Approximate Counting and the Provability of Combinatorial Principles

Oliver Korten* Jiatu Li† Igor C. Oliveira‡ Hanlin Ren§

November 3, 2026

Abstract

Understanding which axioms suffice for the provability of known results in algorithms and complexity and whether they can settle longstanding conjectures in these areas is a fundamental direction at the intersection of logic and computer science. Motivated by this research program and its applications, we investigate bounded arithmetic theories designed to serve as a logical foundation for *probabilistic reasoning*, the relation between these theories, and whether they can prove more theorems than their underlying base theory for *deterministic reasoning*.

In more detail, for nearly twenty years, Jeřábek's theory APC_1 [Jeř07a] has provided a standard framework for the formalization of results from theoretical computer science relying on probabilistic techniques. In a recent work, [CLOW26] introduced a (possibly weaker) theory called APX_1. This theory is able to capture a large set of probabilistic arguments using four simple axioms for *approximate counting*. Both theories extend Cook's theory PV_1 [Coo75] for deterministic polynomial-time reasoning.

Our main result establishes that, in the relativized setting, APX_1 does not prove the retraction weak pigeonhole principle (rwPHP). Consequently, APC_1 is not conservative over APX_1 with respect to relativized $\forall\Sigma^b_1$ sentences. Moreover, we establish that the unprovability of rwPHP is part of a more general phenomenon: for a large class of finite combinatorial principles, the existence of an *infinite model* of the negation of the principle *characterizes* its unprovability in APX_1. This result, which is analogous to a theorem of Riis [Rii94], implies that PV_1 and APX_1 prove exactly the same finite combinatorial principles of this form.

*Institute for Advanced Study. E-mail: korten@ias.edu

†Massachusetts Institute of Technology. E-mail: jiatuli@mit.edu

‡University of Warwick. E-mail: igor.oliveira@warwick.ac.uk

§Institute for Advanced Study and Max Planck Institute for Informatics. E-mail: h4n1in.r3n@gmail.com

Contents

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Results . . . 2
  - 1.2.1 LossyCode, Refuter(Yao), and Total Search Problems of APX_1 . . . 3
  - 1.2.2 Approximate Counting and Weak Pigeonhole Principles . . . 4
  - 1.2.3 Finite Combinatorial Principles, Infinite Models, and the Limits of APX_1 . . . 6
- 1.3 Techniques . . . 9

2 Preliminaries **13**
- 2.1 Basic Notation . . . 13
- 2.2 Bounded Arithmetic and Theory PV_1 . . . 13
- 2.3 Witnessing Theorems . . . 14
- 2.4 Unrelativized Combinatorial Principles . . . 15
  - 2.4.1 Refuter(Yao) . . . 15
  - 2.4.2 Pigeonhole Principles and Total Search Problems . . . 16

3 APX_1, APX_1^*, and Related Theories **16**
- 3.1 APX_1 and APX_1^* . . . 16
- 3.2 Relativized Theories and Relativized Principles . . . 18
- 3.3 $\mathsf{True}_\forall\mathsf{APX}_1$: True Universal Axioms of (Relativized) APX_1 . . . 20

4 LossyCode, Refuter(Yao), and Total Search Problems of APX_1 **20**
- 4.1 Equivalence Between LossyCode (rwPHP) and Refuter(Yao) . . . 21
- 4.2 Witnessing Theorem for the $\forall\Sigma_1^b(\mathsf{PV}_1)$-Consequences of APX_1 . . . 22

5 Approximate Counting and Weak Pigeonhole Principles **23**
- 5.1 Unprovability of Relativized rwPHP in APX_1 . . . 23
- 5.2 Unprovability of Relativized (d)wPHP in $\mathsf{True}_\forall\mathsf{APX}_1$ . . . 25
  - 5.2.1 A Lower Bound for dwPHP . . . 25
  - 5.2.2 A Lower Bound for wPHP . . . 28
- 5.3 Provability of (d)wPHP in APX_1^* . . . 30
  - 5.3.1 Basic Counting Functionalities in APX_1^* . . . 30
  - 5.3.2 An Upper Bound for dwPHP . . . 32
  - 5.3.3 An Upper Bound for wPHP . . . 32

6 Finite Combinatorial Principles, Infinite Models, and the Limits of APX_1 **34**
- 6.1 Winning Strategies from Infinite Models . . . 34
- 6.2 Auxiliary Results for the Reduction to Quantifier-Free Formulas . . . 39
- 6.3 Unprovability in $\mathsf{APX}_1(\mathcal{L})$ . . . 40
- 6.4 Finite Combinatorial Principles Provable in APX_1 . . . 43

