|C_1}(a,u)$ and $(x,\cdot) = Q_1(1^n, D_{w,C_1})$;
If $x \neq x_1$, return the random bit $r$;
Otherwise, return $b_1$.
```

Algorithm 8: Randomized circuit B_1 for M

Now we fix $a \in \{0,1\}^{[n^c] \setminus J_{x_1\|C_1}}$ as above. Let $b_1 \triangleq C_1(x_1) \in \{0,1\}$. We define a randomized circuit B_1 with access to a random bit $r \in \{0,1\}$ as follow (see Algorithm 8).

Since $Q_1 \in \mathsf{FP}^{\Sigma_{i-1}^p}$ and $|D_{w,C_1}| = 2^{n^{o(1)}}$, it is clear that $B_1 \in \mathsf{SIZE}^{\Sigma_{i-1}^p}[2^{O(n)}] \subseteq \Sigma_i\text{-}\mathsf{SIZE}[2^{O(n^d)}]$, so we only need to verify that the randomized circuit B_1 approximates M. For $u \in \{0,1\}^{n^{c/2}}$ such that $u \in S_1 \restriction_a$, we have that

$$\begin{aligned} B_1(u,r) = M(u) &\iff C_1(x_1) = M(u) && (x = x_1 \text{ by the definition of } S_1,\ B(u,r) = b_1 = C(x_1)) \\ &\iff C_1(x_1) \neq D_{w,C_1}(x_1) && (D_{w,C_1}(x_1) = \mathsf{NW}_{\overline{M}}(w, x_1\|C_1) = M(u)) \\ &\iff u \in S_1^{\mathsf{mist}} \restriction_a . \end{aligned}$$

Therefore B_1 and M agree on at least $3/5$ of the inputs $u \in S_1 \restriction_a$. In the other case, the circuit B simply outputs the random bit r, therefore for a specific $r^* \in \{0,1\}$, $B_1(u,r^*)$ and $M(u)$ agree on at least $1/2$ of the inputs $u \notin S_1 \restriction_a$. Since $|S_1 \restriction_a|/2^{n^{c/2}} \geq 2^{-O(n)}$, we obtain that

$$\Pr_{u \in \{0,1\}^{n^{c/2}}} \Big[B_1(u,r^*) = M(u) \Big] \geq \frac{3}{5} \cdot \frac{|S_1 \restriction_a|}{2^{n^{c/2}}} + \frac{1}{2} \cdot \left(1 - \frac{|S_1 \restriction_a|}{2^{n^{c/2}}} \right) = \frac{1}{2} + 2^{-O(n)}.$$

Case 2. Assume that $|S_1^{\mathsf{mist}}| \leq (2/3) \cdot |S_1|$. Let $h(n,C,D,x_1)$ be the function described in Lemma 5.10. Let $y_1(w) \triangleq h(n, C_1, D_{w,C_1}, x_1)$ and $C_2^w = P_2(1^n, D_{w,C_1}, y_1(w))$. Again, by an averaging argument, there are $C_2 \in \{0,1\}^{n^d}$ and $x_2 \in \{0,1\}^n$ such that for a uniformly random $w \in S_1 \setminus S_1^{\mathsf{mist}}$, with probability at least $2^{-O(n^d)}$, $C_2 = C_2^w$ and $Q_2(1^n, D_{w,C_1}, D_{w,C_2}, y_1(w)) = (x_2, \cdot)$. Fix this C_2 and x_2. Let S_2 and S_2^{mist} be sets defined as follows:

$$\begin{aligned} S_2 &\triangleq \Big\{ w \in S_1 \setminus S_1^{\mathsf{mist}} \mid C_2 = C_2^w \wedge Q_2(1^n, D_{w,C_1}, D_{w,C_2}, y(w)) = (x_2, \cdot) \Big\} \\ S_2^{\mathsf{mist}} &\triangleq \{ w \in S_2 \mid D_{w,C_2}(x_2) \neq C_2(x_2) \} \end{aligned}$$

By the definitions of C_2 and x_2, we know that $|S_2|/2^{n^c} \geq (1/3) \cdot 2^{-O(n^d)} = 2^{-O(n^d)}$.

In this case we assume that $|S_2^{\mathsf{mist}}| \geq (2/3) \cdot |S_2|$. Similarly to Case 1, for any $w \in \{0,1\}^{n^c}$, $D_{w,C_2}(x_2) = \overline{M}(w|_{J_{x_2\|C_2}})$. By Lemma E.1, there is an assignment $a \in \{0,1\}^{[n^c] \setminus J_{x_2\|C_2}}$ for the indices outside of $J_{x_2\|C_2}$ such that $|S_2 \restriction_a|/2^{n^{c/2}} \geq 2^{-O(n^d)}$ and $|S_2^{\mathsf{mist}} \restriction_a|/|S_2 \restriction_a| \geq 3/5$. Fix this string a. We will assume the following computation is possible in order to complete this case, returning to it later on:

(∇) Given $w \in S_1$ of the form $a \cup u$ ($u \in \{0,1\}^{J_{x_2\|C_2}}$), there is a deterministic circuit $E(w)$ of size at most $2^{O(n^d)}$ that outputs $(y_1(w), e_1(w))$, where $y_1(w) = h(n, C_1, D_{w,C_1}, x_1)$ and $e_1(w) \in \{0,1\}$ such that

[25] We use $u\|v$ to denote the concatenation of binary strings u and v. Jumping ahead, the idea of concatenating $x\|C$ when defining the NW generator will allow us to establish an analogue of Lemma 5.5 in this proof.

$e_1(w) = 1$ if and only if $w \in S_1^{\mathsf{mist}}$.

Note that if $w \in S_1 \setminus S_1^{\mathsf{mist}}$, $y_1(w)$ given by $E(w)$ witnesses that $\neg\mathsf{Error}(C_1, D_{w,C_1}, x_1)$. Let $b_2 \triangleq C_2(x_2)$. We construct a randomized circuit B_2 as follows (see Algorithm 9).

```
Input  : The input $u \in \{0,1\}^{n^{c/2}}$ for $M$ and random bit $r \in \{0,1\}$
Advice: $x_1, x_2 \in \{0,1\}^n$, $C_1, C_2 \in \{0,1\}^{n^d}$, $a \in \{0,1\}^{[n^c] \setminus J_{x_2 \| C_2}}$ as discussed, $b_2 = C_2(x_2)$, and
        $\Gamma$ to support the subroutine $(\nabla)$
Let $w = r_{x_2\|C_2}(a, u)$ and $(\hat{x}_1, \hat{z}_1) = Q_1(1^n, D_{w,C_1})$;
If $\hat{x}_1 \neq x_1$, then return the random bit $r$; // after this step, $w \in S_1$
Let $(y_1(w), e_1(w)) = E(w)$ by $(\nabla)$;
If $e_1(w) = 1$, then return the random bit $r$; // after this step, $w \in S_1 \setminus S_1^{\mathsf{mist}}$
Let $\hat{C}_2 = P_2(1^n, D_{w,C_1}, y_1(w))$ and $(\hat{x}_2, \hat{z}_2) = Q_2(1^n, D_{w,C_1}, D_{w,C_2}, y_1(w))$;
If $\hat{x}_2 \neq x_2$ or $\hat{C}_2 \neq C_2$, then return the random bit $r$;
Otherwise, return $b_2$. // reaching this line if and only if $w \in S_2$
```

Algorithm 9: Randomized circuit B_2 for M

First, we analyze the complexity of B_2. Since $Q_1, P_2, Q_2 \in \mathsf{FP}^{\Sigma_{i-1}^p}$ and the input length for each of them is of order $2^{O(n)}$, they can be implemented by circuits of size $2^{O(n)}$ with Σ_{i-1}^p oracles. We need $2^{O(n^d)}$ gates to support the computation (∇). Therefore, $B_2 \in \mathsf{SIZE}^{\Sigma_{i-1}^p}[2^{O(n^d)}] \subseteq \Sigma_i\text{-}\mathsf{SIZE}[2^{O(n^d)}]$.

By construction, it is easy to verify that the algorithm reaches the last line if and only if $w = r_{x_2\|C_2}(a, u) \in S_2$. Therefore B_2 will output a random bit when $w \notin S_2$ and output b_2 when $w \in S_2$. In the former case, B_2 agrees with M on $1/2$ of the inputs for an $r^* \in \{0,1\}$. In the latter case, with probability at least $3/5$, $w|_{J_{x_2\|C_2}} \in S_2^{\mathsf{mist}} \restriction_a$, which further means that

$$M(u) = M(w|_{J_{x_2\|C_2}}) = \overline{D_{w,C_2}}(x_2) = C_2(x_2) = b_2 = B_2(u, r).$$

As a result, it follows that

$$\Pr_{u \in \{0,1\}^{n^{c/2}}}\Big[B_2(u, r^*) = M(u)\Big] \geq \frac{3}{5} \cdot \frac{|S_2 \restriction_a|}{2^{n^{c/2}}} + \frac{1}{2} \cdot \left(1 - \frac{|S_2 \restriction_a|}{2^{n^{c/2}}}\right) = \frac{1}{2} + 2^{-O(n^d)}.$$

Case $j \geq 2$. Using the technique for Case 2, we can in fact deal with all the remaining cases. Let $j \in \{2, 3, \ldots, \ell\}$. We recursively define the following values:

(i) $y_{j-1}(w) \triangleq h(n, C_{j-1}, D_{w,C_{j-1}}, x_{j-1})$.

(ii) $C_j^w \triangleq P_j(1^n, D_{w,C_1}, \ldots, D_{w,C_{j-1}}, y_1(w), \ldots, y_{j-1}(w))$.

(iii) Let $C_j \in \{0,1\}^{n^d}$ be the lexicographical first string (encoding an circuit) such that for a uniformly random string $w \in S_{j-1} \setminus S_{j-1}^{\mathsf{mist}}$, with probability at least $2^{-O(n^d)}$, $C_j^w = C_j$. The existence of C_j follows from a counting argument.

(iv) Let $x_j \in \{0,1\}^n$ be the lexicographical first string such that for a uniformly random string $w \in (S_{j-1} \setminus S_{j-1}^{\mathsf{mist}}) \cap \{w \in \{0,1\}^{n^c} \mid C_j^w = C_j\}$, with probability at least 2^{-n},

$$Q_j(1^n, D_{w,C_1}, \ldots, D_{w,C_j}, y_1(w), \ldots, y_{j-1}(w)) = (x_j, \cdot).$$

Thus, for a uniformly random string $w \in S_{j-1} \setminus S_{j-1}^{\mathsf{mist}}$, with probability at least $2^{-O(n^d)} \cdot 2^{-n} = 2^{-O(n^d)}$, $C_j^w = C_j$ and $Q_j(1^n, D_{w,C_1}, \ldots, D_{w,C_j}, y_1(w), \ldots, y_{j-1}(w)) = (x_j, \cdot)$.

(v) S_j and S_j^{mist} be sets recursively defined as

$$\begin{aligned} S_j \triangleq & \Big\{ w \in S_{j-1} \setminus S_{j-1}^{\mathsf{mist}} \mid C_j^w = C_j \wedge \\ & Q_j(1^n, D_{w,C_1}, \ldots, D_{w,C_j}, y_1(w), \ldots, y_{j-1}(w)) = (x_j, \cdot) \Big\} \\ S_j^{\mathsf{mist}} \triangleq & \{ w \in S_j \mid D_{w,C_j}(x_j) \neq C_j(x_j) \} \end{aligned}$$

In Case j we will assume that (1) $|S_j^{\mathsf{mist}}|/|S_j| \geq 2/3$ and (2) for any $i \in \{1, 2, \ldots, j-1\}$, $|S_i^{\mathsf{mist}}|/|S_i| < 2/3$. In particular, by Lemma 5.10 we know that if we reach $j = \ell$ then $S_\ell = S_\ell^{\mathsf{mist}}$, so all the cases can be resolved in this way.

The following lemma will be useful later in the proof.

Lemma 5.11. *For every $1 \leq i < j$, we have $(C_i, x_i) \neq (C_j, x_j)$.*

Proof. First, note that $S_j \cap S_i^{\mathsf{mist}} = \emptyset$. Also, since we are in case j, $S_j^{\mathsf{mist}} \neq \emptyset$, given that $|S_j^{\mathsf{mist}}| \geq 2/3 \cdot |S_j|$ and the (inductively established) density lower bound for $|S_j|$. Now take any $w^* \in S_j^{\mathsf{mist}}$, i.e.,

$$C_j(x_j) \neq D_{w^*,C_j}(x_j). \tag{9}$$

Since $S_j^{\mathsf{mist}} \subseteq S_j$ and $S_j \cap S_i^{\mathsf{mist}} = \emptyset$, we have that $w^* \notin S_i^{\mathsf{mist}}$, i.e.,

$$C_i(x_i) = D_{w^*,C_i}(x_i). \tag{10}$$

Now if we had $(C_i, x_i) = (C_j, x_j)$, this would be in contradiction with Equation (9) and Equation (10). Consequently, either $C_i \neq C_j$ or $x_i \neq x_j$. □

We can prove by induction that $|S_j|/2^{n^c} = 2^{-O(n^d)}$, therefore by Lemma E.1, there is an assignment $a \in \{0,1\}^{[n^c] \setminus J_{x_j \| C_j}}$ such that $|S_j \restriction_a|/2^{n^{c/2}} \geq 2^{-O(n^d)}$ and $|S_j^{\mathsf{mist}} \restriction_a|/|S_j \restriction_a| \geq 3/5$. Fix this string a. Similar to (∇) in Case 2, we need the following computation (∇_j^i) for every $i \in \{1, 2, \ldots, j-1\}$.

(∇_j^i) Given $w \in S_i$ of the form $a \cup u$ ($u \in \{0,1\}^{J_{x_j \| C_j}}$), there is a deterministic circuit $E_i(w)$ of size at most $2^{O(n^d)}$ that outputs $(y_i(w), e_i(w))$, where $y_i(w) = h(n, C_i, D_{w,C_i}, x_i)$ and $e_i(w) \in \{0,1\}$ such that $e_i(w) = 1$ if and only if $w \in S_i^{\mathsf{mist}}$.

Note that $(\nabla_2^1) = (\nabla)$ by definition. Let $b_j \triangleq C_j(x_j)$. Using the subroutines described above, We can now present a randomized circuit B_j that approximates M (see Algorithm 10).

Similarly to Case 2, we can see that $B_j \in \mathsf{SIZE}^{\Sigma_{i-1}^p}[2^{O(n^d)}]$. Now we prove the correctness of B_j. By the definition of S_i, we can prove by induction that the algorithm reaches the end of the i-th iteration within the for-loop if and only if $w \in S_i \setminus S_i^{\mathsf{mist}}$ for any $i \in \{1, 2, \ldots, j-1\}$. We can further check that the algorithm reaches the last line if and only if $w \in S_j$. This means that, by fixing an appropriate bit r^* as the random bit, the algorithm agrees with M on at least $1/2$ of $w \notin S_j$ of the form $w = r_{x_j}(a, u)$, and on at least $3/5$ of $w \in S_j$ of the form $w = r_{x_j}(a, u)$. As before, this translates into a correlation of $2^{-O(n^d)}$ over a random input $u \in \{0,1\}^{n^{c/2}}$ using the lower bound on the density of $S_j \restriction_a$.

```
Input  : The input $u \in \{0,1\}^{n^{c/2}}$ for $M$ and random bit $r \in \{0,1\}$
Advice: $x_1, \dots, x_j \in \{0,1\}^n$, $C_1, \dots, C_j \in \{0,1\}^{n^d}$, $a \in \{0,1\}^{[n^c]\setminus J_{x_j\|C_j}}$ as discussed,
        $b_j = C_j(x_j)$, and $\Gamma$ to support the subroutines $(\nabla_j^i)$
Let $w = r_{x_j}(a, u)$;
for $i = 1, 2, \dots, j$ do
    Let $\hat{C}_i = P_i(1^n, D_{w,C_1}, \dots, D_{w,C_{i-1}}, y_1(w), \dots, y_{i-1}(w))$;
    If $\hat{C}_i \neq C_i$, then return the random bit $r$;
    Let $(\hat{x}_i, \hat{z}_i) = Q_i(1^n, D_{w,C_1}, \dots, D_{w,C_i}, y_1(w), \dots, y_{i-1}(w))$;
    If $\hat{x}_i \neq x_i$, then return the random bit $r$;
    // reaching this line iff $w \in S_i$
    if $i < j$ then
        Let $(y_i(w), e_i(w)) = E_i(w)$ by $(\nabla_j^i)$;
        If $e_i(w) = 1$, then return the random bit $r$;
        // otherwise, $x \in S_i \setminus S_i^{\mathsf{mist}}$
    end
end
// reaching this line iff $w \in S_j$
return $b_j$;
```

Algorithm 10: Randomized circuit B_j for M

Implementation of (∇). To complete the proof it is sufficient to show that (∇_j^i) in the j-th step is computable by $2^{O(n^d)}$-size circuits, for all $j \in \{2, 3, \dots, \ell\}$ and $1 \leq i < j$. Fix any $j \in \{2, 3, \dots, \ell\}$ and $i < j$. Recall that $h(n, C_i, D_{w,C_i}, x_i)$ finds the minimal y_i such that $\neg\phi_1'(C_i, D_{w,C_i}, x_i, y_i)$ holds if $\neg\phi_1(C_i, D_{w,C_i}, x_i)$, where $\neg\phi_1(C_i, D_{w,C_i}, x_i)$ means that $C_i(x_i) = 0 \vee D_{w,C_i}(x_i) = 1$. In case $C_i(x_i) = 0$, we only need to hard-wire a witness of it, since C_i and x_i are fixed with respect to w.

Now we assume that $C_i(x_i) = 1$. By the definition of D_{w,C_i}, we know that $D_{w,C_i}(x_i) = \mathsf{NW}_{\overline{M}}(w, x_i\|C_i)$, where $w = a \cup u$ for $a \in \{0,1\}^{[n^c]\setminus J_{x_j\|C_j}}$, $u \in \{0,1\}^{J_{x_j\|C_j}}$. By Lemma 5.11, $(x_i, C_i) \neq (x_j, C_j)$. Therefore, by the definition of the NW generator, for $w = a \cup u$ with the input $u \in \{0,1\}^{J_{x_j\|C_j}}$, the output $D_{w,C_i}(x_i)$, as well as the desired witness of the outer-most quantified variable in case that $D_{w,C_i}(x_i) = 1$, depends on at most $O(n^d)$ bits of u. In such case, we can hard-wire all the $2^{O(n^d)}$ answers with a deterministic $2^{O(n^d)}$-size circuit.

Similarly, it is not hard to show that the computation $e_i(w)$ can also be implemented by a deterministic circuit of at most this size.