A Auxiliary Results and Missing Formalizations **47**
- A.1 Standard Models and Admissible Models . . . 47
- A.2 Basic Counting Functionalities in APX_1^* . . . 48

1 Introduction

1.1 Context and Motivation

Randomness and probability theory have shaped many central developments in theoretical computer science. Probabilistic techniques underpin the design and analysis of efficient algorithms [MR95], establish the existence of combinatorial objects such as expanders and error-correcting codes [Sha48, HLW06, AS16], and enable advances in cryptography [GM84] and proof verification [AS98]. They also provide powerful tools for establishing computational limitations, as illustrated by the use of random restrictions to prove lower bounds on the size of Boolean circuits [FSS84, Hås86].

These applications raise a natural foundational question: which axioms suffice to formalize probabilistic reasoning? In a seminal work, Jeřábek [Jeř07a] introduced the bounded arithmetic theory APC_1 and developed a robust framework for formalizing results from theoretical computer science that rely on probabilistic techniques. For nearly two decades, APC_1 and its extensions have provided a standard setting for such formalizations. Its applications range from reasoning about randomized algorithms [Jeř07a, LC12], cryptography [HJLM26b], and learning [PS26], to proofs of a variety of circuit size lower bounds [MP20].

While the axioms of APC_1 *suffice* to prove these and many other results, it is unclear whether they are *necessary*. Understanding this question is particularly important in light of recent developments in logic, complexity theory, and cryptography. On the one hand, there is a growing body of work on the unprovability of conjectures of interest in bounded arithmetic theories. Here, meaningful results arise when a theory T formalizes a broad range of proofs yet remains amenable to techniques for establishing unprovability results (see, e.g., [LO23, Oli25, ABM25] and references therein). On the other hand, recent work has revealed unexpected connections between unprovability results and the construction of advanced cryptographic primitives [Ila25a, HJLM26b, HJLM26a]. Weakening the theory underlying the unprovability assumption strengthens these results and makes the security of the resulting cryptographic primitives more plausible.

Recently, Chen, Li, Oliveira, and Williams [CLOW26] investigated whether probabilistic arguments from theoretical computer science could be formalized using weaker axioms. They introduced the theory APX_1 as a base theory for feasible probabilistic reasoning and showed that it supports formalizations of several influential results from algorithms and complexity theory, including the lower bound $\mathsf{PARITY} \notin \mathrm{AC}^0$ [FSS84], the correctness of the BLR linearity test [BLR93], and the Schwartz-Zippel lemma [Sch80, Zip79]. We refer to their paper for further applications of APX_1 to logic and complexity theory.

Both APX_1 and APC_1 extend Cook's theory PV_1 [Coo75], a first-order theory of polynomial-time reasoning whose language contains function symbols for polynomial-time algorithms. While APC_1 relies on the *dual weak pigeonhole principle* (dwPHP), which asserts that a function from a finite set to a set twice as large cannot be surjective, APX_1 builds its probabilistic toolkit from four natural axioms for a fresh *approximate counting* function symbol P. [CLOW26] showed that APC_1 simulates APX_1 for statements in the language of PV_1, and that APX_1 does not prove dwPHP under cryptographic and complexity-theoretic assumptions. Thus, under plausible computational assumptions, APX_1 is strictly weaker than APC_1, at least for statements of higher quantifier complexity such as dwPHP.

> *Remark* 1.1 (Theory APX_1 [CLOW26]). This theory extends $\mathsf{PV}_1(\mathsf{P})$, which allows polynomial-time computations to query an approximate counting function $\mathsf{P}(C, \Delta)$. Its intended output is an estimate of the acceptance probability of a Boolean circuit C under a random input, with additive error $\leq 1/\|\Delta\|$, where $\|\Delta\|$ denotes bit length. Its four axioms postulate that estimates lie in $[0, 1]$, agree with direct evaluation for circuits that do not read their inputs, approximately agree across precision parameters, and approximately respect averaging over the two restrictions obtained by fixing one input bit of C. See Section 3 for the formal definition.

In this work, we investigate the gap between PV_1, APX_1, and APC_1, and the strength of APX_1 and its extensions with respect to the provability of combinatorial principles of lower quantifier complexity.