Wrapping things up. Finally we can combine all these facts to conclude this theorem. By assuming $\mathsf{T}_{\mathsf{PV}}^i \vdash \mathsf{LB}^i(s_1, s_2, m, n_0)$, we proved that for sufficiently large n and all Π_i-circuits $M : \{0,1\}^{n^{c/2}} \to \{0,1\}$ of size $s_1(n^{c/2}) = n^{dc/2}$, there is a deterministic circuit $B : \{0,1\}^{n^{c/2}} \to \{0,1\}$ with Σ_{i-1}^p oracle gates of size at most $2^{O(n^d)}$ that agrees with M on a $1/2 + 2^{-O(n^d)}$ fraction of inputs. By Theorem 2.4, we know that B can be implemented by Σ_i-circuits of size $2^{O(n^d)}$. If we choose $c > 2d/\delta$, B is of size $\leq 2^{(n^{c/2})^\delta}$ and agrees with M on a $\geq 1/2 + 2^{-(n^{c/2})^\delta}$ fraction of the inputs $u \in \{0,1\}^{n^{c/2}}$, which means that $\mathbb{N} \vDash \neg\mathsf{LB}(s_1, s_2, m, n_0)$ for the corresponding choice of $m(n)$. This is a contradiction to the soundness of $\mathsf{T}_{\mathsf{PV}}^i$. □

Similarly to what was noted in Remark 4.12 and Corollary 5.6, the proof presented above shows that one can approximate *every* Π_i-$\mathsf{SIZE}[2^{n^{o(1)}}]$ circuit M by small-size Σ_{i-1}^p-oracle circuits, assuming the provability of the worst-case circuit lower bound sentence $\mathsf{LB}_{\mathsf{wst}}(s_1, s_2, n_0)$. We simply use $D_{w,C}(x) \triangleq \mathsf{NW}_{\overline{M}}(w, x\|C)$ and proceed as above. By padding dummy input bits as in Remark 4.12, we can obtain the following corollary.

Corollary 5.12. *Fix $i \geq 1$. Assume that for some $n_0 \in \mathbb{N}$, $\delta \in \mathbb{Q}\cap(0,1)$, and $d \geq 1$, $\mathsf{T}_{\mathsf{PV}}^i \vdash \mathsf{LB}_{\mathsf{wst}}^i(s_1, s_2, n_0)$ for $s_1(n) = n^d$ and $s_2(n) = 2^{n^\delta}$. Then for every constant $\epsilon > 0$, every sufficiently large n, and circuit $A \in \Pi_i$-$\mathsf{SIZE}[t(n)]$ where $t(n) = 2^{n^{o(1)}}$ is some constructive funtion, there is a Σ_{i-1}^p-oracle circuit B of size 2^{n^ϵ} such that*

$$\Pr_{x\sim\{0,1\}^n}[A(x) = B(x)] \geq \frac{1}{2} + \frac{1}{2^{n^\epsilon}}.$$

5.2.3 Relaxing the average-case complexity parameter

As in Section 5.1.3, we now utilize the hardness amplification theorem (see Theorem 2.7) to relax the average-case complexity parameter.

Theorem 5.13. *For every $i \geq 1$, $n_0 \in \mathbb{N}$, $\delta \in \mathbb{Q} \cap (0, 1)$, and $d \geq 1$, $\mathsf{T}_{\mathsf{PV}}^i \nvdash \mathsf{LB}^i(s_1, s_2, m, n_0)$, where $s_1(n) = n^d$, $s_2(n) = 2^{n^\delta}$, and $m = 2^n/n$.*

Proof. Suppose that $s_1 = s_1(n)$, $s_2 = s_2(n)$, m, and n_0 are defined as above. Towards a contradiction, we assume that $\mathsf{T}_{\mathsf{PV}}^i \vdash \mathsf{LB}^i(s_1, s_2, m, n_0)$.

(i) Under the unprovability of the almost-everywhere average-case lower bound $\mathsf{LB}(s_1, s_2, m_0)$, we obtain from the soundness of $\mathsf{T}_{\mathsf{PV}}^i$ that (in the standard model) for sufficiently large n, there is a circuit $C \in \Pi_i$-$\mathsf{SIZE}[s_1(n)]$ such that for every Σ_{i-1}^p-oracle circuit D of size 2^{n^δ}, we have

$$\Pr_{x\sim\{0,1\}^n}[C(x) = D(x)] \leq 1 - \frac{1}{n}.$$

(ii) By the assumption that $\mathsf{T}_{\mathsf{PV}}^i \vdash \mathsf{LB}^i(s_1, s_2, m, n_0)$, under any reasonable formalization, we know that $\mathsf{T}_{\mathsf{PV}}^i$ also proves the *worst-case* version of the lower bound $\mathsf{LB}_{\mathsf{wst}}^i(s_1, s_2, n_0)$. Then by Corollary 5.12, we get that for every constant $\epsilon > 0$, every sufficiently large n, and every Π_i-$\mathsf{SIZE}[2^{n^{o(1)}}]$ circuit, there is a Σ_{i-1}^p-oracle circuit D of size 2^{n^ϵ} such that

$$\Pr_{x\sim\{0,1\}^n}[C(x) = D(x)] \geq \frac{1}{2} + \frac{1}{2^{n^\epsilon}}. \tag{11}$$

(iii) Now we assume that n is sufficiently large and $f_n : \{0,1\}^n \to \{0,1\}$ is the function computable by Π_i-$\mathsf{SIZE}[s_1(n)]$ circuits in Item (i) that is hard on average against Σ_{i-1}^p-oracle circuit. By Theorem 2.7, there is a function $h_\ell : \{0,1\}^\ell \to \{0,1\}$ for some $\ell = O(n^2)$ that is computable by Π_i-$\mathsf{SIZE}[\mathsf{poly}(n) \cdot s_1(n)]$ circuits, such that for every Σ_{i-1}^p-oracle circuit D of size $2^{\gamma\ell^{\gamma\delta}}$,

$$\Pr_{x\sim\{0,1\}^\ell}[h_\ell(x) = D(x)] \leq \frac{1}{2} + \frac{1}{2^{\gamma\ell^{\gamma\delta}}}.$$

By setting $\epsilon = (1/2) \cdot \delta \cdot \gamma$, this contradicts Equation (11).

Therefore we conclude that $\mathsf{T}_{\mathsf{PV}}^i \nvdash \mathsf{LB}^i(s_1, s_2, m, n_0)$. □

References

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

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

[Ajt83] Miklós Ajtai. $\sum_1^1$-formulae on finite structures. *Ann. Pure Appl. Log.*, 24(1):1–48, 1983.

[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3):14:1–14:36, 2010.

[And85] Alexander E. Andreev. On a method for obtaining lower bounds for the complexity of individual monotone functions. *Soviet Math. Dokl*, 31(3):530–534, 1985.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *Transactions on Computation Theory* (TOCT), 1(1), 2009.

[Bey09] Olaf Beyersdorff. On the correspondence between arithmetic theories and propositional proof systems – a survey. *Mathematical Logic Quarterly*, 55(2):116–137, 2009.

[BGS75] Theodore P. Baker, John Gill, and Robert Solovay. Relativizatons of the $\mathsf{P} =? \mathsf{NP}$ Question. *SIAM J. Comput.*, 4(4):431–442, 1975.

[BKKK20] Sam Buss, Valentine Kabanets, Antonina Kolokolova, and Michal Koucký. Expander construction in VNC^1. *Ann. Pure Appl. Log.*, 171(7):102796, 2020.

[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.

[BKT14] Samuel R. Buss, Leszek A. Kołodziejczyk, and Neil Thapen. Fragments of approximate counting. *J. Symb. Log.*, 79(2):496–525, 2014.

[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.

[Bus86] Samuel R. Buss. *Bounded Arithmetic*. Bibliopolis, 1986.

[Bus95] Samuel R. Buss. Relating the bounded arithmetic and polynomial time hierarchies. *Ann. Pure Appl. Log.*, 75(1-2):67–77, 1995.

[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.

[Bus08] Samuel R. Buss. Bounded arithmetic, cryptography and complexity. *Theoria*, 63:147–167, 2008.

[CHO$^+$22] Lijie Chen, Shuichi Hirahara, Igor C. Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. Beyond natural proofs: Hardness magnification and locality. *J. ACM*, 69(4):25:1–25:49, 2022.

[CJW19] Lijie Chen, Ce Jin, and Ryan Williams. Hardness magnification for all sparse NP languages. In *Symposium on Foundations of Computer Science* (FOCS), pages 1240–1255, 2019.

[CK07] Stephen A. Cook and Jan Krajíček. Consequences of the provability of $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$. *J. Symb. Log.*, 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.

[CN10] Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010.

[Cob65] Alan Cobham. The intrinsic computational difficulty of functions. *Proc. Logic, Methodology and Philosophy of Science*, pages 24–30, 1965.

[Coo75] Stephen A. Cook. Feasibly constructive proofs and the propositional calculus (preliminary version). In *Symposium on Theory of Computing* (STOC), pages 83–97, 1975.

[CT06] Stephen A. Cook and Neil Thapen. The strength of replacement in weak arithmetic. *ACM Trans. Comput. Log.*, 7(4):749–764, 2006.

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-$3n$ lower bound for the circuit complexity of an explicit function. In *Symposium on Foundations of Computer Science* (FOCS), pages 89–98, 2016.

[FLY22] Zhiyuan Fan, Jiatu Li, and Tianqi Yang. The exact complexity of pseudorandom functions and the black-box natural proof barrier for bootstrapping results in computational complexity. In *Symposium on Theory of Computing* (STOC), pages 962–975, 2022.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Syst. Theory*, 17(1):13–27, 1984.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Symposium on Theory of Computing* (STOC), pages 6–20, 1986.

[HVV06] Alexander Healy, Salil P. Vadhan, and Emanuele Viola. Using nondeterminism to amplify hardness. *SIAM J. Comput.*, 35(4):903–931, 2006.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *Symposium on Foundations of Computer Science* (FOCS), pages 538–545. IEEE Computer Society, 1995.

[Jeř04] Emil Jeřábek. Dual weak pigeonhole principle, boolean complexity, and derandomization. *Ann. Pure Appl. Log.*, 129(1-3):1–37, 2004.

[Jeř05] Emil Jeřábek. *Weak pigeonhole principle and randomized computation*. PhD thesis, 2005.

[Jeř06] Emil Jeřábek. The strength of sharply bounded induction. *Mathematical Logic Quarterly*, 52(6):613–624, 2006.

[Jeř07a] Emil Jeřábek. Approximate counting in bounded arithmetic. *J. Symb. Log.*, 72(3):959–993, 2007.

[Jeř07b] Emil Jeřábek. On independence of variants of the weak pigeonhole principle. *J. Log. Comput.*, 17(3):587–604, 2007.

[Jer22] Emil Jerábek. Iterated multiplication in VTC^0. *Archive for Mathematical Logic*, pages 1–63, 2022.

[KH82] Clement F. Kent and Bernard R. Hodgson. An arithmetical characterization of NP. *Theor. Comput. Sci.*, 21:255–267, 1982.

[KKMP21] Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos H. Papadimitriou. Total functions in the polynomial hierarchy. In *Innovations in Theoretical Computer Science Conference* (ITCS), pages 44:1–44:18, 2021.

[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.

[Kor21] Oliver Korten. The hardest explicit construction. In *Symposium on Foundations of Computer Science* (FOCS), pages 433–444, 2021.

[KP98] Jan Krajícek and Pavel Pudlák. Some consequences of cryptographical conjectures for S^1_2 and EF. *Inf. Comput.*, 140(1):82–94, 1998.

[KPT91] Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti. Bounded arithmetic and the polynomial hierarchy. *Ann. Pure Appl. Log.*, 52(1-2):143–153, 1991.

[Kra92] Jan Krajíček. No counter-example interpretation and interactive computation. In Yiannis N. Moschovakis, editor, *Logic from Computer Science*, pages 287–293, New York, NY, 1992. Springer New York.

[Kra95] Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995.

[Kra01] Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 1(170):123–140, 2001.

[Kra11] Jan Krajícek. On the proof complexity of the Nisan-Wigderson generator based on a hard NP $\cap$ coNP function. *J. Math. Log.*, 11(1), 2011.

[Kra19] Jan Krajíček. *Proof Complexity*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019.

[Kra21] Jan Krajícek. Small circuits and dual weak PHP in the universal theory of p-time algorithms. *ACM Trans. Comput. Log.*, 22(2):11:1–11:4, 2021.

[LC11] Dai Tri Man Le and Stephen A. Cook. Formalizing randomized matching algorithms. *Log. Methods Comput. Sci.*, 8(3), 2011.

[LY22] Jiatu Li and Tianqi Yang. $3.1n$ - $o(n)$ circuit lower bounds for explicit functions. In *Symposium on Theory of Computing* (STOC), pages 1180–1193, 2022.

[Lê14] Dai Tri Man Lê. *Bounded Arithmetic and Formalizing Probabilistic Proofs*. PhD thesis, 2014.

[McK10] Richard McKinley. A sequent calculus demonstration of Herbrand's theorem. *arXiv preprint arXiv:1007.3414*, 2010.

[MP20] Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Ann. Pure Appl. Log.*, 171(2), 2020.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[Oja04] Kerry Ojakian. *Combinatorics in Bounded Arithmetic*. PhD thesis, 2004.

[OS18] Igor C. Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In *Symposium on Foundations of Computer Science* (FOCS), pages 65–76, 2018.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

[Pic14] Ján Pich. *Complexity Theory in Feasible Mathematics*. PhD thesis, 2014.

[Pic15a] Ján Pich. Circuit lower bounds in bounded arithmetics. *Ann. Pure Appl. Log.*, 166(1):29–45, 2015.

[Pic15b] Ján Pich. Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic. *Log. Methods Comput. Sci.*, 11(2), 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), 2021.

[Pud92] Pavel Pudlák. Some relations between subsystems of arithmetic and the complexity theory. In *Proc. Conf. Logic from Computer Science*, pages 499–519. Springer-Verlag, 1992.

[Pud06] Pavel Pudlák. Consistency and games - in search of new combinatorial principles. In V. Stoltenberg-Hansen and J. Väänänen, editors, *Logic Colloquium '03*, volume 24 of *Lecture Notes in Logic*, pages 244–281. ASL, 2006.

[Raz85] Alexander A. Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Doklady Akademii Nauk SSSR*, 281:798–801, 1985. English translation in: Soviet Mathematics Doklady 31:354–357, 1985.

[Raz87] Alexander A. Razborov. Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(4):598–607, 1987.

[Raz95a] Alexander A. Razborov. Bounded arithmetic and lower bounds in boolean complexity. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 344—-386. Birkhäuser, 1995.

[Raz95b] 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. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

[RSW22] Hanlin Ren, Rahul Santhanam, and Zhikun Wang. On the range avoidance problem for circuits. In *Symposium on Foundations of Computer Science* (FOCS), 2022.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Symposium on Theory of Computing* (STOC), pages 77–82, 1987.

[Sto76] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):1–22, 1976.

[TC21] Iddo Tzameret and Stephen A. Cook. Uniform, integral, and feasible proofs for the determinant identities. *J. ACM*, 68(2):12:1–12:80, 2021.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Wil14] Ryan Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2:1–2:32, 2014.

[Wra76] Celia Wrathall. Complete sets and the polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):23–33, 1976.

[Zam96] Domenico Zambella. Notes on polynomially bounded arithmetic. *J. Symb. Log.*, 61(3):942–966, 1996.

A Provability in $\mathsf{T}^i_{\mathsf{PV}}$

In this section, we further elaborate on the strength of the theories $\mathsf{T}^i_{\mathsf{PV}}$. Similarly to the relation between the complexity classes P, NP, and the different levels of PH, it is currently open if the theories $\mathsf{T}^i_{\mathsf{PV}}$ form a proper hierarchy, i.e., if $\mathsf{T}^j_{\mathsf{PV}}$ can prove more sentences than $\mathsf{T}^i_{\mathsf{PV}}$ when $j > i$. However, as explained in this section, this is the case under standard computational hardness assumptions. Conversely, separating the theories would lead to new complexity class separations.

In Section A.1, we show that $\mathsf{T}^i_{\mathsf{PV}}$ proves every true $\forall\Sigma^b_{i-1}$-sentence extended with sharply bounded quantifiers.

In Sections A.2 and A.3, we relate the relative strength of these theories to the hierarchy of total functions and to the polynomial time hierarchy, respectively. The results presented in these sections are closely related to results from [KPT91], which explore the strength of Buss's theories S^i_2 and T^i_2 and related questions.

In Section A.4, we exhibit a complexity lower bound statement of comparatively higher quantifier complexity that is provable in $\mathsf{T}^2_{\mathsf{PV}}$ (under a minimal assumption). In more detail, we show that if $\mathsf{NP} \nsubseteq$ (i.o.)P is true then it is provable in $\mathsf{T}^2_{\mathsf{PV}}$.

A.1 Sentences with sharply bounded quantifiers

Recall that we have defined $\mathsf{T}^i_{\mathsf{PV}}$ as the theory consisting of all true (strict) $\forall\Sigma^b_{i-1}$-sentences. In some contexts, it can also be desirable to allow sharply bounded quantifiers of the form $\forall x \leq |t|$ and $\exists x \leq |t|$ to appear arbitrarily in a Σ^b_i-formula (without increasing its quantifier complexity). It is therefore also reasonable to consider a "stronger" theory $\widetilde{\mathsf{T}}^i_{\mathsf{PV}}$ that consists of all true $\forall\Sigma^b_{i-1}$-sentences where sharply bounded quantifiers can freely appear in the axioms (see a standard reference such as [Kra95] for the formal definition of this more general class of sentences). Note that the introduction of sharply bounded quantifiers could in principle be an issue in our unprovability results, since the *replacement principle* [Bus86] that is used to manipulate sharply bounded quantifiers is unlikely to be provable in weak bounded theories [CT06].

In this subsection, we show that $\widetilde{\mathsf{T}}^i_{\mathsf{PV}} = \mathsf{T}^i_{\mathsf{PV}}$. Consequently, we can use without loss of generality strict $\forall\Sigma^b_{i-1}$-sentences when defining each theory $\mathsf{T}^i_{\mathsf{PV}}$.

Lemma A.1. *For every formula $\varphi(\vec{x})$ that contains only sharply bounded quantifiers, there is a quantifier-free formula $\hat{\varphi}(\vec{x})$ such that $\mathsf{T}^1_{\mathsf{PV}} \vdash \forall\vec{x}\,(\varphi(\vec{x}) \leftrightarrow \hat{\varphi}(\vec{x}))$.*

Proof Sketch. We use induction on the number of (sharply bounded) quantifiers in $\varphi(\vec{x})$. By replacing sharply bounded quantifiers with polynomial time functions that enumerate over their domains, we can reduce the number of sharply bounded quantifiers while maintaining the equivalence over $\mathsf{T}^1_{\mathsf{PV}}$. We omit the details. □

Lemma A.2. *For every $i \geq 1$, if $\exists y \leq t\ \phi$ is a Σ^b_i-formula without sharply bounded quantifiers, then there exists a Π^b_{i-1}-formula ϕ' without sharply bounded quantifiers and a $\mathcal{L}_{\mathsf{PV}}$-term s such that $\mathsf{T}^1_{\mathsf{PV}} \vdash \exists y \leq t\ \phi \leftrightarrow \exists z \leq s\ \phi'$.*

Similarly, if $\forall y \leq t\ \phi$ is a Π^b_i-formula without sharply bounded quantifiers, then there exists a Σ^b_{i-1}-formula ϕ' without sharply bounded quantifiers and a $\mathcal{L}_{\mathsf{PV}}$-term s such that $\mathsf{T}^1_{\mathsf{PV}} \vdash \forall y \leq t\ \phi \leftrightarrow \forall z \leq s\ \phi'$

Proof Sketch. We can make the upper bound s sufficiently large, merge all outermost bounded existential quantifiers of ϕ into a single existential quantifier $\exists z \leq s$ (or $\forall z \leq s$ in the other case), and use PV-definable pairing functions to simulate the original block of existential quantifiers. See, e.g., the proof of Lemma D.1 for more details. □

Lemma A.3. *Let $i \geq 2$. For every true $\forall\Sigma^b_{i-1}$-sentence φ with sharply bounded quantifiers, there is a true $\forall\Sigma^b_{i-1}$-sentence $\hat{\varphi}$ such that $\mathsf{T}^i_{\mathsf{PV}} \vdash \hat{\varphi}(x) \to \varphi(x)$, where no sharply bounded quantifier in $\hat{\varphi}$ appears outside of a (non-sharply) bounded quantifier.*

Proof. By applying prenexification rules, we can obtain a $\forall\Sigma^b_{i-1}$-sentence φ' that is logically equivalent to φ. We will also assume without loss of generality that φ' is in negation normal form. A pair of quantifiers (Q_1, Q_2) in φ' is said to be a *bad pair* if Q_1 is a sharply bounded quantifier, Q_2 is a (non-sharply) bounded quantifier, and Q_1 quantifies over a subformula containing Q_2. We prove the lemma by induction on the number of bad pairs within φ'. If there is no bad pair, we simply let $\hat{\varphi} = \varphi'$ and the lemma follows.

Now we assume that φ' contains $\ell \geq 1$ bad pairs. Fix a bad pair (Q_1, Q_2) such that Q_2 is innermost and Q_2 is outermost, so that there are no other quantifiers in between and there is no bad pair within the formula quantified by Q_2. By Lemma A.1, we can further assume without loss of generality that there is no sharply bounded quantifier within the formula quantified by Q_2. Consider $Q_1 x \leq |t|\ Q_2 y \leq s\ \phi$ as a subformula of φ'. If $Q_1 = Q_2 = \forall$ or $Q_1 = Q_2 = \exists$, we can simply exchange them to obtain a logically equivalent sentence with $\ell - 1$ bad pairs, which completes the proof via the induction hypothesis.

Case 1. Assume that $Q_1 = \forall$ and $Q_2 = \exists$. By the replacement axiom (see, e.g., [Bus86]), there are $\mathcal{L}_{\mathsf{PV}}$ terms α, β and a quantifier-free formula γ such that

$$\big(\forall x \leq |t|\ \exists y \leq s\ \phi\big) \leftrightarrow \big(\exists w \leq \alpha(s,t)\ \forall x \leq |t|\ (\phi(y/\beta(x,w)) \land \gamma(x,w))\big) \tag{12}$$

holds in the standard model. (Note that ϕ may have free variables $\vec{v}$ other than x, y.) Since φ' is a $\forall\Sigma^b_{i-1}$-sentence containing $\forall x \leq |t|\ \exists y \leq s\ \phi$ as a subformula, we can see that $\exists y \leq s\ \phi$ must be a Σ^b_{i-1}-formula. By Lemma A.2, we can assume without loss of generality that ϕ is a Π^b_{i-2}-formula. Then

$$\begin{aligned}\Psi &\triangleq \forall\vec{v}\,\Big(\big(\exists w \le \alpha(s,t)\,\forall x \le |t|\,(\phi(y/\beta(x,w)) \wedge \gamma(x,w))\big) \to \big(\forall x \le |t|\,\exists y \le s\,\phi\big)\Big)\\ &\Leftrightarrow \forall\vec{v}\,\Big(\big(\forall w \le \alpha(s,t)\,\exists x \le |t|\,(\neg\phi(y/\beta(x,w)) \vee \neg\gamma(x,w))\big) \vee \big(\forall x \le |t|\,\exists y \le s\,\phi\big)\Big)\end{aligned}$$

is a true $\forall\Sigma^b_{i-1}$-sentence (where $\Leftrightarrow$ is in the meta-language and denotes logical equivalence). Moreover, since ϕ contains no sharply bounded quantifiers, we know that Ψ is a $\forall\Sigma^b_{i-1}$-sentence even if we treat sharply bounded quantifiers simply as bounded quantifiers. Therefore $\mathsf{T}^i_{\mathsf{PV}} \vdash \Psi$.

Let φ'' be the $\forall\Sigma^b_{i-1}$-sentence (with sharply bounded quantifiers) obtained from φ' by replacing the LHS of Equation (12) with the RHS. Since $\mathsf{T}^i_{\mathsf{PV}} \vdash \Psi$, we know that $\mathsf{T}^i_{\mathsf{PV}} \vdash \varphi'' \to \varphi'$ as φ' is in negation normal form. Moreover, φ'' has $\ell - 1$ bad pairs. This completes the proof by the induction hypothesis.

Case 2. Assume that $Q_1 = \exists$ and $Q_2 = \forall$. Again, by the replacement axiom, we know that

$$\big(\exists x \le |t|\,\forall y \le s\,\phi\big) \leftrightarrow \big(\forall w \le \alpha(s,t)\,\exists x \le |t|\,(\phi(y/\beta(x,w)) \vee \gamma(x,w))\big) \tag{13}$$

is true in the standard model, where α, β are $\mathcal{L}_{\mathsf{PV}}$ terms γ is a quantifier-formula. Since φ' is a $\forall\Sigma^b_{i-1}$-sentence containing $\exists x \le |t|\,\forall y \le s\,\phi$ as a subformula, we get that $\forall y \le s\,\phi$ is a Π^b_{i-2}-formula and $i > 2$. By Lemma A.2, we can assume without loss of generality that ϕ is a Σ^b_{i-3}-formula. Then

$$\begin{aligned}\Psi &\triangleq \forall\vec{v}\,\Big(\big(\forall w \le \alpha(s,t)\,\exists x \le |t|\,(\phi(y/\beta(x,w)) \vee \gamma(x,w))\big) \to \big(\exists x \le |t|\,\forall y \le s\,\phi\big)\Big)\\ &\Leftrightarrow \forall\vec{v}\,\Big(\big(\exists w \le \alpha(s,t)\,\forall x \le |t|\,(\neg\phi(y/\beta(x,w)) \wedge \neg\gamma(x,w))\big) \vee \big(\exists x \le |t|\,\forall y \le s\,\phi\big)\Big)\end{aligned}$$

is a true $\forall\Sigma^b_{i-1}$-sentence even if we treat sharply bounded quantifiers as bounded quantifiers. Therefore $\mathsf{T}^i_{\mathsf{PV}} \vdash \Psi$. Then we can resolve this case as in Case 1. □

Theorem A.4. *For every $i \ge 1$, $\mathsf{T}^i_{\mathsf{PV}}$ proves every true $\forall\Sigma^b_{i-1}$-sentence even if sharply bounded quantifiers are allowed to appear arbitrarily in the sentence. In other words, $\widetilde{\mathsf{T}}^i_{\mathsf{PV}} = \mathsf{T}^i_{\mathsf{PV}}$.*

Proof. If $i = 1$, the result immediately follows from Lemma A.1. For $i \ge 2$, we can first move sharply bounded quantifiers via Lemma A.3 so they only appear as innermost quantifiers. We can then remove the sharply bounded quantifiers using Lemma A.1. □

A.2 Strength of $\mathsf{T}^i_{\mathsf{PV}}$ and the hierarchy of total functions

In this section, we show that separating the theories $\mathsf{T}^i_{\mathsf{PV}}$ would lead to new complexity class separations. For instance, we prove that $\mathsf{T}^2_{\mathsf{PV}} = \mathsf{T}^1_{\mathsf{PV}}$ if and only if the search problem of every TFNP relation can be solved in polynomial time. A related result holds for every $i \ge 1$ (see Theorem A.8 below for the precise statement).

The relationship between these theories and the corresponding complexity collapses provides evidence that the theories $\mathsf{T}^i_{\mathsf{PV}}$ form a strict hierarchy. However, it also shows that unconditionally establishing that this is the case will be quite difficult.

For convenience, in the statements below we identify $\mathsf{TF}\Sigma^p_0$ with FP. We refer to Section 2.3 for definitions and to [KKMP21] for more information about total functions in the polynomial hierarchy. Abusing notation, in the statements below we view $\mathsf{TF}\Sigma^p_i$ as a class of search problems, i.e., given x the goal is to find y such that $R(x, y)$ holds, where $R \in \mathsf{TF}\Sigma^p_i$.

We will need the following lemmas, which can be proved using standard techniques from complexity and logic.

Lemma A.5. *For every $i \geq 1$, $\mathsf{P}^{\mathsf{TF}\Sigma^p_{i-1}} \subseteq \Sigma^p_{i-1} \subseteq \mathsf{P}^{\mathsf{TF}\Sigma^p_i}$.*

Lemma A.6. *For every $i \geq 1$, $\Sigma^p_i \subseteq \Sigma^p_{i-1}$ if and only if $\Sigma^p_i \subseteq \mathsf{P}^{\mathsf{TF}\Sigma^p_{i-1}}$.*

Lemma A.7. *Let $i \geq 1$, $t(x)$ be an $\mathcal{L}_{\mathsf{PV}}$ term, and $\phi(x, y)$ be a $\Pi^b_{i-1}(\mathcal{L}_{\mathsf{PV}})$-formula. If $\mathsf{T}^i_{\mathsf{PV}} \vdash \forall x\, \exists y \leq t(x)\, \phi(x, y)$, then there exists a $\mathsf{FP}^{\Sigma^p_{i-1}}$ algorithm $A(x)$ such that for every $x \in \mathbb{N}$, $\phi^{\mathbb{N}}(x, A(x))$ holds.*

The next theorem relates the relative strength of theories $\mathsf{T}^i_{\mathsf{PV}}$ to the computational complexity of the search problems associated with the relations in $\mathsf{TF}\Sigma^p_j$.

Theorem A.8. *For every $i \geq 1$, the following propositions hold:*

(i) If $\mathsf{TF}\Sigma^p_i \subseteq \mathsf{FP}^{\mathsf{TF}\Sigma^p_{i-1}}$, then $\mathsf{T}^i_{\mathsf{PV}} \equiv \mathsf{T}^{i+1}_{\mathsf{PV}}$.

(ii) If $\mathsf{T}^i_{\mathsf{PV}} \equiv \mathsf{T}^{i+1}_{\mathsf{PV}}$, then $\mathsf{TF}\Sigma^p_i \subseteq \mathsf{FP}^{\Sigma^p_{i-1}}$.

In particular, $\mathsf{TFNP} = \mathsf{FP}$ if and only if $\mathsf{T}^1_{\mathsf{PV}} \equiv \mathsf{T}^2_{\mathsf{PV}}$.

Proof. (1) Assume that $\mathsf{TF}\Sigma^p_i \subseteq \mathsf{FP}^{\mathsf{TF}\Sigma^p_{i-1}}$. We need to show that for every $\varphi \in \mathsf{T}^{i+1}_{\mathsf{PV}}$, $\mathsf{T}^i_{\mathsf{PV}} \vdash \varphi$. Since it is enough to prove this for the axioms of $\mathsf{T}^{i+1}_{\mathsf{PV}}$, we can assume without loss of generality that $\varphi = \forall x\, \exists y \leq t(x)\, \phi(x, y)$ for some Π^b_{i-1}-formula ϕ. Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be the search problem such that $(x, y) \in R$ if and only if $y \leq t(x)$ and $\phi^{\mathbb{N}}(x, y)$. Using the assumption in Item (*i*), we get that this search problem can be solved in $\mathsf{FP}^{\mathsf{TF}\Sigma^p_{i-1}}$. In particular, there is a Σ^b_{i-1}-formula $\beta(x, y)$ that is total over $\mathbb{N}$ and only accepts a pair (x, y) if $(x, y) \in R$. Thus $\mathsf{T}^i_{\mathsf{PV}} \vdash \forall x\, \exists y \leq t(x)\, \beta(x, y)$ and $\mathsf{T}^i_{\mathsf{PV}} \vdash \forall x\, \forall y \leq t(x)\, (\beta(x, y) \to \phi(x, y))$ by counting the quantifier complexity of these two sentences. It then follows that $\mathsf{T}^i_{\mathsf{PV}} \vdash \varphi$.

(2) Assume that $\mathsf{T}^{i+1}_{\mathsf{PV}} \equiv \mathsf{T}^i_{\mathsf{PV}}$. Let $R \in \mathsf{TF}\Sigma^p_i$ be a total relation such that for every $(x, y) \in R$, $|y| \leq |x|^c$. Let $\beta(x, y)$ be a Π^b_{i-1}-formula that captures over the standard model that $(x, y) \in R$. Then $\mathsf{T}^{i+1}_{\mathsf{PV}} \vdash \forall x\, \exists y \in \{0,1\}^{|x|^c}\, \beta(x, y)$, which further means by the assumption in Item (*ii*) that $\mathsf{T}^i_{\mathsf{PV}} \vdash \forall x\, \exists y \in \{0,1\}^{|x|^c}\, \beta(x, y)$. By Lemma A.7, we get that the search problem corresponding to R can be solved in $\mathsf{FP}^{\Sigma^p_{i-1}}$. □

A.3 Strength of $\mathsf{T}^i_{\mathsf{PV}}$ and the polynomial hierarchy

In this section, we relate the collapse of theories $\mathsf{T}^i_{\mathsf{PV}}$ to a collapse of the polynomial hierarchy. More precisely, we show that if $\mathsf{T}^{i+2}_{\mathsf{PV}} = \mathsf{T}^i_{\mathsf{PV}}$ then $\Sigma^p_{i+1} = \Pi^p_{i+1}$. Consequently, under the widely believed assumption that PH does not collapse, the theories $\mathsf{T}^i_{\mathsf{PV}}$ can prove more sentences as i increases.

We will need a technical lemma from [KPT91] employed there to relate a certain collapse in Buss's hierarchy of theories of bounded arithmetic to a corresponding collapse of the polynomial hierarchy. First, we review the following statement.

Principle $\mathbf{\Omega}(i)$. There is a constant $k \in \mathbb{N}$ such that the following holds. For every relation $P(x, y) \in \Pi^p_i$, there are $\mathsf{FP}^{\Sigma^p_i}$ functions $f_1(a), f_2(a, b_1), \ldots, f_k(a, b_1, \ldots, b_{k-1})$ such that:

- Either $\forall z\; P^*(a, f_1(a), z)$ is true, or for every b_1 s.t. $\neg P^*(a, f_1(a), b_1)$, it holds that:
- Either $\forall z\; P^*(a, f_2(a, b_1), z)$ is true, or for every b_2 s.t. $\neg P^*(a, f_2(a, b_1), b_2)$, it holds that:

- Either $\forall z\ P^*(a, f_3(a, b_1, b_2), z)$ is true, or ...
- ...
- $\forall z\ P^*(a, f_k(a, b_1, b_2, \ldots, b_k), z)$ is true;

where $P^*(x, y, z) \triangleq |y| \le |x| \wedge (y = 0 \vee P(x, y)) \wedge (|y| < |z| \le |x| \to \neg P(x, z))$.

Lemma A.9 ([KPT91], Lemma 2.2)**.** *For every $i \ge 0$, if Principle $\Omega(i)$ is true, then $\Sigma^p_{i+1} \subseteq \mathsf{P}^{\Sigma^p_i}_{/\mathsf{poly}}$ and thus also $\Sigma^p_{i+2} = \Pi^p_{i+2}$.*

Theorem A.10. *For every $i \ge 1$, if $\mathsf{T}^i_{\mathsf{PV}} \equiv \mathsf{T}^{i+2}_{\mathsf{PV}}$, then $\Sigma^p_i \subseteq \mathsf{P}^{\Sigma^p_{i-1}}_{/\mathsf{poly}}$ and thus also $\Sigma^p_{i+1} = \Pi^p_{i+1}$.*

Proof. By Lemma A.9, it suffices to show that $\mathsf{T}^i_{\mathsf{PV}} \equiv \mathsf{T}^{i+2}_{\mathsf{PV}}$ implies Principle $\Omega(i-1)$, for each $i \ge 1$. Assume that $\mathsf{T}^i_{\mathsf{PV}} \equiv \mathsf{T}^{i+2}_{\mathsf{PV}}$. For every relation $P(x, y) \in \Pi^p_{i-1}$, consider the $\Pi^b_{i-1}(\mathcal{L}_{\mathsf{PV}})$-formula $\alpha(x, y)$ that defines it and let $\alpha^*(x, y, z)$ be defined as

$$\alpha^*(x, y, z) \triangleq |y| \le |x| \wedge (y = 0 \vee \alpha(x, y)) \wedge (|y| < |z| \le |x| \to \neg\alpha(x, z)).$$

Let $\varphi \triangleq \forall x\ \exists |y| \le |x|\ \forall |z| \le |x|\ \alpha^*(x, y, z)$. Since $\mathbb{N} \vDash \varphi$ and φ is a $\forall\Sigma^b_{i+1}$ sentence, we obtain that $\mathsf{T}^{i+2}_{\mathsf{PV}} \vdash \varphi$ and thus $\mathsf{T}^i_{\mathsf{PV}} \vdash \varphi$ by the assumption that $\mathsf{T}^i_{\mathsf{PV}} \equiv \mathsf{T}^{i+2}_{\mathsf{PV}}$. By Theorem 2.14, it follows that $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi$. Moreover, by Lemma 2.13, we know that

$$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall x\ \exists y\ \forall z\ |y| \le |x| \wedge (y = 0 \vee f_\alpha(x, y)) \wedge (|y| < |z| \le |x| \to \neg f_\alpha(x, z)),$$

where $f^{\mathbb{N}}_\alpha(x, y)$ is exactly $P(x, y)$. Principle $\Omega(i-1)$ then follows directly from the KPT Witnessing Theorem (Theorem 2.9) and Theorem 2.17. $\square$

A.4 On the provability of $\mathsf{NP} \not\subseteq$ (i.o.)P

Recall that the axioms of $\mathsf{T}^2_{\mathsf{PV}}$ consist of all true $\forall\Sigma^b_1$-sentences in the language $\mathcal{L}_{\mathsf{PV}}$. In this section, we give a simple example of a complexity lower bound encoded by a collection of $\forall\Sigma^b_2(\mathcal{L}_{\mathsf{PV}})$-sentences provable in $\mathsf{T}^2_{\mathsf{PV}}$, assuming the lower bound holds. (Note that the formalization below uses $n \in \mathsf{Log}$, while our unprovability results hold even for $n \in \mathsf{LogLog}$.)

Theorem A.11. *Assume that $\mathsf{NP} \not\subseteq$ (i.o.)P. For every polynomial-time Turing machine A, there is a constant $n_0 \in \mathbb{N}$ such that $\mathsf{T}^2_{\mathsf{PV}}$ proves*

$$\mathsf{Fail}(A) \triangleq \forall n \in \mathsf{Log}\ \exists \varphi(x_1, \ldots, x_m) \in \{0,1\}^n \left(n > n_0 \to \mathsf{Error}(A, \varphi)\right),$$

where φ is an 3-CNF formula, and

$$\mathsf{Error}(A, \varphi) \triangleq (\exists x \in \{0,1\}^m\ \varphi(x) = 1 \wedge A(\varphi) = 0) \vee (\forall x \in \{0,1\}^m\ \varphi(x) = 0 \wedge A(\varphi) = 1).$$

Proof. Assume that $\mathsf{NP} \not\subseteq$ (i.o.)P. Then $\mathsf{3SAT} \not\subseteq$ (i.o.)P, which means that for every polynomial-time Turing machine A, there exists a constant n_0 such that A does not solve 3SAT on instances of length $n > n_0$. Let A be an arbitrary polynomial-time Turing machine. We would like to show that $\mathsf{T}^2_{\mathsf{PV}} \vdash \mathsf{Fail}(A)$.

```
Input : A string $\varphi \in \{0,1\}^n$ encoding a 3-CNF formula.