Specifically, we are interested in the provability of $\forall\Sigma^b_1(\mathsf{PV}_1)$ sentences, i.e., statements of the form[1]

$$\forall x\, \exists y \leq \mathrm{poly}(\|x\|)\ \phi(x, y), \tag{1.1}$$

where $\phi(x, y)$ is a quantifier-free formula in the language of PV_1. Such statements express the totality of search problems in TFNP (see, e.g., [Kra95, Chapter 7] and [Pap94, Section 10.3]), an active research area with several connections to algorithms and complexity theory (see, e.g., [dRGR22, Tha24]). We ask which such problems are provably total in APX_1, and how they compare with those of PV_1 and APC_1.

This investigation is also motivated by the desire to understand the power of randomness in proofs. While the computational power of randomness is central to the P versus BPP question, asking whether APX_1 proves more combinatorial principles than PV_1 provides an analogous question in proof complexity. In other words, it asks whether access to a framework for probabilistic reasoning yields existence proofs beyond those available in the underlying theory of deterministic computation.

In the context of APX_1 versus APC_1, a result of Thapen [Tha02] (see [Jeř04, Proposition 1.14]) implies that APX_1 and APC_1 have the same $\forall\Sigma^b_1(\mathsf{PV}_1)$ consequences if and only if APX_1 proves the *retraction weak pigeonhole principle* (rwPHP). This combinatorial principle asserts the totality of the search problem LossyCode, which has attracted recent attention in complexity theory [Kor22, LPT24, CLO24, LLR26, FGJ+26]. Given Boolean circuits $\mathrm{Enc}\colon \{0,1\}^n \to \{0,1\}^{n-1}$ and $\mathrm{Dec}\colon \{0,1\}^{n-1} \to \{0,1\}^n$, the task is to find x such that $\mathrm{Dec}(\mathrm{Enc}(x)) \neq x$. Thus a solution witnesses the failure of a proposed compression scheme to preserve every input.

Whether APX_1 proves rwPHP was one of the main questions left open in [CLOW26]. In this direction, they showed that every TFNP problem provably total in APX_1 reduces in polynomial time to a problem called Refuter(Yao). Although a reduction from Refuter(Yao) to LossyCode was known, the converse remained open. Moreover, APX_1 was not shown to prove the totality of Refuter(Yao) (i.e., Equation (1.1) for the relevant formula ϕ), leaving open whether this computational problem is harder than the provably total search problems of APX_1.

1.2 Results

Our contributions can be summarised as follows:

1. First, we show that Refuter(Yao) and LossyCode (corresponding to the principle rwPHP) are equivalent under polynomial-time reductions, and that their totality statements are equivalent over PV_1. We also describe a complete problem for the TFNP consequences of APX_1 by applying the framework of Thapen [Tha24]. This problem, called $\mathsf{Local}(\mathsf{APX}_1)$, reduces to Refuter(Yao) and LossyCode.

2. Then, we investigate the provability of rwPHP and other pigeonhole principles in APX_1 and its extensions:

   - We obtain unconditional results in the relativized setting, showing that dwPHP and the weak pigeonhole principle wPHP are not provable in APX_1. These unprovability results extend to the true universal theory $\mathsf{True}_\forall\mathsf{APX}_1$. In contrast, rwPHP is provable in $\mathsf{True}_\forall\mathsf{APX}_1$.
   - Through a substantially more involved argument, we show that rwPHP is not provable in relativized APX_1. Thus, in the relativized setting, APC_1 is not $\forall\Sigma^b_1$-conservative over APX_1.
   - Complementing these results, we show that wPHP and dwPHP are provable in a multiplicative extension APX^*_1 of APX_1, which allows approximate counting with relative rather than additive precision. Together with the preceding lower bounds, this separates APX_1 from APX^*_1 in the relativized setting.

[1] To avoid confusion with the absolute value $|\cdot|$, in this paper we use $\|x\|$ to denote the bit-length of a binary string x.

3. Finally, we establish that the unprovability of rwPHP in relativized APX_1 is part of a more general phenomenon: for the class of finite combinatorial principles specified in Section 1.2.3, the existence of an *infinite model* of the negation of a principle *characterizes* its unprovability in APX_1. This result, analogous to a theorem of Riis [Rii94], applies to rwPHP and to many other natural combinatorial principles. Furthermore, it allows us to show that, in the relativized setting, PV_1 and APX_1 prove exactly the same finite combinatorial principles of this form.

In the following sections, we explain our results in detail.