Let $\varphi_1(x_1, x_2, \ldots, x_m)$ be $\varphi$;
Let $z \in \{0,1\}^m$ be a string to be determined;
If $A(\varphi_1) = 0$, return $(0, 0)$;
for $i = 1, 2, \ldots, m$ do
    if $A(\varphi_i(x_i/0)) = 1$ then
        Let $z_i = 0$ and $\varphi_{i+1} = \varphi_i(x_i/0)$;
    else if $A(\varphi_i(x_i/1)) = 1$ then
        Let $z_i = 1$ and $\varphi_{i+1} = \varphi_i(x_i/1)$;
    else
        return $(1, \varphi_i)$;
    end
end
return $(2, z)$;
```

Algorithm 11: Search-SAT Algorithm S

Search-to-Decision Reduction. We firstly use a standard search-to-decision reduction to construct an efficient algorithm S that searches for a satisfying assignment, assuming that A solves SAT. An explicit description of S appears below (see Algorithm 11: Search-SAT Algorithm S).

Without loss of generality, we assume that for every $\varphi = \varphi_1 \in \{0,1\}^n$ and $i \in [m]$, the corresponding formulas $\varphi_i(x_i/0)$ and $\varphi_i(x_i/1)$ can also be encoded as n-bit strings. Let A' be the following polynomial-time algorithm: given an instance $\varphi \in \{0,1\}^n$ encoding a 3-CNF formula; run $S(\varphi) = (b, z)$; accept if and only if $b = 2$ and $\varphi(z) = 1$.

Claim A.12. *There is a constant* $n_1 \in \mathbb{N}$ *such that* $\mathsf{T}^2_{\mathsf{PV}} \vdash \forall n \in \mathsf{Log}\ \exists \varphi(x_1, \ldots, x_m) \in \{0,1\}^n\ \exists x \in \{0,1\}^m\ (n > n_1 \to \varphi(x) = 1 \wedge A'(\varphi) = 0)$.

Proof. By the definition of A' we can see that it has only one-sided error, i.e., for every φ such that $A'(\varphi) = 1$, φ is satisfiable. Since 3SAT $\not\subseteq$ (i.o.)P, the sentence is a $\forall\Sigma^b_1$-sentence that is true in the standard model provided that n_1 is large enough, which further means that it is provable in $\mathsf{T}^2_{\mathsf{PV}}$. □

Claim A.13. *We have that* $\mathsf{T}^2_{\mathsf{PV}} \vdash \forall n \in \mathsf{Log}\ \forall \varphi(x_1, \ldots, x_m) \in \{0,1\}^n\ (A(\varphi) = 1 \wedge A'(\varphi) = 0 \to \exists \varphi' \in \{0,1\}^n\ \mathsf{Error}(A, \varphi'))$.

Proof. Indeed, it is possible to establish even in PV that if $\neg\mathsf{Error}(A, \varphi')$ holds for every $\varphi' \in \{0,1\}^n$ then the search-to-decision reduction works as desired and consequently $\neg(A(\varphi) = 1 \wedge A'(\varphi) = 0)$. We omit the details. □

Provability of the Hardness of 3SAT**.** Now we prove in $\mathsf{T}^2_{\mathsf{PV}}$ that $\mathsf{Fail}(A)$ holds for $n_0 \triangleq n_1$, where $n_1 \in \mathbb{N}$ is the constant in Claim A.12. Arguing in the theory, let $n \in \mathsf{Log}$ be larger than n_1. Towards a contradiction, assume that for every $\varphi(x_1, \ldots, x_m) \in \{0,1\}^n$, $\neg\mathsf{Error}(A, \varphi)$. Let $\varphi(x_1, \ldots, x_m) \in \{0,1\}^n$ be a 3-CNF formula from Claim A.12 such that $\exists x \in \{0,1\}^m\ (n > n_1 \to \varphi(x) = 1 \wedge A'(\varphi) = 0)$. Since φ is satisfiable and by assumption $\neg\mathsf{Error}(A, \varphi)$, we get that $A(\varphi) = 1$. Consequently, we have both $A(\varphi) = 1 \wedge A'(\varphi) = 0$. In turn, Claim A.13 yields the existence of $\varphi' \in \{0,1\}^n$ such that $\mathsf{Error}(A, \varphi')$. This is in contradiction to the initial assumption on the correctness of A' on all instances of length n. □

B Proofs of the Witnessing Theorems

In this section, we present some omitted proofs for the witnessing theorems discussed in Section 3.

B.1 Proof of Theorem 3.1 via Herbrand's Theorem

We now demonstrate a proof of the witnessing theorem using Herbrand's Theorem.[26] The latter appears in different forms in the literature; in this section, we refer to the exposition in [Bus98].[27]

We start by clarifying some basic definitions. We work with connectives and quantifiers $\{\forall, \exists, \wedge, \vee, \neg\}$ and define other connectives from them. We always assume that first-order sentences are written in *negation normal form*, i.e., negations are placed only over atoms. A formula is said to be in *prenex normal form* if it can be written as $Q_1x_1\ Q_2x_2\ \ldots\ Q_kx_k\ P$, where $Q_1, \ldots, Q_k \in \{\forall, \exists\}$ and P is quantifier-free. We identify formulas that differ only by a renaming of bounded variables.

Definition B.1. Let $\varphi(x)$ be a formula. A *prenexification* of $\varphi(x)$ is a formula in prenex normal form obtained by successive applications of the following operations $Qx\ \phi \star \psi \mapsto Qx\ (\phi \star \psi)$ and $\phi \star Qx\ \psi \mapsto Qx\ (\phi \star \psi)$, where $Q \in \{\forall, \exists\}$ and $\star \in \{\wedge, \vee\}$. As usual, variables are renamed whenever necessary.

Definition B.2. An $\vee$*-expansion* of a formula φ is any formula obtained from φ through applications of the following operation:

> If ψ is a subformula of an $\vee$-expansion φ' of φ, replacing ψ in φ' with $\psi \vee \psi$ produces another $\vee$-expansion of φ.

A *strong* $\vee$*-expansion* of a formula φ restricts ψ to be a subformula where the outermost connective is an existential quantifier. Similarly to the previous definition, multiple applications of the rule are allowed.

Definition B.3. Let $\mathcal{T}$ be a theory and φ be a formula in prenex normal form:

$$\varphi \triangleq \forall x_1 \ldots \forall x_{n_1}\ \exists y_1\ \forall x_{n_1+1} \ldots \forall x_{n_2}\ \exists y_2 \ldots \forall x_{n_{r-1}+1} \ldots \forall x_{n_r}\ \exists y_r\ \forall x_{n_r+1} \ldots \forall x_{n_{r+1}}\ \psi(\vec{x}, \vec{y}).$$

A *witnessing substitution for* φ *over* $\mathcal{T}$ is a sequence of terms $t_1, \ldots, t_r$ such that $\mathcal{T} \vdash \forall \vec{x}\ \varphi(\vec{x}, \vec{y}/\vec{t})$, where t_i contains variables only from $x_1, \ldots, x_{n_i}$ for every $i \in [r]$.

Theorem B.4 (Herbrand's Theorem (see, e.g., [Bus98, McK10]))**.** *Let $\mathcal{T}$ be a universal theory and φ be a first-order formula. Then $\mathcal{T} \vdash \varphi$ if and only if there is a prenexification of a strong $\vee$-expansion of φ that admits a witnessing substitution over $\mathcal{T}$.*

Reminder of Theorem 3.1. *Let $\mathcal{T}$ be a universal bounded theory with vocabulary $\mathcal{L}$ that is closed under if-then-else. Let φ be a bounded $\mathcal{L}$-formula of the form*

$$\begin{aligned}\varphi(x)\ \triangleq\ & \exists y_1 \le t_1(x)\ \forall x_1 \le s_1(x, y_1)\ \exists y_2 \le t_2(x, y_1, x_1) \ldots \forall x_{k-1} \le s_{k-1}(x, y_1, x_1, \ldots, y_{k-1}) \\ & \exists y_k \le t_k(x, y_1, x_1, \ldots, y_{k-1}, x_{k-1})\ \forall x_k \le s_k(x, y_1, x_1, \ldots, y_k)\ \phi(x, x_1, \ldots, x_k, y_1, \ldots, y_k),\end{aligned}$$

where $\phi(x, \vec{x}, \vec{y})$ is a quantifier-free $\mathcal{L}$-formula. Then $\mathcal{T} \vdash \forall x\ \varphi(x)$ if and only if there is a universal winning $\mathcal{L}$-strategy of length $O(1)$ for the truthifier in the corresponding tree exploration game of $\varphi(x)$.

[26] We than an anonymous reviewer for suggesting this perspective, which simplified our previous presentation relying on a direct analysis based on a sequent calculus. J. Krajíček has also proposed a simplification of the original proof via Gentzen's Midsequent Theorem that we do not explore here.

[27] See also [McK10] for a correction in the proof of Herbrand's Theorem presented in [Bus98].

The "if" direction of the theorem is simpler. Assume that $\mathcal{T} \nvdash \forall x\ \varphi(x)$. Then by the completeness theorem there is a model $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ and $n_0 \in \mathcal{D}$ such that $\varphi^{\mathcal{M}}(n_0)$ is false, which further means that there is a winning strategy of the falsifier in the evaluation game of $\varphi(x)$ on the broad $(\mathcal{M}, n_0)$. Consider the strategy of the falsifier in the tree exploration game that simply simulates this winning strategy, i.e., after the truthifier adds a node and specifies an element on the edge, the falsifier treats the path from the root to this node as a partial transcript of the evaluation game and chooses an element according to the strategy of the evaluation game. It is clear that the truthifier cannot reach a winning node, thus it does not have a universal winning $\mathcal{L}$-strategy of the tree exploration game.

In the rest of this sub-section, we prove the "only if" direction of the theorem, that is to extract a winning strategy from $\mathcal{T} \vdash \forall x\ \varphi(x)$.

Step 1: Unbounded Tree Exploration Game. Due to technical reasons, we need to define unbounded variants of the tree exploration games.

Let $\varphi(x)$ be a Σ_k^b-formula in prenex normal form (with bounded quantifiers). We define the following translation $[\cdot]_{\mathsf{imp}}$ that transforms a bounded formula in prenex normal form into a logically equivalent formula with only unbounded quantifiers:

- If $\varphi(\vec{x})$ is quantifier free, $[\varphi(\vec{x})]_{\mathsf{imp}} \triangleq \varphi(\vec{x})$.
- If $\varphi(\vec{x}) = \forall y \leq t(\vec{x})\ \phi(\vec{x}, y)$ and $[\phi]_{\mathsf{imp}} = Q_1 z_1\ Q_2 z_2 \dots Q_k z_k\ \alpha(\vec{x}, y, \vec{z})$, where $Q_i \in \{\forall, \exists\}$ for $i \in [k]$ and α is quantifier-free, then $[\varphi(\vec{x})]_{\mathsf{imp}} \triangleq \forall y\ Q_1 z_1\ Q_2 z_2 \dots Q_k z_k (\neg(y \leq t(\vec{x})) \vee \alpha(\vec{x}, y, \vec{z}))$.
- If $\varphi(\vec{x}) = \exists y \leq t(\vec{x})\ \phi(\vec{x}, y)$ and $[\phi]_{\mathsf{imp}} = Q_1 z_1\ Q_2 z_2 \dots Q_k z_k\ \alpha(\vec{x}, y, \vec{z})$, where $Q_i \in \{\forall, \exists\}$ for $i \in [k]$ and α is quantifier-free, then $[\varphi(\vec{x})]_{\mathsf{imp}} \triangleq \exists y\ Q_1 z_1\ Q_2 z_2 \dots Q_k z_k (y \leq t(\vec{x}) \wedge \alpha(\vec{x}, y, \vec{z}))$.

We say a formula φ is *implicitly bounded* if there is a bounded formula ψ in prenex normal form such that $\varphi = [\psi]_{\mathsf{imp}}$.

Let $\varphi(x) = \exists y_1\ \forall x_1 \dots \exists y_k\ \forall x_k\ \phi(x, \vec{x}, \vec{y})$ be an implicitly bounded $\mathcal{L}$-formula as discussed above and $(\mathcal{M} = (\mathcal{D}, \mathcal{I}), n_0)$ be a board. The *unbounded tree exploration game* of φ is defined as follows. In each round, the truthifier chooses a node u on the tree (which only consists of the root at the beginning) and specifies a number $m \in \mathcal{D}$; the falsifier then specifies a number $n \in \mathcal{D}$; after this round, a child of u is added to the tree by an edge labeled (m, n). The truthifier wins if and only if there is a node on the tree such that the pairs on the path from the root to the node form a satisfying assignment of $\phi(x/n_0, \vec{x}, \vec{y})$ within $\mathcal{M}$, where the truthifier's moves are for $\vec{y}$ and the falsifier's moves are for $\vec{x}$.

An $\mathcal{L}$-strategy of the truthifier of length $\ell \in \mathbb{N}$ is a sequence

$$\tau = \langle p_1, r_1, p_2, r_2, \dots, p_\ell, r_\ell \rangle\ ,$$

where p_i is an $\mathcal{L}$-term and $r_i \in \mathbb{N}$ such that $1 \leq r_i \leq i$. Let $(\mathcal{M}, n_0)$ be a board. The game-theoretic strategy for the unbounded tree exploration game induced by τ is the following strategy:

- In the i-th move, the truthifier introduces a node numbered $i + 1$ as a child of the node r_i, and chooses the element $v_i \triangleq p_i^{\mathcal{M}}(n_0, T, \Gamma) \in \mathcal{M}$, where Γ describes the moves of previous rounds (including $v_1, \dots, v_{i-1}$ and the falsifier's moves).

A length-ℓ $\mathcal{L}$-strategy is said to be a universal winning strategy if the truthifier playing the induced game-theoretic strategy wins within ℓ moves against any strategy of the falsifier on any board $(\mathcal{M}, n_0)$.

Lemma B.5. *Let $\mathcal{T}$ be a bounded theory over the language $\mathcal{L}$ that is closed under if-then-else. If there is an $O(1)$-length $\mathcal{L}$-strategy that is a universal winning strategy of the truthifier for the unbounded tree* *exploration game of $\varphi = [\psi]_{\mathsf{imp}}$, then there is an $O(1)$-length $\mathcal{L}$-strategy that is a universal winning strategy of the truthifier for the tree exploration game of ψ.*

Proof. Assume that $\tau = \langle p_1, r_1, p_2, r_2, \ldots, p_\ell, r_\ell \rangle$ is a universal winning $\mathcal{L}$-strategy of length $\ell \in \mathbb{N}$ for the unbounded tree exploration game. Let $p_i'(x, \Gamma)$ be the term defined as follows:

(i) Parse $\Gamma = (m_1, n_1, m_2, n_2, \ldots, m_{i-1}, n_{i-1})$ as the moves in previous rounds.

(ii) Define $\hat{\Gamma}_0$ to be the empty list and $\hat{\Gamma}_{j+1}$ to be

$$\hat{\Gamma}_{j+1} = \begin{cases} \hat{\Gamma}_j; (m_{j+1}, n_{j+1}) & \text{if } m_{j+1} = p_{j+1}(n_0, \hat{\Gamma}_j) \\ \hat{\Gamma}_j; (p_{j+1}(n_0, \hat{\Gamma}_j), 0) & \text{otherwise} \end{cases}$$

(iii) Output 0 if $p_i(n_0, \hat{\Gamma}_{i-1})$ is not a valid move (i.e., it violates the inequality for the bounded variable); and output $p_i(n_0, \hat{\Gamma}_{i-1})$ otherwise.

Note that such p_i' always exists as $\mathcal{T}$ is closed under if-then-else. We now argue that the $\mathcal{L}$-quasi-strategy $\tau' \triangleq \langle p_1', r_1, p_2', r_2, \ldots, p_\ell', r_\ell \rangle$ is indeed a universal winning $\mathcal{L}$-strategy for the tree exploration game of ψ. Intuitively, τ' is the following $\mathcal{L}$-quasi-strategy: it simulates τ if it gives a valid move; otherwise, it simply outputs 0 and "forgets" the response of the falsifier, pretending that in this round it simulates τ and the falsifier's response were 0.

By the definition of p_i' it is easy to see that τ' is an $\mathcal{L}$-strategy for the tree exploration game of ψ, since it will never output an invalid move. Towards a contradiction we assume that it is not a universal winning strategy. In such case, there exist a board $(\mathcal{M}, n_0)$ and a strategy $\tau_{\mathtt{f}}'$ for the falsifier that prevents the truthifier from winning within ℓ rounds on the board against the truthifier playing the induced strategy of τ'. We now construct a strategy $\tau_{\mathtt{f}}$ of the falsifier for the unbounded tree exploration game of φ on the board $(\mathcal{M}, n_0)$ that prevents τ from winning within ℓ rounds and thus leads to a contradiction.

- Assume that the moves of $\tau_{\mathtt{f}}'$ against τ' are $n_1', n_2', \ldots, n_\ell'$. In the i-th move, if the truthifier's move is an invalid move in the (bounded) tree exploration game (i.e., it violates the inequality for the bounded variable), the falsifier chooses $n_i \triangleq 0$; otherwise the falsifier chooses $n_i \triangleq n_i'$.

It is easy to check that against this strategy of the falsifier, τ cannot win within ℓ rounds. This is because the transcript of $\tau_{\mathtt{f}}$ vs τ is exactly the lists $\hat{\Gamma}$ in the definition of τ'; and since τ' cannot win against $\tau_{\mathtt{f}}'$ within ℓ rounds, τ also cannot win against $\tau_{\mathtt{f}}$ within ℓ rounds. □

This lemma shows that to obtain a winning strategy of the tree exploration game, we only need to construct a winning strategy of the unbounded tree exploration game. In practice, this means that we do not need to treat bounded quantifiers in a special way.

Step 2: Strategy from Herbrand's Theorem. Let $\varphi(x)$ be any implicitly bounded formula of form $\varphi(x) = \exists y_1\ \forall x_1 \ldots \exists y_k\ \forall x_k\ \phi(x, \vec{x}, \vec{y})$ and $\mathcal{T}$ be a universal theory, where ϕ is quantifier-free. Assume that $\mathcal{T} \vdash \forall x\ \varphi(x)$. Then by Theorem B.4 there is a prenexification of a strong $\vee$-expansion of $\forall x\ \varphi(x)$ that admits a witnessing substitution over $\mathcal{T}$. Our goal is to extract a winning strategy for the unbounded tree exploration game of $\varphi(x)$ from the strong $\vee$-expansion, prenexification, and witnessing substitution.

Let φ^{exp} be the strong $\vee$-expansion of $\forall x\ \varphi(x)$ and φ^{pre} be a prenexification of φ^{exp}. We can see that the existential quantifiers within φ^{exp} form a tree structure with respect to the sub-formula relation. More formally: we introduce a node ε_i for each existential quantifier $\exists_i$ in φ^{exp}, and define the node ε_i to be a child of ε_j if and only if $\exists_i$ is inside $\exists_j$ within φ^{exp} and there is no other existential quantifiers in between. We introduce a root node ε_0 corresponding to the entire sentence that has all nodes without parent as children.

Let T be the tree defined above. It is easy to see that the tree has depth k and every leaf in T is in the k-th level (the root is in the 0-th level).

We can observe that for every existential quantifier $\exists_i$ in φ^{exp}, there is a universal quantifier $\forall_i$ that immediately follows it (i.e. $\forall_i$ is the outermost quantifier of the formula quantified by $\exists_i$), and this pair $(\exists_i, \forall_i)$ is a copy of an adjacent pair of quantifiers in $\varphi(x)$. Conversely, every universal quantifier (except for the outermost one) directly follows an existential quantifier. Therefore the quantifiers of φ^{exp} except for the outermost one are partitioned into disjoint pairs $(\exists_i, \forall_i)$ as defined above.

Recall that φ^{pre} is a prenexification of φ^{exp}, that is, we turn φ^{exp} into its prenex normal form by prenexification rules. Moreover, the order of existential quantifiers of φ^{pre} is essentially a traversal of T. That is, for every ε_i and ε_j in T such that ε_j is a child of ε_i, where ε_i corresponds to $\exists_i$ and ε_j corresponds to $\exists_j$, then $\exists_i$ is to the left of $\exists_j$ in φ^{pre}. In addition, for every existential quantifier $\exists_i$, its corresponding universal quantifier $\forall_i$ appears to the right of $\exists_i$. Let φ^{pn} be the sentence obtained from φ^{pre} by moving $\forall_i$ to the immediate right of φ^{pre}, for every pair $(\exists_i, \forall_i)$ of corresponding quantifiers, as defined above. We note that if φ^{pre} has a witnessing substitution, then φ^{pn} also has a witnessing substitution. (Intuitively, this is because $\exists x\, \forall y\, \phi(x, y)$ implies $\forall y\, \exists x\, \phi(x, y)$.)

Now we focus on the structure of φ^{pn}. The quantifiers of φ^{pn} (except for the outermost universal quantifier) are obtained from a traversal of T, where every universal quantifier immediate follows its existential quantifier. The quantifier-free formula within all the quantifiers is a disjunction of copies of $\phi(x, \vec{x}, \vec{y})$, where:

- x is a bounded variable quantified by the outermost universal quantifier;
- $\vec{x} = (x_1, \ldots, x_k)$ and $\vec{y} = (y_1, \ldots, y_k)$ are bounded variables quantified by universal and existential quantifiers within φ^{pn}, respectively, where y_i and x_i are quantified by a pair of corresponding pairs of existential and universal quantifiers.
- Assume that y_i, x_i are quantified by $\exists_i, \forall_i$, respectively, for every $i \in [k]$. Let ε_i be the node in T corresponding to $\exists_i$. Then $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k$ forms a path in T from the root to a leaf.
- Conversely, for every such path $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k$ corresponding to $\exists_1, \exists_2, \ldots, \exists_k$, there is a copy of $\phi(x, \vec{x}, \vec{y})$ appearing in the disjunction in φ^{pn} such that $\vec{y}$ are quantified by $\exists_1, \ldots, \exists_k$ and $\vec{x}$ are quantified by the universal quantifiers corresponding to $\exists_1, \ldots, \exists_k$.

Therefore, the paths from the root to the leaves in T corresponds to the copies of $\phi(x, \vec{x}, \vec{y})$ in the disjunction in φ^{pn}.

Now we spell out the strategy for the (unbounded) tree exploration game of $\varphi(x)$ from the tree T, φ^{pn}, and the witnessing substitution of φ^{pn}. Let

$$\varphi^{\mathsf{pn}} = \forall_0 x\, \exists_1 y_1\, \forall_1 x_1\, \exists_2 y_2\, \forall_2 x_2 \ldots \exists_d y_d\, \forall_d x_d\, \hat{\phi}, \quad \hat{\phi} \triangleq \bigvee_{i=1}^{\ell} \phi^i\ ,$$

where each ϕ^i is a copy of $\phi(x, \vec{x}, \vec{y})$ corresponding to a leaf in T. (We add subscripts to the quantifiers for simplicity of the presentation.) Let $t_1, t_2, \ldots, t_d$ is a witnessing substitution of φ^{pn}, where t_j contains $x, x_1, \ldots, x_{j-1}$ as free variables for every $j \in [n]$. The strategy is as follows.

- Fix a model $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ and any $n_0 \in \mathcal{D}$. In the first round, the truthifier chooses the root, adds a child, and puts $t_1^{\mathcal{M}}(n_0)$ on the edge. Suppose that the falsifier puts $n_1 \in \mathcal{D}$ on the edge (so that the edge is labeled with $(t_1(n_0), n_1)$). In the second round, the truthifier works as follows.

- If $\exists_2$ is a child of $\exists_1$ in T, then the truthifier chooses the node corresponding to $\exists_1$, adds a child, and puts $t_2^{\mathcal{M}}(n_0, n_1)$ on the edge.
- Otherwise, $\exists_2$ must be a child of the root in T. Then the truthifier chooses the root, adds a child, and puts $t_2^{\mathcal{M}}(n_0, n_1)$ on the edge.

- Suppose that the falsifier's responses in the first $i-1$ rounds are $n_1, n_2, \ldots, n_{i-1}$. The parent of $\exists_i$ in T is either $\exists_j$ for some $j < i$ or the root. In the i-th round, the truthfier chooses the node introduced in the j-th round if $\exists_j$ is the parent of $\exists_i$ in T, and chooses the root if the parent of $\exists_i$ is the root in T. The truthifer then adds a child, and puts $t_i^{\mathcal{M}}(n_0, n_1, \ldots, n_{i-1})$ on the edge.

It is clear that the strategy can be described by an $\mathcal{L}$-term strategy, so it remains to show that it wins the game within at most d rounds. Suppose that the falsifier's responses in the first d rounds are $n_1, n_2, \ldots, n_d \in \mathcal{D}$. We observe that:

(i) The game tree explored by the truthifier is identical to T. Moreover, the order of explored nodes follows exactly the traversal of T specified by the order of existential quantifiers in φ^{pre} (and φ^{pn}).

(ii) In the i-th round, the truthifier puts $t_i^{\mathcal{M}}(n_0, n_1, \ldots, n_{i-1})$ on the edge. Therefore in the explored game tree, the edge connecting the i-th explored node and its parent is labeled by $(t_i^{\mathcal{M}}(n_0, n_1, \ldots, n_{i-1}), n_i)$.

Let σ be the assignment of variables $\{x \mapsto n_0, x_i \mapsto n_i, y_i \mapsto t_i^{\mathcal{M}}(n_0, n_1, \ldots, n_{i-1})\}$. Since $t_1, \ldots, t_d$ come from witnessing substitution, $\hat{\phi}[\sigma]$ is true in $\mathcal{M}$. In other words, for some $i \in [\ell]$, $\phi^i[\sigma]$ is true in $\mathcal{M}$. Fix this $i \in [\ell]$. Recall that ϕ^i is a copy of $\phi(x, \vec{x}, \vec{y})$ and corresponds to a path in T from the root to a leaf, in the sense that the bounded variables in ϕ^i appears on the path. By Item (i) above, it also corresponds to a path in the *explored game tree* from the root to a leaf.

Suppose that the path corresponds to $\exists_{j_1}, \exists_{j_2}, \ldots, \exists_{j_k}$ from the root to the leaf, where $1 \le j_1 < j_2 < \cdots < j_k \le d$. By Item (ii), the labels on the edges in the path are

$$(t_{j_1}^{\mathcal{M}}(n_0, \ldots, n_{j_1-1}), n_{j_1}), (t_{j_2}^{\mathcal{M}}(n_0, \ldots, n_{j_2-1}), n_{j_2}), \ldots, (t_{j_k}^{\mathcal{M}}(n_0, \ldots, n_{j_k-1}), n_{j_k}). \tag{14}$$

By the definition of σ, the path (14) is a truthifier's winning transcript in the evaluation game. This shows that the truthifier wins the tree exploration game. As the responses of the falsifier can be arbitrary, the aforementioned strategy is a winning strategy of the tree exploration for the truthifier. This completes the proof.

B.2 Oblivious falsifiers: Self-contained proof of Theorem 3.2 via Herbrandization

The *no-counterexample interpretation* (see, e.g., [Koh08, Section 2.3] and [Kra92]) is a standard tool in proof theory to extract computational content from provable sentences of high quantifier complexity. In this section, we use this perspective to provide a different proof of Theorem 3.2. We refer to Section 3.2 for the necessary definitions and notation.

Let $\mathcal{T}$ be a universal theory over $\mathcal{L}$, and let

$$\varphi(x) \triangleq \exists y_1\, \forall x_1\, \exists y_2 \ldots \forall x_{k-1}\, \exists y_k\, \forall x_k\, \phi(x, \vec{x}, \vec{y})$$

be an $\mathcal{L}$-formula, where ϕ is quantifier-free. The *Herbrand normal form* of $\varphi(x)$ is defined as

$$\varphi^H(x) \triangleq \exists y_1\, \exists y_2 \ldots \exists y_k\, \phi(x, x_1/f_1(x, y_1), x_2/f_2(x, y_1, y_2), \ldots, x_k/f_k(x, y_1, y_2, \ldots, y_k), \vec{y}),$$

where $f_1, f_2, \ldots, f_k$ are new function symbols not in $\mathcal{L}$. By a simple model-theoretical argument, $\mathcal{T} \vdash \forall x\, \varphi(x)$ if and only if $\mathcal{T} \vdash \forall x\, \varphi^H(x)$. Under the assumption that $\mathcal{T} \vdash \forall x\, \varphi(x)$, we can apply Theorem 2.8

to extract $\mathcal{L}(f_1, f_2, \ldots, f_k)$-terms that witness the existential quantifiers. In particular, if $\mathcal{T}$ is T_{PV} and $\mathcal{L}$ is $\mathcal{L}_{\mathsf{PV}}$, this witnessing result implies that for every $x \in \mathbb{N}$ and all interpretations of $f_1, f_2, \ldots, f_k$ over $\mathbb{N}$, we can find suitable $y_1, y_2, \ldots, y_k \in \mathbb{N}$ in polynomial-time given oracle access to $f_1^{\mathbb{N}}, f_2^{\mathbb{N}}, \ldots, f_k^{\mathbb{N}}$.

Let $\mathcal{M}$ be a structure over the vocabulary $\mathcal{L}$ such that $\mathcal{M} \vDash \mathcal{T}$ (e.g., $\mathcal{T} = \mathsf{T}_{\mathsf{PV}}$ and $\mathcal{M} = \mathbb{N}$), and let n_0 be an object in the domain of $\mathcal{M}$. It is instructive to consider the following game on the board $(\mathcal{M}, n_0)$. There are two players in the game: a truthifier (or student) that claims $\mathcal{M} \vDash \varphi(n_0)$, and a falsifier (or teacher) that claims $\mathcal{M} \vDash \neg\varphi(n_0)$. In the i-th step, first the truthifier chooses an element n_i for y_i, then the falsifier chooses an element m_i for x_i. The truthifier (resp. falsifier) wins if and only if $\mathcal{M} \vDash \varphi(n_0, m_1, \ldots, m_k, n_1, \ldots, n_k)$ holds (resp. does not hold). It is easy to see that $\mathcal{M} \vDash \varphi(n_0)$ if and only if the truthifier has a winning strategy for the game on board $(\mathcal{M}, n_0)$. The interpretation of the function symbols $f_1, \ldots, f_k$ corresponds naturally to a strategy for the falsifier. The no-counterexample interpretation essentially means that if $\mathcal{T} \vdash \forall x\, \varphi(x)$, for every board $(\mathcal{M}, n_0)$ and every strategy $f_1, \ldots, f_k$ of the falsifier, the truthifier has a winning strategy that can be expressed by terms in $\mathcal{L}(f_1, f_2, \ldots, f_k)$. Next, we transform such a strategy into $\mathcal{L}$-strategies with ancillary information for the truthifier in the evaluation game of $\varphi(x)$.

Theorem (Reminder of Theorem 3.2)**.** *Let $\mathcal{T}$ be a universal theory over the language $\mathcal{L}$ that is closed under if-then-else. Let $\varphi(x)$ be the formula*

$$\begin{aligned}\varphi(x) \triangleq\ & \exists y_1 \le t_1(x)\, \forall x_1 \le s_1(x, y_1)\, \exists y_2 \le t_2(x, y_1, x_1) \ldots \forall x_{k-1} \le s_{k-1}(x, y_1, x_1, \ldots, y_{k-1}) \\ & \exists y_k \le t_k(x, y_1, x_1, \ldots, y_{k-1}, x_{k-1})\, \forall x_k \le s_k(x, y_1, x_1, \ldots, y_k)\, \phi(x, x_1, \ldots, x_k, y_1, \ldots, y_k),\end{aligned}$$

where $\phi(x, \vec{x}, \vec{y})$ is a quantifier-free $\mathcal{L}$-formula. If $\mathcal{T} \vdash \forall x\ \varphi(x)$, then there is a constant $\ell \in \mathbb{N}$ and $\mathcal{L}$-strategies $\tau_1^{\mathtt{t}}, \tau_2^{\mathtt{t}}, \ldots, \tau_\ell^{\mathtt{t}}$ (with ancillary information) such that, for any board $(\mathcal{M}, n_0)$ and evaluation game of $\varphi(x)$ on $(\mathcal{M}, n_0)$, for every strategy $\tau^{\mathtt{f}}$ of the falsifier:

- *either $\hat{\tau}_1^{\mathtt{t}} \triangleq \tau_1^{\mathtt{t}}[\varnothing]$ beats $\tau^{\mathtt{f}}$,*
- *or $\hat{\tau}_2^{\mathtt{t}} \triangleq \tau_2^{\mathtt{t}}[\langle \hat{\tau}_1^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle]$ beats $\tau^{\mathtt{f}}$,*
- *or $\hat{\tau}_3^{\mathtt{t}} \triangleq \tau_3^{\mathtt{t}}[\langle \hat{\tau}_1^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle, \langle \hat{\tau}_2^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle]$ beats $\tau^{\mathtt{f}}$,*
- *$\ldots$,*
- *or $\hat{\tau}_\ell^{\mathtt{t}} \triangleq \tau_\ell^{\mathtt{t}}[\langle \hat{\tau}_1^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle, \langle \hat{\tau}_2^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle, \ldots, \langle \hat{\tau}_{\ell-1}^{\mathtt{t}} : \tau^{\mathtt{f}} \rangle]$ beats $\tau^{\mathtt{f}}$.*

Proof. We introduce Herbrandization functions $f_1, f_2, \ldots, f_k$ such that in $\mathcal{L}^* \triangleq \mathcal{L} \cup \{f_1, \ldots, f_k\}$,

$$\mathcal{T} \vdash \forall x\, \exists \vec{y} \le \vec{t}\, \phi(x, \vec{x}^*, \vec{y}),$$

where $x_j^* = f_j(x, y_1, y_2, \ldots, y_j)$ for all $j \in [k]$. By Herbrand's Theorem (Theorem 2.8), there is a constant $r \in \mathbb{N}$ and $\mathcal{L}^*$-terms $q_j^i(x)$ ($i \in [r], j \in [k]$) such that

$$\mathcal{T} \vdash \forall x \left(\bigvee_{i=1}^{r} \phi_i(x) \right),$$

where $\phi_i(x) \triangleq \phi(x, x_1/f_1(x, q_1^i(x)), \ldots, x_k/f_k(x, q_1^i(x), \ldots, q_k^i(x)), y_1/q_1^i(x), \ldots, y_k/q_k^i(x))$.

We will translate $(q_1^i, q_2^i, \ldots, q_k^i)$ into ℓ_i $\mathcal{L}$-strategies $\tau_{i,1}^{\mathtt{t}}, \tau_{i,2}^{\mathtt{t}}, \ldots, \tau_{i,\ell_i}^{\mathtt{t}}$ for some $\ell_i \in \mathbb{N}$, such that for every board $(\mathcal{M} = (\mathcal{D}, \mathcal{I}), n_0)$ and every interpretation $F_1, F_2, \ldots, F_k$ of $f_1, f_2, \ldots, f_k$ over $\mathcal{D}$ derived from $\hat{\tau}^{\mathtt{f}}$, if $\mathcal{M}(F_1, F_2, \ldots, F_k) \vDash \phi_i(x/n_0)$, then $\tau_{i,1}^{\mathtt{t}}, \tau_{i,2}^{\mathtt{t}}, \ldots, \tau_{i,\ell_i}^{\mathtt{t}}$ will satisfy the conclusion of the theorem against the strategy $\hat{\tau}^{\mathtt{f}}$. If this is possible, then

$$\tau_{1,1}^{\mathtt{t}}, \tau_{1,2}^{\mathtt{t}}, \ldots, \tau_{1,\ell_1}^{\mathtt{t}}, \tau_{2,1}^{\mathtt{t}}, \tau_{2,2}^{\mathtt{t}}, \ldots, \tau_{2,\ell_2}^{\mathtt{t}}, \ldots, \tau_{r,1}^{\mathtt{t}}, \tau_{r,2}^{\mathtt{t}}, \ldots, \tau_{r,\ell_r}^{\mathtt{t}}$$

is a sequence of $\mathcal{L}$-strategies as required. The argument is as follows. Fix any board $(\mathcal{M} = (\mathcal{D}, \mathcal{I}), n_0)$ and any strategy $\tau^{\mathtt{f}}$ of the falsifier. Let $F_1, \ldots, F_k$ be the interpretation of $f_1, \ldots, f_k$ corresponding to this strategy, i.e., for every $j \in [k]$,

$$F_j(n, m_1, m_2, \ldots, m_k) \triangleq \begin{cases} \text{the move of } \tau^{\mathtt{f}} \text{ in the } j\text{-th step} & \text{if } n = n_0 \text{ and } n_1, F_1(n_0, m_1), n_2, F_2(n_0, m_1, m_2), \\ & \ldots, n_{j-1}, F_{j-1}(n_0, m_1, \ldots, m_{j-1}) \text{ is a prefix of a} \\ & \text{valid transcript over } (\mathcal{M}, n_0); \\ 0 & \text{otherwise.} \end{cases}$$

Then there is an index $i \in [r]$ such that $\mathcal{M}(F_1, F_2, \ldots, F_j) \vDash \phi_i(x/n_0)$ holds.

Before presenting the translation, we explain the main difficulty and how to address it. The issue is that $(q_1^i, q_2^i, \ldots, q_k^i)$ are $\mathcal{L}^*$-terms, while the desired strategy in the evaluation game consists of $\mathcal{L}$-terms only. For simplicity, suppose $q_j^i(x)$ invokes a single function from the list $f_1, \ldots, f_k$ of new function symbols, and assume it is $f_1(x, y_1)$. The idea is to replace the computation $F_1(w_1, w_2)$ over inputs w_1, w_2 by forcing the falsifier to compute its value in a previously played game. To achieve this, we use that $\tau^{\mathtt{f}}$ is fixed. In other words, if $w_1 = n_0$, the falsifier must play and reveal $F_1(n_0, w_1)$ if the truthifier plays w_1 in the first round. (On the other hand, if $w_1 \neq n_0$ we have $F_1(w_1, w_2) = 0$ by definition.) Consequently, by playing more games we guarantee that the necessary information appears in the transcript, which allows us to replace calls to functions f_j and express the winning strategy using $\mathcal{L}$-terms. To streamline the presentation, in the description below we omit the trivial case where the first input to a function f_j is different than x, the input to the $\mathcal{L}^*$-terms q_j^i (corresponding to the case $w_1 \neq n_0$ we have just explained).