1.2.1 LossyCode, Refuter(Yao), and Total Search Problems of APX_1

We first recall the search problem $\mathsf{Refuter}(\mathsf{Yao})$ introduced in [CLOW26]. An instance consists of a circuit $G\colon (\{0,1\}^n)^m \to [n] \times \{0,1\}^s$, called a *predictor generator*, that takes a list $\mathcal{D}$ of m strings in $\{0,1\}^n$ and outputs an index $i \in \{1,\dots,n\}$ and a circuit $P\colon \{0,1\}^{i-1} \to \{0,1\}$ of description length at most s. For an advantage parameter $\delta > 0$, the task is to find a list $\mathcal{D}$ such that, writing $(i,P) = G(\mathcal{D})$,

$$\Pr_{x \leftarrow \mathcal{D}}\big[P(x_{<i}) = x_i\big] < \frac{1}{2} + \delta,$$

where x is a uniformly chosen entry of the list and $x_{<i}$ denotes its first $i-1$ bits. Thus a solution refutes the claim that G always produces a predictor with advantage at least δ for the distribution it receives. The parameters satisfy $(\delta^2/10)m \geq s + \lceil \log n \rceil + 1$, which guarantees totality. Our first result identifies the complexity of this problem with that of $\mathsf{LossyCode}$.

Theorem 1.2 ($\mathsf{Refuter}(\mathsf{Yao})$ and $\mathsf{LossyCode}$)**.** *The problems* $\mathsf{Refuter}(\mathsf{Yao})$ *and* $\mathsf{LossyCode}$ *are equivalent under deterministic polynomial-time mapping reductions. Moreover, their totality statements are equivalent over* PV_1.

This equivalence does not settle whether APX_1 proves the totality of $\mathsf{Refuter}(\mathsf{Yao})$. As recalled above, every TFNP problem provably total in APX_1 reduces in polynomial time to $\mathsf{Refuter}(\mathsf{Yao})$ [CLOW26], but this witnessing result does not establish that $\mathsf{Refuter}(\mathsf{Yao})$ itself is provably total in APX_1.

To obtain a problem that captures precisely the TFNP consequences of APX_1, we apply Thapen's framework [Tha24] and define $\mathsf{Local}(\mathsf{APX}_1)$. An instance is a sequence of circuits describing an inspector's strategy for a finite interaction with a purported approximate counting oracle. In each round, the inspector uses the previous answers to choose a query (C,Δ), and the oracle responds with a rational number of the prescribed bit-length. A solution is a transcript of answers that agree on repeated queries and violate no counting axiom whose oracle values are all present in the transcript. The inspector therefore fails to expose an inconsistency. Both the length of a transcript and the time needed to verify it are polynomial in the size of the instance. We refer to Section 4.2 for the formal definition of $\mathsf{Local}(\mathsf{APX}_1)$.

Theorem 1.3 (A complete problem for the TFNP consequences of APX_1)**.** *The theory* APX_1 *proves the totality of* $\mathsf{Local}(\mathsf{APX}_1)$. *Moreover, every* TFNP *problem provably total in* APX_1 *reduces to* $\mathsf{Local}(\mathsf{APX}_1)$ *by a deterministic polynomial-time mapping reduction.*

Writing $\mathsf{TFNP}(\mathsf{APX}_1)$ for the TFNP problems provably total in APX_1, we can summarize these results as follows:

$$\mathsf{TFNP}(\mathsf{APX}_1) \overset{\text{Theorem 1.3}}{\equiv_p} \mathsf{Local}(\mathsf{APX}_1) \leq_p \big(\mathsf{Refuter}(\mathsf{Yao}) \overset{\text{Theorem 1.2}}{\equiv_p} \mathsf{LossyCode}\big).$$

Here the relations $\leq_p$ and $\equiv_p$ refer to deterministic polynomial-time mapping reductions. The reduction from $\mathsf{Local}(\mathsf{APX}_1)$ to $\mathsf{Refuter}(\mathsf{Yao})$ follows by applying the witnessing theorem of [CLOW26] to the totality of $\mathsf{Local}(\mathsf{APX}_1)$.[2] We prove the two theorems above in Section 4.

These results leave open whether $\mathsf{APX}_1 \vdash^? \mathsf{rwPHP}$ and whether the reverse reduction from $\mathsf{LossyCode}$ to $\mathsf{Local}(\mathsf{APX}_1)$ exists, making all three problems equivalent under polynomial-time reductions. A proof of rwPHP in APX_1 would imply such a reduction by the aforementioned witnessing theorem.