Let $\tau^{\mathtt{f}}$ be the strategy specified by $f_1, f_2, \ldots, f_k$ (i.e. f_i denotes the falsifier's move in the i-th round). We prove by structural induction on the terms that we can decompose each q_j^i ($j \in [k]$), which consists of $\mathcal{L}$-functions and $f_1, f_2, \ldots, f_k$, into finitely many $\mathcal{L}$-strategies $\tau_1^{i,j}, \tau_2^{i,j}, \ldots, \tau_{d_{i,j}}^{i,j}$ and an $\mathcal{L}$-term $p^{i,j}$ such that $\mathcal{M}(F_1, F_2, \ldots, F_j) \vDash q_j^i(n_0) = p^{i,j}(\Gamma(n_0))$ for every board $(\mathcal{M}, \mathcal{I})$ and strategy $\tau^{\mathtt{f}}$ of the falsifier, where $F_1, F_2, \ldots, F_j$ is the interpretation of $\tau^{\mathtt{f}}$ corresponding to the strategy and $\Gamma(n_0)$ is a sequence of transcripts produced as follows.

- For each $u \in [d_{i,j}]$, let $\Gamma_u(n_0) \triangleq \left\langle \tau_u^{i,j}[\Gamma_1(n_0), \Gamma_2(n_0), \ldots, \Gamma_{u-1}(n_0)] : \tau^{\mathtt{f}} \right\rangle$.
- Let $\Gamma(n_0) \triangleq (\Gamma_1(n_0), \Gamma_2(n_0), \ldots, \Gamma_{d_{i,j}}(n_0))$.

Concretely, we translate each term as follows. Let the term be $g(v_1, v_2, \ldots, v_d)$. By induction hypothesis, for every $r \in [d]$, we can decompose the term v_r into a sequence of $c_r \in \mathbb{N}$ $\mathcal{L}$-strategies $\tau_1^r, \tau_2^r, \ldots, \tau_{c_r}^r$ and an $\mathcal{L}$-term p_r, such that for every board $(\mathcal{M}, m)$, $\mathcal{M} \vDash v_r(n_0) = p_r(\Gamma^r(n_0))$ (where $\Gamma^r(n_0)$ is the transcript of games as described above).

- If $g(\cdot)$ is a function symbol in the original language $\mathcal{L}$, it is easy to see that the $\mathcal{L}$-term

  $$g(p_1(\Gamma(n_0)), p_2(\Gamma(n_0)), \ldots, p_d(\Gamma(n_0)))$$

  and the strategies

  $$(\tau_1^1, \tau_2^1, \ldots, \tau_{c_1}^1, \tau_1^2, \tau_2^2, \ldots, \tau_{c_2}^2, \ldots, \tau_1^d, \tau_2^d, \ldots, \tau_{c_d}^d)$$

  provide what we want, where $\Gamma(n_0) \triangleq (\Gamma^1(n_0), \Gamma^2(n_0), \ldots, \Gamma^{c_d}(n_0))$.
- If $g(\cdot) = f_j$ for some $j \in [k]$, we define a new $\mathcal{L}$-strategy τ^{f_j} as follows: suppose that the ancillary information consists of the transcripts Γ of $\tau_1^1, \tau_2^1, \ldots, \tau_{c_1}^1, \tau_1^2, \tau_2^2, \ldots, \tau_{c_2}^2, \ldots, \tau_1^d, \tau_2^d, \ldots, \tau_{c_d}^d$ vs $\tau^{\mathtt{f}}$;

in the i-th round for $i \le j$, the truthifier's move is

$$\hat{p}_i(n_0, m_1, n_1, \dots, n_{i-1}, \Gamma) \triangleq \begin{cases} p_i(\Gamma) & p_i(\Gamma) \le t_i(n_0, m_1, n_1, \dots, n_{i-1}) \\ 0 & \text{otherwise} \end{cases}$$

while in the remaining $n - i$ rounds the truthifier always chooses 0. Note that $\hat{p}_i$ is expressible since $\mathcal{T}$ is closed under if-then-else. It is clear that the following $\mathcal{L}$-term v_g that takes Γ and the transcript $\langle \tau^{f_j}[\Gamma] : \tau^{\mathtt{f}} \rangle$ as input parameters outputs $f_j(v_1(n_0), \dots, v_d(n_0))$:

- If $\bigvee_i p_i(\Gamma) > t_i(n_0, m_1, n_1, \dots, n_{i-1})$ holds, then v_g outputs 0.
- Otherwise, v_g outputs the j-th move of the falsifier in the transcript $\langle \tau^{f_j}[\Gamma] : \tau^{\mathtt{f}} \rangle$.

Therefore, we can obtain a term v_g that simply reads the transcripts $\Gamma, \langle \tau^{f_j}[\Gamma] : \tau^{\mathtt{f}} \rangle$ and outputs $f_j(v_1(n_0), \dots, v_d(n_0))$, together with the strategies

$$\tau^1_1, \tau^1_2, \dots, \tau^1_{c_1}, \tau^2_1, \tau^2_2, \dots, \tau^2_{c_2}, \dots, \tau^d_1, \tau^d_2, \dots, \tau^d_{c_d}, \tau^{f_j},$$

as promised in the induction hypothesis.

Now we go back to the translation of $(q^i_1, q^i_2, \dots, q^i_k)$ into $\mathcal{L}$-strategies. Assume that each q^i_j for $j \in [k]$ has been decomposed into strategies $\tau^{i,j}_1, \dots, \tau^{i,j}_{d_{i,j}}$ and a term $p^{i,j}(\Gamma)$ as discussed above. Define an $\mathcal{L}$-strategy τ^{q^i} with ancillary information as follows: suppose that the ancillary information is the transcripts $\Gamma(n_0)$ of $\tau^{\mathtt{f}}$ vs

$$\tau^{i,1}_1, \tau^{i,1}_2, \dots \tau^{i,1}_{d_{i,1}}, \tau^{i,2}_1, \tau^{i,2}_2, \dots \tau^{i,2}_{d_{i,2}}, \dots, \tau^{i,k}_1, \tau^{i,k}_2, \dots \tau^{i,k}_{d_{i,k}}$$

in which the latter strategies are given the transcripts of $\tau^{\mathtt{f}}$ vs previous strategies. In the j-th round, the truthifier's move is $p^{i,j}(\Gamma(n_0))$. By construction, it is easy to see that for every board $(\mathcal{M}, n_0)$ and every strategy $\tau^{\mathtt{f}}$ of the falsifier, given correct ancillary information $\Gamma(n_0)$, τ^{q^i} will choose $q^i_j(n_0)$ in the j-th round. Therefore, the strategy will beat $\tau^{\mathtt{f}}$ as long as $\mathcal{M}(F_1, F_2, \dots, F_j) \vDash \phi_i(n_0)$, where $F_1, \dots, F_j$ constitute the interpretation of $f_1, \dots, f_j$ corresponding to $\tau^{\mathtt{f}}$. This completes the proof by previous discussions. □

C Proof of Hardness Amplification in PH

Reminder of Theorem 2.7. *There is a constant $\gamma > 0$ and $\ell = \ell(n) = \mathsf{poly}(n)$ such that the following holds for every $i \ge 1$. Let $s_1, s_2 \colon \mathbb{N} \to \mathbb{N}$ be non-decreasing functions, where $s_2(n) = n^{\omega(1)}$, and suppose there is a function $f_n \colon \{0,1\}^n \to \{0,1\}$ computable by Σ_i-$\mathsf{SIZE}[s_1(n)]$ circuits (resp. Π_i-$\mathsf{SIZE}[s_1(n)]$ circuits) such that each Σ^p_{i-1}-oracle circuit A_n of size at most $s_2(n)$ satisfies*

$$\Pr_{x \in \{0,1\}^n}[f_n(x) = A_n(x)] \;\le\; 1 - \frac{1}{n}.$$

Then there exist a function $h_\ell \colon \{0,1\}^\ell \to \{0,1\}$ computable by Σ_i-$\mathsf{SIZE}[\mathsf{poly}(\ell) \cdot s_1(\ell)]$ circuits (resp. Π_i-$\mathsf{SIZE}[\mathsf{poly}(\ell) \cdot s_1(\ell^\gamma)]$ circuits) such that each Σ^p_{i-1}-oracle circuit B_ℓ of size at most $s_2(\ell^\gamma)^\gamma$ satisfies

$$\Pr_{y \in \{0,1\}^\ell}[h_\ell(y) = B_\ell(y)] \;\le\; \frac{1}{2} + \frac{1}{s_2(\ell^\gamma)^\gamma}.$$

Note that since Σ_{i-1}^p-oracle circuits are closed under complementation, we only need to prove the case where f_n is computable by Σ_i circuits. More formally, given a function $f : \{0,1\}^n \to \{0,1\}$ computable by Π_i-$\mathsf{SIZE}[s_1(n)]$ circuits that is hard on average against Σ_{i-1}^p-oracle circuits of size $s_2(n)$, we can consider $g(x) \triangleq \neg f(x)$ that is computable by Σ_i-$\mathsf{SIZE}[s_1(n)]$ circuits and still hard on average against the same class. By the hardness amplification theorem for Σ_i circuits, we can obtain a function h_ℓ computable by Σ_i-$\mathsf{SIZE}[\mathsf{poly}(\ell) \cdot s_1(\ell)]$ circuits that is strongly hard on average against Σ_{i-1}^p-oracle circuits of size $s_2(\ell^\gamma)^\gamma$. The negation of h_ℓ is then the required hard function computable in Π_i-$\mathsf{SIZE}[\mathsf{poly}(\ell) \cdot s_1(\ell)]$.

We first fix the notation.

- A *probabilistic function* is a Boolean function with two inputs $h(x;r)$ where the second input is treated as random bits. If the random bits are omitted, a probabilistic function is treated as a function mapping the input to a random variable distributed according to the output of the function over the random bits.
- Let g be a function probabilistic function) with input length n, the *k-th direct product* is defined as the function (resp. probabilistic function) with input length $k \cdot n$ and output length k as follows:
$$g^{\otimes k}(x_1, \ldots, x_k) \triangleq g(x_1) \| \ldots \| g(x_k) .$$
- The *bias* of a random variable X is defined as $\mathsf{Bias}(X) \triangleq \big|\Pr[X = 0] - \Pr[X = 1]\big|$. The *bias* of a probabilistic function $h(x;r)$ is defined as the bias of the random variable $h(x;r)$ for a uniformly random x and r. The probabilistic function h is said to be *balanced* if $\mathsf{Bias}(h) = 0$.
- A probabilistic function $h : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}$ is *δ-random* if h is balanced and there is a subset $H \subseteq \{0,1\}^n$ of size $2\delta \cdot 2^n$ such that h is a "coin flip" over H and deterministic outside H (i.e., $\Pr[h(x) = 1] = 1/2$ for every $x \in H$, and $h(x)$ is deterministic for every $x \notin H$).
- The *expected bias* of a probabilistic function h is defined as $\mathsf{ExpBias}(h) \triangleq \mathbb{E}_x\left[\mathsf{Bias}(h(x))\right]$.
- The *noise stability* of a Boolean function $C : \{0,1\}^k \to \{0,1\}$ with respect to the noise rate δ is defined as
$$\mathsf{NoiseStab}_\delta(g) \triangleq 2 \cdot \Pr_{x,\eta}[C(x) = C(x \oplus \eta)] - 1,$$
where $x \sim \{0,1\}^k$ and each bit of η is 1 independently with probability δ. By Lemma 3.7 of [HVV06], $\mathsf{ExpBias}[C \circ g^{\otimes k}] \le \sqrt{\mathsf{NoiseStab}_\delta[C]}$ for every δ-random probabilistic function g.
- Two random variables X_1 and X_2 are said to be ε-indistinguishable for size s, denoted by $X_1 \approx_\varepsilon^s X_2$, if for every Σ_{i-1}^p-oracle circuit C of size s, $\big|\Pr[C(X_1) = 1] - \Pr[C(X_2) = 1]\big| \le \varepsilon$. Note that our definition of the indistinguishability differs from the original definition in [HVV06] since we are proving hardness amplification against Σ_{i-1}^p-oracle circuits.
- For simplicity, we say a function $f : \{0,1\}^n \to \{0,1\}$ is ε-hard for size s, if for every Σ_{i-1}^p-oracle circuit C of size s, $C(x) = f(x)$ for at most an ε fraction of $x \in \{0,1\}^n$.

We assume that f_n is *balanced*, that is, $\Pr_x[f_n(x) = 1] = 1/2$ for every $n \ge 1$. This is without loss of generality, since we can first increase the input length by one then use non-uniformity to make the resulting function balanced, without a relevant change of parameters.

The hardness amplification of [HVV06] proceeds as follows.

The Construction. Fix any $n \ge 1$. Let $f : \{0,1\}^n \to \{0,1\}$ be the hard function and $C : \{0,1\}^k \to \{0,1\}$ be an explicit circuit to be determined later. Let $G : \{0,1\}^\ell \to (\{0,1\}^n)^k$ be an explicit function in the sense that given $\sigma \in \{0,1\}^\ell$ and $i \in [k]$, we can compute the i-th block $X_i \in \{0,1\}^n$ of the output of $G(\sigma)$ in $\mathsf{poly}(\ell, \log k)$ time.[28] The amplified function is defined as $\mathsf{Amp}_f : \{0,1\}^\ell \to \{0,1\}$:

$$\mathsf{Amp}_f(\sigma) \triangleq C(f(X_1), f(X_2), \dots, f(X_k)),$$

where $(X_1, X_2, \dots, X_k) \triangleq G(\sigma)$. We need to carefully choose C and G such that $\mathsf{Amp}_f(\sigma)$ is computable in Σ_i-$\mathsf{SIZE}[\mathsf{poly}(n) \cdot s_1(n)]$ and can amplify the hardness of f.

The Choice of G. To ensure the hardness of Amp_f, the function $G_k : \{0,1\}^\ell \to (\{0,1\}^n)^k$ should satisfy the following two technical requirements.

- G_k is *indistinguishability-preserving for size* $t = k^2$: Let $f_1, \dots, f_k, g_1, \dots, g_k$ be probabilistic functions such that for every $i \in [k]$, $x \| f_i(x) \approx_\varepsilon^s x \| g_i(x)$ for $x \sim \{0,1\}^n$, then

  $$\sigma \| f_1(X_1) \| \dots \| f_k(X_k) \approx_{k \cdot \varepsilon}^{s-t} \sigma \| g_1(X_1) \| \dots \| g_k(X_k),$$

  where $\sigma \sim \{0,1\}^\ell$ and $(X_1, \dots, X_k) \triangleq G_k(\sigma)$.
- G_k is 2^{-n}-pseudorandom against (read-once oblivious) branching programs of size 2^n and block-size n:[29] for every branching program B of size 2^n and block-size n, we have

  $$\left| \Pr_{x \sim \{0,1\}^\ell}[B(G_k(x)) = 1] - \Pr_{y \sim \{0,1\}^{nk}}[B(y) = 1] \right| \le 2^{-n}.$$

Lemma C.1 (Generalized version of [HVV06, Theorem 5.12])**.** *For every $k \le 2^n$, there is an explicit computable generator $G_k : \{0,1\}^\ell \to (\{0,1\}^n)^k$ that satisfies the requirements below:*

(i) There is an algorithm that computes the i-th block of $G_k(\sigma)$ in $\mathsf{poly}(\ell, \log k)$ time given σ, i.

(ii) G_k is indistinguishability-preserving for size $t = k^2$.

(iii) G_k is 2^{-n}-pseudorandom against branching programs of size 2^n and block-size n.

Proof. The only difference between this lemma and [HVV06, Lemma 5.12] is that in our definition, the indistinguishability-preserving property holds against Σ_{i-1}^p-oracle circuits instead of standard circuits, which will not cause any issue since their argument only requires mild closure properties of the adversary. For completeness, we sketch their proof here.

The generator G_k is defined as the XOR of two generators: a Nisan-Wigderson based generator $\mathsf{NW}_k : \{0,1\}^{\ell_{\mathsf{NW}}} \to (\{0,1\}^n)^k$ that is efficiently computable and indistinguishability-preserving; and Nisan's unconditional PRG $\mathsf{N}_k : \{0,1\}^{\ell_{\mathsf{N}}} \to (\{0,1\}^n)^k$ against (probabilistic) branching programs (see, e.g., [HVV06, Theorem 5.6] and [Nis92]). That is, $G_k(x,y) \triangleq \mathsf{NW}_k(x) \oplus \mathsf{N}_k(y)$. Both N_k and NW_k have seed length at most $O(n^2)$, hence G_k has seed length $\ell = O(n^2)$. Next, we discuss the properties of the generator.

- Both NW_k and N_k are efficiently computable in the sense that given σ and i, we can compute the i-th block of the output in $\mathsf{poly}(\ell, \log k)$ time. Therefore Item (i) holds.

[28] We note that C is used to replace the XOR function in the standard hardness amplification based on Yao's XOR Lemma (see, e.g., Theorem 19.2 of [AB09]), while G is used as a pseudorandom generator that (in some sense) "fools" $C \circ f^{\otimes k}$.

[29] See [HVV06, Definition 5.4] for the precise definition of this branching program model.

- To prove Item *(ii)*, we need to show that any indistinguishability-preserving generator XORed with a fixed string is still indistinguishability-preserving. Towards a contradiction, assume that G_k is not indistinguishability-preserving. This means that there are $f_1, \ldots, f_k, g_1, \ldots, g_k$ such that for every $i \in [k]$, $x \| f_i(x) \approx^s_\varepsilon x \| g_i(x)$ for $x \sim \{0,1\}^n$, while for $(\sigma_1, \sigma_2) \sim \{0,1\}^{\ell_{\mathsf{NW}}} \times \{0,1\}^{\ell_{\mathsf{N}}}$ and $(X_1, \ldots, X_k) \triangleq \mathsf{NW}_k(\sigma_1) \oplus \mathsf{N}_k(\sigma_2)$,
$$\sigma_1 \| \sigma_2 \| f_1(X_1) \| \ldots \| f_k(X_k) \not\approx^{s-t}_\varepsilon \sigma_1 \| \sigma_2 \| g_1(X_1) \| \ldots \| g_k(X_k).$$
By an averaging argument, there is a $\sigma_2^* \in \{0,1\}^{\ell_{\mathsf{N}}}$ such that
$$\sigma_1 \| f_1(X_1 \oplus y_1) \| \ldots \| f_k(X_k \oplus y_k) \not\approx^{s-t}_\varepsilon \sigma_1 \| g_1(X_1 \oplus y_1) \| \ldots \| g_k(X_k \oplus y_k), \tag{15}$$
where $(y_1, \ldots, y_k) \triangleq \mathsf{N}_k(\sigma_2^*)$. Let $f_i'(x) \triangleq f_i(x \oplus y_i)$ and $g_i'(x) \triangleq g_i(x \oplus y_i)$ for $i \in [k]$. Clearly for every $i \in [k]$, $x \| f_i'(x) \approx^s_\varepsilon x \| g_i'(x)$,[30] which is impossible since NW_k is indistinguishability-preserving but Equation (15) holds.
- Similarly, we can show that since N_k is 2^{-n}-pseudorandom against branching programs of size 2^n, after XORed with another generator, G_k is still 2^{-n}-pseudorandom against branching programs of size 2^n. This implies Item *(iii)*.

It remains to verify that the Nisan-Wigderson based generator NW_k is indistinguishability-preserving for size k^2 against Σ^p_{i-1}-oracle circuits. Let $\ell = O(n^2)$ and $S_1, S_2, \ldots, S_k \subseteq [\ell]$ be an $(\ell, n, \log k)$-design (see Section 2.4 and [Nis92]). Then $\mathsf{NW}_k : \{0,1\}^\ell \to (\{0,1\}^n)^k$ is defined as
$$\mathsf{NW}_k(\sigma) \triangleq (\sigma|_{S_1}, \sigma|_{S_2}, \ldots, \sigma|_{S_k}).$$
Let $f_1, \ldots, f_k, g_1, \ldots, g_k$ be probabilistic functions such that for every $i \in [k]$, $x \| f_i(x) \approx^s_\varepsilon x \| g_i(x)$ for $x \sim \{0,1\}^n$. Suppose, for the sake of contradiction, that
$$\sigma \| f_1(\sigma|_{S_1}) \| \ldots \| f_k(\sigma|_{S_k}) \not\approx^{s-k^2}_{k\cdot\varepsilon} \sigma \| g_1(\sigma|_{S_1}) \| \ldots \| g_k(\sigma|_{S_k}). \tag{16}$$
For every $i \in [0, k]$, we define the hybrid distribution
$$H_i = \sigma \| g_1(\sigma|_{S_1}) \| \ldots \| g_i(\sigma|_{S_i}) \| f_{i+1}(\sigma|_{S_{i+1}}) \| \ldots \| f_k(\sigma|_{S_k}).$$
Then the distinguisher D for Equation (16), which is a Σ^p_{i-1}-oracle circuit of size $s - k^2$, can distinguish between H_i and H_{i+1} with advantage at least ε for some $i \in [0, k-1]$. Note that
$$\begin{aligned} H_i &= \sigma \| g_1(\sigma|_{S_1}) \| \ldots \| g_i(\sigma|_{S_i}) \| f_{i+1}(\sigma|_{S_{i+1}}) \| f_{i+2}(\sigma|_{S_{i+2}}) \| \ldots \| f_k(\sigma|_{S_k}) \qquad \text{and} \\ H_{i+1} &= \sigma \| g_1(\sigma|_{S_1}) \| \ldots \| g_i(\sigma|_{S_i}) \| g_{i+1}(\sigma|_{S_{i+1}}) \| f_{i+2}(\sigma|_{S_{i+2}}) \| \ldots \| f_k(\sigma|_{S_k}) \end{aligned}$$
differ only on the $(i+2)$-th part: H_i has $f_{i+1}(\sigma|_{S_{i+1}})$ while H_{i+1} has $g_{i+1}(\sigma|_{S_{i+1}})$.

By an averaging argument, we can fix all the bits of σ outside of S_{i+1} so that $\hat{H}_i$ and $\hat{H}_{i+1}$ are still distinguishable with advantage ε, where $\hat{H}_i$ and $\hat{H}_{i+1}$ refer to the distribution H_i and H_{i+1} after we fix the bits of σ outside of S_{i+1}. Since for every $j \neq i+1$, $|S_j \cap S_{i+1}| \leq \log k$, we can construct a Σ^p_{i-1}-oracle circuit of size at most $(s - k^2) + 2^{\log k} \cdot k = s$ that hardwires all possibilities for the common parts of H_i and H_{i+1} such that:

- Given the unfixed bits of σ and $f_{i+1}(\sigma)$, it generates $\hat{H}_i$ and outputs $D(\hat{H}_i)$.
- Given the unfixed bits of σ and $g_{i+1}(\sigma)$, it generates $\hat{H}_{i+1}$ and outputs $D(\hat{H}_{i+1})$.

Since D can distinguish between $\hat{H}_i$ and $\hat{H}_{i=1}$ with advantage ε, the circuit above can distinguish bewteen $\sigma \| f_{i+1}(\sigma \| S_{i+1})$ and $\sigma \| g_{i+1}(\sigma \| S_{i+1})$ with advantage ε. This leads to a contradiction. □

[30] There is no loss in the circuit size of the adversary if we define the circuit model so that NOT gates are free.

The Choice of C. The outer function C, which serves as the counterpart of the XOR function in Yao's XOR Lemma (see, e.g., [AB09, Theorem 12.9]), is chosen according to the following lemma.

Lemma C.2 (Generalized version of [HVV06, Lemma 5.15])**.** *For every $i \geq 1$, $\delta(n) = 1/\mathsf{poly}(n)$, and $k = k(n)$ such that $n^{\omega(1)} \leq k \leq 2^n$, there is a function $C_k : \{0,1\}^k \to \{0,1\}$ such that:*

(i) $\mathsf{NoiseStab}_\delta[C_k] \leq 1/k^{\Omega(1)}$;

(ii) *For every $f : \{0,1\}^n \to \{0,1\}$ computable by Σ_i-$\mathsf{SIZE}[s(n)]$ circuits, $(C_k \circ f^{\otimes k}) \circ G_k : \{0,1\}^\ell \to \{0,1\}$ is computable by Σ_i-$\mathsf{SIZE}[\mathsf{poly}(n) \cdot s(n)]$ circuits.*

(iii) *C_k is computable by a branching program of size $\mathsf{poly}(n) \cdot k$ and by a deterministic circuit of size $\mathsf{poly}(n) \cdot k$.*

Proof. Let $\delta = \delta(n) \geq 1/\mathsf{poly}(n)$ and $k = k(n)$ such that $n^{\omega(1)} \leq k \leq 2^n$. We will define C_k as the composition of two functions defined as follows:

- The *recursive-majority function* $\mathsf{RMaj}_r : \{0,1\}^{3^r} \to \{0,1\}$ is recursively defined by

$$\begin{aligned} \mathsf{RMaj}_1(x_1, x_2, x_3) &\triangleq \mathsf{Maj}(x_1, x_2, x_3) \\ \mathsf{RMaj}_r(x_1, \ldots, x_{3^r}) &\triangleq \mathsf{RMaj}_{r-1}(\mathsf{Maj}(x_1, x_2, x_3), \ldots, \mathsf{Maj}(x_{3^r-2}, x_{3^r-1}, x_{3^r})) \end{aligned}$$

  where $\mathsf{Maj}(x_1, x_2, x_3)$ is the majority value among $x_1, x_2, x_3 \in \{0,1\}$.

- The *tribes function* of k bits is defined by

$$\mathsf{Tribes}_k(x_1, \ldots, x_k) \triangleq (x_1 \wedge \cdots \wedge x_b) \vee (x_{b+1} \wedge \cdots \wedge x_{2b}) \vee \cdots \vee (x_{k-b+1} \wedge \cdots \wedge x_k),$$

  where $b = O(\log k)$ is the largest integer such that $(1 - 2^{-b})^{k/b} \geq 1/2$.

Let $r \triangleq c \cdot \log(1/\delta)$ for a constant c to be determined later. Assuming without loss of generality that r and $k/3^r$ are integers, we define $C_k : \{0,1\}^k \to \{0,1\}$ by

$$C_k \triangleq \mathsf{Tribes}_{k/3^r} \circ \mathsf{RMaj}_r^{\otimes k/3^r}.$$

As [HVV06, Section 5.5] in the proof of Lemma 5.15, we know that for some sufficiently large constant c, the noise stability of C_k is at most $1/k^{\Omega(1)}$. Also they showed that C_k can be computed by a branching program of size $\mathsf{poly}(n) \cdot k$ and a deterministic circuit of size $\mathsf{poly}(n) \cdot k$.

It remains to determine the complexity of $(C_k \circ f^{\otimes k}) \circ G_k$ for $f : \{0,1\}^n \to \{0,1\}$ computable by Σ_i-$\mathsf{SIZE}[s(n)]$ circuits. Consider the following Σ_i-circuit. We first guess (using non-determinism) a clause K of the upper $\mathsf{Tribes}_{k/3^r}$ function that is satisfied. For every RMaj_r function feeding into this clause (there are $b = O(\log k) = \mathsf{poly}(n)$ such RMaj_r functions), we guess the input bits of the upper C_k sub-circuit (or equivalently, the output bits of the lower f functions) that are 1 and

(i) we verify that these input bits that are 1 make the clause K accept, which can be done by a deterministic circuit of size $\mathsf{poly}(3^r) = \mathsf{poly}(n)$ since RMaj is a monotone function;

(ii) for every guessed input bit of C_k (or equivalently, the output bit of one of f in the middle $f^{\otimes k}$ layer) that is supposed to be 1, we use the Σ_i-$\mathsf{SIZE}[s(n)]$ circuit for f to verify that it is indeed 1. The input to this function f is one of the n-bit blocks of the output of G_k, which can be computed by a deterministic algorithm in $\mathsf{poly}(\ell, \log k) = \mathsf{poly}(n)$ time (see Lemma C.1).

The overall Σ_i-circuit complexity of $(C_k \circ f^{\otimes k}) \circ G_k$ is at most $\mathsf{poly}(n) \cdot s(n)$. □

Note that the second item means that the function $(C_k \circ f^{\otimes k}) \circ G$ is efficiently computable *even if k is as large as* 2^n. The argument relies on the explicitness of C_k and G as well as on the power of Σ_i-circuits. This is crucial for hardness amplification up to $1/2 - 1/s_2(\ell^\gamma)^\gamma$ (instead of only $1/2 - 1/\mathsf{poly}(\ell)$).

Proof of the Hardness Amplification. Following [HVV06, Section 5], we now argue that if f is δ-hard for size $s(n) \geq n^{\omega(1)}$, where $\delta \geq 1/\mathsf{poly}(n)$, then we can construct $\mathsf{Amp}_f : \{0,1\}^\ell \to \{0,1\}$ with $\ell = \mathsf{poly}(n)$ that is $(1/2 - 1/s(\sqrt{\ell})^{\Omega(1)})$-hard for size $s(\sqrt{\ell})^{\Omega(1)}$. To prove this, we need the following two technical lemmas.

Lemma C.3 ([HVV06, Lemma 5.7 and Lemma 5.12])**.** *Let g be an n-input single output δ-random function, and C_k and G_k be defined as above. Then*

$$\mathsf{ExpBias}[(C_k \circ g^{\otimes k}) \circ G_k] \leq \sqrt{\mathsf{NoiseStab}_\delta(C_k) + 2^{-n+1}}.$$

Lemma C.4 (Generalized version of [HVV06, Lemma 5.2])**.** *Assume that $f : \{0,1\}^n \to \{0,1\}$ is δ-hard for size $s = n^{\omega(1)}$. There is a δ'-random function g with $\delta' \in [\delta/2, \delta]$ such that $\mathsf{Amp}_f : \{0,1\}^\ell \to \{0,1\}$ has hardness*

$$\frac{1}{2} - \frac{\mathsf{ExpBias}[(C \circ g^{\otimes k}) \circ G]}{2} - \frac{k}{s^{1/3}}$$

for size $\Omega(s^{1/3}/\log(s/\delta)) - k^2 - \mathsf{poly}(n) \cdot k$.

Before proving this lemma, we need to verify that Impagliazzo's hardcore lemma (see, e.g., [AB09, Section 19.1.2]) holds against adversaries with access to Σ_{i-1}^p oracles.

Lemma C.5 (Generalized version of Impagliazzo's Hardcore Lemma)**.** *Assume that $2n < s < 0.001 \cdot (\varepsilon\delta)^2 \cdot 2^n/n$. Let $f : \{0,1\}^n \to \{0,1\}$ be a balanced function that is δ-hard for Σ_{i-1}^p-oracle circuits of size s. There exists a δ'-random function $g : \{0,1\}^n \to \{0,1\}$ such that $X \| f(X) \approx_\varepsilon^{s'} X \| g(X)$ for $X \sim \{0,1\}^n$, where $s' = \Omega(s\varepsilon^2/\log(1/(\delta\varepsilon)))$ and $\delta' \in [\delta/2, \delta]$.*

Proof Sketch. We follow the proof presented in [AB09, Section 19.1.2] based on the *min-max theorem* for zero-sum games (also see, e.g., [Imp95]). We say that a distribution $\mathcal{H}$ over $\{0,1\}^n$ has density δ if for every $x \in \{0,1\}^n$, $\mathcal{H}(x) \leq 1/(\delta 2^n)$. Let $\delta_1 = 0.99\delta$. We first show that there is a distribution $\mathcal{H}$ of density δ_1 such that for every Σ_{i-1}^p-oracle circuit C of size s', $\Pr[f(x) = C(x)] < 1/2 + \varepsilon/2$ for $x \sim \mathcal{H}$.

Towards a contradiction, we assume that such distribution does not exist. By a game-theoretic argument using the min-max theorem, we can construct a distribution $\mathcal{C}$ over Σ_{i-1}^p-oracle circuits of size s' such that for every distribution $\mathcal{H}$ of density δ_1, a random $C \sim \mathcal{C}$ can approximate f over $\mathcal{H}$ with error $\leq 1/2 - \varepsilon/2$.

An input $x \in \{0,1\}^n$ is said to be *bad* if $\Pr[C(x) \neq f(x)] > 1/2 - \varepsilon/2$ for $C \sim \mathcal{C}$. It is said to be *good* otherwise. There are at most $\delta_1 \cdot 2^n$ bad inputs, since otherwise we can let $\mathcal{H}$ be the uniform distribution over a set of $\delta_1 \cdot 2^n$ bad inputs and violate the aforementioned property of $\mathcal{C}$. Let $t = O(\varepsilon^{-2}\log(1/(\delta\varepsilon)))$ and C be the following probabilistic circuit (with Σ_{i-1}^p oracles): given input x, obtain t independent samples $C_1, \dots, C_t \sim \mathcal{C}$, and output the majority of $C_1(x), \dots, C_t(x)$. This probabilistic circuit has size at most $t \cdot s' \leq s$. By the Chernoff bound, it computes $f(x)$ for any good x with error at most $\exp(-\Omega(\varepsilon^2 t)) \leq 0.001 \cdot \delta$. This means that for a uniformly random $x \sim \{0,1\}^n$, the probabilistic Σ_{i-1}^p-oracle circuit (and also deterministic Σ_{i-1}^p-orcle circuit by an averaging argument) can approximate $f(x)$ with error at most $\delta_1 + \delta/2 \leq \delta$ for an $x \sim \{0,1\}^n$, which is impossible.

We then prove via a probabilistic argument that there is a subset H of size $\delta' \in [\delta/2, \delta]$ such that no Σ_{i-1}^p-oracle circuit of size s can approximate f on H with advantage ε. Let H be a *random* subset defined as follows: for every $x \in \{0,1\}^n$, we let $x \in H$ independently with probability $\mathcal{H}(x)$. By a "concentration bound then union bound" argument, we get with non-zero probability that H has size $\delta' \in [\delta/2, \delta]$ *and* for every C of size s, $\Pr[f(x) = C(x)] \le 1/2 + 1/\varepsilon$. This means that the δ'-random function g defined over H satisfies the conditions of the lemma. □

Proof of Lemma C.4. Assume that $f : \{0,1\}^n \to \{0,1\}$ is δ-hard for size $s = n^{\omega(1)}$. By Impagliazzo's hardcore lemma, there is a δ'-random function $g : \{0,1\}^n \to \{0,1\}$ such that $X \| f(X) \approx_{\varepsilon}^{s'} X \| g(X)$ for $X \sim \{0,1\}^n$, where $s' = \Omega(s\varepsilon^2 / \log(1/(\delta\varepsilon)))$ and $\delta' \in [\delta/2, \delta]$. Since G is indistinguishability-preserving for size k^2, we get that

$$\sigma \| f(X_1) \| \dots \| f(X_k) \approx_{k\varepsilon}^{s'-k^2} \sigma \| g(X_1) \| \dots \| g(X_k),$$

where $\sigma \sim \{0,1\}^\ell$ and $(X_1, \dots, X_k) = G(\sigma)$. Since C_k has complexity bounded by $\mathsf{poly}(n) \cdot k$ this further means that

$$\sigma \| C_k(f(X_1), \dots, f(X_k)) \approx_{k\varepsilon}^{s''} \sigma \| C_k(g(X_1), \dots, g(X_k)),$$

where $s'' = s' - k^2 - \mathsf{poly}(n) \cdot k$. Note that

$$C_k(f(X_1), \dots, f(X_k)) = (C_k \circ f^{\otimes k}) \circ G_k(\sigma) \quad \text{and} \quad C_k(g(X_1), \dots, g(X_k)) = (C_k \circ g^{\otimes k}) \circ G_k(\sigma)\,.$$

Also we can see that the for every probabilistic function h, the statistical distance between $X \| h(X)$ and $X \| b$ for $X \sim \{0,1\}^n$ and $b \sim \{0,1\}$ is exactly $\mathsf{ExpBias}[h]/2$ (see, e.g., [HVV06, Lemma 3.4]). Therefore we know that

$$\Delta(\sigma \| (C_k \circ g^{\otimes k}) \circ G_k(\sigma), \sigma \| b) \le \frac{\mathsf{ExpBias}[(C_k \circ g^{\otimes k}) \circ G_k]}{2},$$

where $\sigma \sim \{0,1\}^\ell$ and $b \sim \{0,1\}$. This further means that $\sigma \| (C_k \circ f^{\otimes k}) \circ G_k(\sigma)$ and $\sigma \| b$ are $k\varepsilon + (1/2) \cdot \mathsf{ExpBias}[(C_k \circ g^{\otimes k}) \circ G_k]$ indistinguishable for size s''. By setting $\varepsilon = s^{-1/3}$, we obtain the lemma. □

Let $k = k(n) = s(n)^{1/7}$, C_k be the function in Lemma C.2, and G_k be the generator in Lemma C.1 with $\ell = O(n^2)$. Recall that $\mathsf{Amp}_f : \{0,1\}^\ell \to \{0,1\}$ is defined as $\mathsf{Amp}_f \triangleq (C_k \circ f^{\otimes k}) \circ G_k$, and note that the upper bound on the complexity of Amp_f is guaranteed by Lemma C.2. By Lemma C.4, we know that Amp_f has hardness

$$\frac{1}{2} - \frac{\mathsf{ExpBias}[(C_k \circ g^{\otimes k}) \circ G_k]}{2} - \frac{k}{s^{1/3}} \tag{17}$$

for size $\Omega(s^{1/3}/\log(s/\delta)) - k^2 - \mathsf{poly}(n) \cdot k = s(\sqrt{\ell})^{\Omega(1)}$, where g is some δ'-random function with $\delta' \in [\delta/2, \delta]$. By Lemma C.3, we can bound Equation (17) using the noise stability bound for C_k given in Lemma C.2:

$$\begin{aligned} (17) \ \ge\ & \frac{1}{2} - \frac{\sqrt{\mathsf{NoiseStab}_{\delta'}(C_k) + 2^{-n+1}}}{2} - \frac{k}{s^{1/3}} \\ \ge\ & \frac{1}{2} - \frac{\sqrt{k^{-\Omega(1)} + 2^{-n+1}}}{2} - \frac{k}{s(n)^{1/3}} \\ \ge\ & \frac{1}{2} - \frac{1}{s(\sqrt{\ell})^{\Omega(1)}}. \end{aligned}$$

This completes the argument.

D A Universal Theory for $\mathsf{T}^i_{\mathsf{PV}}$

In this section, we describe the proofs omitted from Section 2.7. First, we will need some auxiliary results.

Lemma D.1. *Let $i \geq 0$. For every Σ^b_i-formula (resp. Π^b_i-formula) $\alpha(\vec{z})$ in the language $\mathcal{L}_{\mathsf{PV}}$, there exists a formula $\alpha^{\mathsf{norm}}(\vec{z}) = Q_1 x_1 \leq t_1(\vec{z})\ Q_2 x_2 \leq t_2(\vec{z}) \ldots Q_i x_i \leq t_i(\vec{z})\ \phi(\vec{z}, \vec{x})$, where $Q_1 = \exists$ (resp. $Q_1 = \forall$), $Q_j \in \{\forall, \exists\}$, and $Q_j \neq Q_{j+1}$ for every $j \leq i - 1$, such that $\mathsf{T}^1_{\mathsf{PV}} \vdash \forall \vec{z}\, (\alpha(\vec{z}) \leftrightarrow \alpha^{\mathsf{norm}}(\vec{z}))$.*

Proof. Note that for every Σ^b_i-formula (resp. Π^b_i-formula) $\alpha(\vec{z})$, we can firstly find its prenex normal form $\alpha^{\mathsf{pnf}}(\vec{z})$ with $i - 1$ quantifier alternations starting with an existential (resp. universal) quantifier that is logically equivalent to $\alpha(\vec{z})$. Note that $\mathsf{T}^1_{\mathsf{PV}}$ defines pairing and unpairing functions. Concretely, there are functions $\langle \cdot, \cdot \rangle, \pi_1(\cdot), \pi_2(\cdot)$ such that $\mathsf{T}^1_{\mathsf{PV}} \vdash \forall x\, \forall y\, (\pi_1(\langle x, y \rangle) = x \wedge \pi_2(\langle x, y \rangle) = y \wedge |\langle x, y \rangle| \leq 10 \cdot (|x| + |y|))$. It is then easy to see that

$$\mathsf{T}^1_{\mathsf{PV}} \vdash \Big(\forall x \leq s\ \forall y \leq t\ \varphi(x, y)\Big) \leftrightarrow \Big(\forall p \leq (s \cdot t)^{10} (\pi_1(p) \leq s \wedge \pi_2(p) \leq t \rightarrow \varphi(\pi_1(p), \pi_2(p)))\Big)$$
$$\mathsf{T}^1_{\mathsf{PV}} \vdash \Big(\exists x \leq s\ \exists y \leq t\ \varphi(x, y)\Big) \leftrightarrow \Big(\exists p \leq (s \cdot t)^{10} (\pi_1(p) \leq s \wedge \pi_2(p) \leq t \wedge \varphi(\pi_1(p), \pi_2(p)))\Big).$$

Therefore we can further collapse adjacent quantifiers of the same kind to obtain $\alpha^{\mathsf{norm}}(\vec{z})$ as described above such that $\mathsf{T}^1_{\mathsf{PV}} \vdash \forall \vec{z}\, (\alpha(z) \leftrightarrow \alpha^{\mathsf{pnf}}(\vec{z}) \leftrightarrow \alpha^{\mathsf{norm}}(\vec{z}))$. □

Lemma D.2. *For every $i \geq 1$ and $(\Pi^b_{i-1} \cup \Sigma^b_{i-1})$-formula $\alpha(\vec{x})$, we have* (1) $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{x}\, (f_\alpha(\vec{x}) = 1 \leftrightarrow f_{\neg\alpha}(\vec{x}) = 0)$ *and* (2) $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{x}\, (f_\alpha(\vec{x}) = 0 \vee f_\alpha(\vec{x}) = 1)$.

Proof. By the definition of each $f^{\mathbb{N}}_\alpha$, we can see that the universal sentences $\forall \vec{x}\, (f_\alpha(\vec{x}) = 1 \leftrightarrow f_{\neg\alpha}(\vec{x}) = 0)$ and $\forall \vec{x}\, (f_\alpha(\vec{x}) = 0 \vee f_\alpha(\vec{x}) = 1)$ are both true in the standard model, so they are provable in $\mathsf{U}^i_{\mathsf{PV}}$. □

Reminder of Lemma 2.12. *Let $i \geq 2$, $\beta(\vec{x}, y)$ be any Σ^b_{i-1}-formula in $\mathcal{L}_{\mathsf{PV}}$, and t be any term in $\mathcal{L}_{\mathsf{PV}}$. Then $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{x}\, ((\exists y \leq t(\vec{x})\ f_\beta(\vec{x}, y) = 1) \leftrightarrow f_\beta(\vec{x}, g_{\beta,t}(\vec{x})) = 1)$.*

Proof. We will show separately that:

$$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{x}\, ((\exists y \leq t(\vec{x})\ f_\beta(\vec{x}, y) = 1) \rightarrow f_\beta(\vec{x}, g_{\beta,t}(\vec{x})) = 1), \tag{Case 1}$$
$$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{x}\, (f_\beta(\vec{x}, g_{\beta,t}(\vec{x})) = 1 \rightarrow (\exists y \leq t(\vec{x})\ f_\beta(\vec{x}, y) = 1)). \tag{Case 2}$$

- Case 1: It's easy to see that the sentence we want to prove (in $\mathsf{U}^i_{\mathsf{PV}}$) is logically equivalent to the following universal sentence: $\forall \vec{x}\, \forall y \leq t(\vec{x})\ (f_\beta(\vec{x}, y) = 1 \rightarrow f_\beta(\vec{x}, g_{\beta,t}(\vec{x})) = 1)$ $(*)$. Furthermore, $(*)$ is a true universal sentence in the standard model by the definition of $g^{\mathbb{N}}_{\beta,t}$ and $f^{\mathbb{N}}_\beta$. Therefore $\mathsf{U}^i_{\mathsf{PV}}$ proves $(*)$.
- Case 2: By the definition of $g^{\mathbb{N}}_{\beta,t}$, the universal sentence $\forall \vec{x}\, g_{\beta,t}(\vec{x}) \leq t(\vec{x})$ is true in the standard model, which further means that $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{x}\, g_{\beta,t}(\vec{x}) \leq t(\vec{x})$. This sentence logically implies the sentence we want to prove in $\mathsf{U}^i_{\mathsf{PV}}$. □

Reminder of Lemma 2.13. *For every $i \geq 1$ and $(\Pi^b_{i-1} \cup \Sigma^b_{i-1})$-formula $\alpha(\vec{z})$ in the language $\mathcal{L}_{\mathsf{PV}}$, $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (\alpha(\vec{z}) \leftrightarrow f_\alpha(\vec{z}) = 1)$.*

Proof. Fix any $i \geq 1$. Let $\varphi_\alpha \triangleq \forall \vec{z}\, (\alpha(\vec{z}) \leftrightarrow f_\alpha(\vec{z}) = 1)$. We firstly prove that $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi_\alpha$ for every bounded $\mathcal{L}_{\mathsf{PV}}$-formula $\alpha(\vec{z}) = Q_1 x_1 \leq t_1(\vec{z})\ Q_2 x_2 \leq t_2(\vec{z}) \ldots Q_k x_k \leq t_k(\vec{z})\ \phi(\vec{z}, x_1, \ldots, x_k)$, where ϕ is quantifier free, $k \leq i - 1$, $Q_i \in \{\forall, \exists\}$, and $Q_i \neq Q_{i+1}$ for every $i \in [k - 1]$. We will prove this by induction over k.

- **(Case 0).** Assume that $k = 0$ and $\alpha(\vec{z})$ is a quantifier-free formula. Then φ_α is a universal sentence. Furthermore, by the definition of the interpretation of f_α over the standard model, we know that $\mathbb{N} \vDash \varphi_\alpha$, which means that $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi_\alpha$.

- **(Case 1).** Assume that $\alpha(\vec{z}) = \forall x \leq t(\vec{z})\, \alpha'(x, \vec{z})$. In such case, $i \geq 2$. By the induction hypothesis, $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi_{\alpha'}$. To show that $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi_\alpha$ it is sufficient to prove that $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (f_\alpha(\vec{z}) = 1 \to \alpha(\vec{z}))$ and $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (\alpha(\vec{z}) \to f_\alpha(\vec{z}) = 1)$. Now we prove them separately.

  (i) Since $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi_{\alpha'}$, we know that $\mathsf{U}^i_{\mathsf{PV}}$ proves $\forall \vec{z}\, \forall x \leq t(\vec{z})\, (f_{\alpha'}(x, \vec{z}) = 1 \to \alpha'(x, \vec{z}))$ $(\star)$. Consider the universal sentence $\psi \triangleq \forall \vec{z}\, \forall x \leq t(\vec{z})\, (f_\alpha(\vec{z}) = 1 \to f_{\alpha'}(x, \vec{z}) = 1)$. By the definition of the interpretations of f_α and $f_{\alpha'}$, ψ is a true sentence, therefore $\mathsf{U}^i_{\mathsf{PV}} \vdash \psi$ $(\Diamond)$. Combining $(\star)$ and $(\Diamond)$ we get that

  $$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, \forall x \leq t(\vec{z})\, (f_\alpha(\vec{z}) = 1 \to \alpha'(x, \vec{z})).$$

  This means that $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (f_\alpha(\vec{z}) = 1 \to \alpha(\vec{z}))$.

  (ii) Recall that we need to show that $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, ((\forall x \leq t(\vec{z})\, \alpha'(x, \vec{z})) \to f_\alpha(\vec{z}) = 1)$. Since $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi_{\alpha'}$, it is sufficient to prove that

  $$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, ((\forall x \leq t(\vec{z})\, f_{\alpha'}(\vec{z}, x) = 1) \to f_\alpha(\vec{z}) = 1).$$

  Since α is a Π^b_{i-1}-formula of the form above, $\neg\alpha'$ is a $\Sigma^b_{i-2} \cup \Pi^b_{i-2}$-formula. By Lemma D.2, $\mathsf{U}^i_{\mathsf{PV}} \vdash f_{\alpha'}(\vec{z}, x) = 1 \leftrightarrow f_{\neg\alpha'}(\vec{z}, x) = 0$. So we only need to prove that

  $$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, ((\forall x \leq t(\vec{z})\, f_{\neg\alpha'}(\vec{z}, x) = 0) \to f_\alpha(\vec{z}) = 1).$$

  By Lemma 2.12, we know that $\mathsf{U}^i_{\mathsf{PV}} \vdash (\exists x \leq t(\vec{z})\, f_{\neg\alpha'}(\vec{z}, x) = 1) \leftrightarrow f_{\neg\alpha'}(\vec{z}, g_{\neg\alpha',t}(\vec{z})) = 1$, which means we only need to prove that

  $$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (f_{\neg\alpha'}(\vec{z}, g_{\neg\alpha',t}(\vec{z})) = 0 \to f_\alpha(\vec{z}) = 1). \tag{18}$$

  By considering the interpretations of f_α, $f_{\neg\alpha'}$, and $g_{\neg\alpha',t}$ in the standard model, it follows that the universal sentence (18) is true in the standard model. Therefore it is provable in $\mathsf{U}^i_{\mathsf{PV}}$. This completes this case.

- **(Case 2).** Assume that $\alpha(\vec{z}) = \exists x \leq t(\vec{z})\, \alpha'(x, \vec{z})$. Let $\overline{\alpha}(\vec{z})$ be the formula obtained by pushing the negation in $\neg\alpha(\vec{z})$ into the quantifiers. Note that $\vdash \neg\alpha(\vec{z}) \leftrightarrow \overline{\alpha}(\vec{z})$. By applying Case 1, we can show that

  $$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, \left(f_{\overline{\alpha}}(\vec{z}) = 1 \leftrightarrow \neg\alpha(\vec{z})\right).$$

  Since $\forall \vec{z}\, (f_{\overline{\alpha}}(\vec{z}) = 1 \leftrightarrow f_\alpha(z) \neq 1)$ is a universal sentence that is true in the standard model, we know that it is provable in $\mathsf{U}^i_{\mathsf{PV}}$, which further implies that

  $$\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (f_\alpha(\vec{z}) \neq 1 \leftrightarrow \neg\alpha(\vec{z})).$$

  This yields $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi_\alpha$.

Now we consider the case when α is an arbitrary $(\Pi^b_{i-1} \cup \Sigma^b_{i-1})$-formula. By Lemma D.1, we can see that $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (\alpha(\vec{z}) \leftrightarrow \alpha^{\mathsf{norm}}(\vec{z}))$. According the discussion above, we know that $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (\alpha^{\mathsf{norm}}(\vec{z}) \leftrightarrow f_{\alpha^{\mathsf{norm}}}(\vec{z}) = 1)$. Moreover, we have that $\mathsf{U}^i_{\mathsf{PV}} \vdash \forall \vec{z}\, (f_\alpha(\vec{z}) = 1 \leftrightarrow f_{\alpha^{\mathsf{norm}}}(\vec{z}) = 1)$, since this is a true universal sentence in the standard model. It follows from the provability of these three sentences that $\mathsf{U}^i_{\mathsf{PV}} \vdash \varphi_\alpha$, as desired. □

Reminder of Theorem 2.18. *For every $i \geq 1$, the theory $\mathsf{UT}^i_{\mathsf{PV}}$ satisfies the following properties:*

(i) $\mathsf{UT}^i_{\mathsf{PV}}$ *is a universal theory.*

(ii) *Every $\mathcal{L}^i_{\mathsf{PV}}$-sentence provable in $\mathsf{U}^i_{\mathsf{PV}}$ is also provable in $\mathsf{UT}^i_{\mathsf{PV}}$.*

(iii) *Every $\mathcal{L}_{\mathsf{PV}}$-sentence provable in $\mathsf{T}^i_{\mathsf{PV}}$ is also provable in $\mathsf{UT}^i_{\mathsf{PV}}$.*

(iv) *Let t be an arbitrary $\mathcal{L}^i_{\mathsf{UT}}$-term, and consider its interpretation $t^{\mathbb{N}} \colon \mathbb{N}^k \to \mathbb{N}$ over the standard model. Then $t^{\mathbb{N}} \in \mathsf{FP}^{\Sigma^p_{i-1}}$.*

(v) $\mathsf{UT}^i_{\mathsf{PV}}$ *is closed under if-then-else.*

(vi) $\mathsf{UT}^i_{\mathsf{PV}}$ *is sound, i.e., every sentence provable in $\mathsf{UT}^i_{\mathsf{PV}}$ is true over $\mathbb{N}$.*

Proof. We prove each item in turn.

(i) This is immediate from the definition of the theory.

(ii) Let φ be an $\mathcal{L}^i_{\mathsf{PV}}$-sentence provable in $\mathsf{U}^i_{\mathsf{PV}}$. It is enough to argue that every axiom of $\mathsf{U}^i_{\mathsf{PV}}$ is provable in $\mathsf{UT}^i_{\mathsf{PV}}$. Since $\mathsf{U}^i_{\mathsf{PV}}$ is the theory consisting of all universal true sentences (over the standard model) in $\mathcal{L}^i_{\mathsf{PV}}$, $\mathcal{L}^i_{\mathsf{PV}} \subseteq \mathcal{L}^i_{\mathsf{UT}}$, and $\mathsf{UT}^i_{\mathsf{PV}}$ is the theory of all universal sentences in $\mathcal{L}^i_{\mathsf{UT}}$ that are true in the standard model, the result is immediate.

(iii) Let φ be an $\mathcal{L}_{\mathsf{PV}}$-sentence provable in $\mathsf{T}^i_{\mathsf{PV}}$. It follows from Theorem 2.14 that φ is provable in $\mathsf{U}^i_{\mathsf{PV}}$. Consequently, the claim follows from the previous item.

(iv) This follows from Theorem 2.17, the definition of $\mathcal{L}^i_{\mathsf{UT}}$, and the closure of the functions in $\mathsf{FP}^{\Sigma^p_{i-1}}$ under composition.

(v) To show this, let $\varphi(x_1, \ldots, x_k)$ be a quantifier-free $\mathcal{L}^i_{\mathsf{UT}}$-formula, and consider $\mathcal{L}^i_{\mathsf{UT}}$-terms $t_1(x_1, \ldots, x_k)$ and $t_2(x_1, \ldots, x_k)$. We must prove that there exists an $\mathcal{L}^i_{\mathsf{UT}}$-term $t(x_1, \ldots, x_k)$ such that

$$\mathsf{UT}^i_{\mathsf{PV}} \vdash \big(t(\vec{x}) = t_1(\vec{x}) \wedge \varphi(\vec{x})\big) \vee \big(t(\vec{x}) = t_2(\vec{x}) \wedge \neg\varphi(\vec{x})\big). \tag{19}$$

Consider the interpretations of terms $t_1^{\mathbb{N}}, t_2^{\mathbb{N}} \colon \mathbb{N}^k \to \mathbb{N}$ over the standard model. Let $f \colon \mathbb{N}^k \to \mathbb{N}$ be the function defined as follows:

$$f(\vec{a}) = \begin{cases} t_1^{\mathbb{N}}(\vec{a}) & \text{if } \varphi^{\mathbb{N}}(\vec{a}) \text{ is true;} \\ t_2^{\mathbb{N}}(\vec{a}) & \text{otherwise.} \end{cases}$$

Since φ is a quantifier-free formula, thanks to Item (iii), it is easy to see that $f \in \mathsf{FP}^{\Sigma^p_{i-1}}$. Consequently, the corresponding function symbol $f_{\mathsf{UT}} \in \mathcal{L}^i_{\mathsf{UT}}$. Take t as f_{UT}. It follows from the definition of f and of t that, for every $\vec{a} \in \mathbb{N}^k$,

$$\mathbb{N} \models \big(t(\vec{a}) = t_1(\vec{a}) \wedge \varphi(\vec{a})\big) \vee \big(t(\vec{a}) = t_2(\vec{a}) \wedge \neg\varphi(\vec{a})\big).$$

Since the formula above is free of quantifiers, the definition of $\mathsf{UT}^i_{\mathsf{PV}}$ immediately yields Equation (19).

(vi) This is obvious from its definition. □

E The Counting Lemma: Existence of a Good Restriction

Notation. Recall that for $m \geq 1$, a set $S \subseteq \{0,1\}^{[m]}$, and a string $a \in \{0,1\}^I$, where $I \subseteq [m]$, we define the *restriction of* S *with respect to* a as the set

$$S\restriction_a \triangleq \{w \in S \mid w|_I = a\}.$$

For a non-empty set U and a set $S \subseteq U$, we define $\mathsf{dens}_U(S) \triangleq |S|/|U|$.

For simplicity of the exposition, we consider without loss of generality restrictions with respect to the first m_1 input bits. Let $S \subseteq \{0,1\}^m$, where $m = m_1 + m_2$. Suppose that $\mathsf{dens}_{\{0,1\}^m}(S) = \delta$. Now let $T \subseteq S$ be a set such that $\mathsf{dens}_S(T) > 2/3$. The following result appears implicit in [Kra11, Pic15a].

Lemma E.1 (Counting Lemma)**.** *Under these assumptions, there is* $a \in \{0,1\}^{m_1}$ *such that*

$$\frac{|S\restriction_a|}{2^{m_2}} \geq \frac{1}{100} \cdot \delta \quad \textit{and} \quad \frac{|T\restriction_a|}{|S\restriction_a|} \geq \frac{2}{3} - \frac{1}{100}. \tag{20}$$

Proof. Suppose this is not the case, i.e., for every $a \in \{0,1\}^{m_1}$, at least one of the two inequalities above does not hold. We use this to contradict $|T| > (2/3) \cdot |S| = (2/3) \cdot \delta \cdot 2^m$. Under the assumption, and using that $T\restriction_a \subseteq S\restriction_a$,

$$\begin{aligned}
|T| &= \sum_{a \in \{0,1\}^{m_1}} |T\restriction_a| \\
&\leq \sum_{a \in \{0,1\}^{m_1}} \left(\frac{1}{100} \cdot \delta \cdot 2^{m_2} + \left(\frac{2}{3} - \frac{1}{100} \right) |S\restriction_a| \right) \\
&= 2^{m_1+m_2} \cdot \frac{1}{100} \cdot \delta + \left(\frac{2}{3} - \frac{1}{100} \right) \cdot |S| \\
&= \frac{\delta}{100} \cdot 2^m + \left(\frac{2}{3} - \frac{1}{100} \right) \cdot \delta \cdot 2^m \\
&= \frac{2}{3} \cdot \delta \cdot 2^m.
\end{aligned} \tag{21}$$

This completes the proof. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il