Establishing the unprovability of rwPHP in APX_1 under computational assumptions appears difficult. A standard approach is to use witnessing theorems to extract an algorithm from a hypothetical proof and then contradict a hardness assumption. For example, randomized witnessing rules out the provability of wPHP in APX_1 assuming the existence of collision resistant hash functions. A more sophisticated witnessing argument from [ILW23] rules out dwPHP under sub-exponentially secure indistinguishability obfuscation together with the assumption that coNP is not contained infinitely often in $\mathsf{NP/poly}$ [CLOW26]. However, randomized witnessing applied to a hypothetical proof of rwPHP would yield a randomized algorithm for $\mathsf{LossyCode}$ running in expected polynomial time, and such an algorithm already exists. Indeed, at most 2^{n-1} strings $x \in \{0,1\}^n$ can satisfy $\mathrm{Dec}(\mathrm{Enc}(x)) = x$, since Enc is injective on the set of correctly decoded strings. A uniformly random x is therefore a solution with probability at least $1/2$, and repeated sampling with verification finds and confirms a solution in expected polynomial time. Consequently, the existence of the extracted algorithm provides no contradiction.

This limitation motivates our investigation of unconditional unprovability results in the relativized setting, to which we turn next.

1.2.2 Approximate Counting and Weak Pigeonhole Principles

Before stating our next results, we explain how rwPHP and APX_1 can be relativized in a natural way.

Remark 1.4 (Relativizations of APX_1 and rwPHP). Relativization replaces the explicit compression and decompression circuits in rwPHP by fresh function symbols f and g. The relativized retraction principle asserts that a map f from $[2N]$ to $[N]$ cannot have a left inverse g, where $[N] = \{0, \ldots, N-1\}$, and we typically think of $N = 2^n$. This is captured by the sentence

$$\mathsf{rwPHP}(f,g) \triangleq \forall N \left(N > 1 \rightarrow \exists x < 2N \left(f(x) \geq N \vee g(f(x)) \neq x\right)\right).$$

It is also possible to formulate the principle using binary strings, in which case the oracle f maps n bits to $n-1$ bits, the oracle g maps $n-1$ bits to n bits, and the principle asserts that some $x \in \{0,1\}^n$ satisfies $g(f(x)) \neq x$. The chosen formulation is inessential in our results.

The theory $\mathsf{APX}_1(f,g)$ allows these same oracles both in the underlying polynomial-time computations and in the Boolean circuits submitted to P. Thus $\mathsf{P}(C,\Delta)$ is intended to approximate $\Pr_z[C^{f,g}(z) = 1]$ with additive error at most $1/\|\Delta\|$. Formally, the theory extends $\mathsf{PV}_1(\mathsf{P},f,g)$ by the four approximate counting axioms, now applied to circuits with oracle gates for f and g. We refer to Section 3 for the details.

We obtain the following unconditional result.[3]

Theorem 1.5 (Unprovability of the retraction weak pigeonhole principle)**.**

$$\mathsf{APX}_1(f,g) \nvdash \mathsf{rwPHP}(f,g).$$

[2] The result stated in [CLOW26] provides a Turing reduction, but one can check that a mapping reduction suffices. In fact, it is not hard to show that every TFNP problem that admits a deterministic polynomial-time Turing reduction to $\mathsf{Refuter}(\mathsf{Yao})$ also admits a deterministic polynomial-time mapping reduction to $\mathsf{Refuter}(\mathsf{Yao})$.

[3] In a concurrent work, Chen, Jain, Li, and Roy [CJLR26] showed that under $\mathsf{prBPP} = \mathsf{prP}$, LWE, and an assumption related to the unprovability of (unrelativized) rwPHP from APX_1 augmented with non-logical axioms ($\mathsf{prBPP} = \mathsf{prP}$ and LWE), there is a new construction of SNARGs for NP. Theorem 1.5 provides evidence for $\mathsf{APX}_1 \nvdash \mathsf{rwPHP}$ and shows that the assumption from [CJLR26] is unlikely to be trivially false.

This theorem identifies a limitation of the approximate counting axioms. They support an internally consistent notion of size, but do not suffice to establish the expected lower bound on the size of the set of inputs on which decoding fails. Indeed, although this set contains at least half of all inputs in every standard interpretation, the theory cannot prove that it is non-empty for every input length. At the same time, the formalizations in [CLOW26] show that these axioms support many substantial probabilistic arguments from algorithms and complexity theory.

We derive Theorem 1.5 from a more general result about finite combinatorial principles, discussed in the next subsection.

Both wPHP and dwPHP imply rwPHP already over PV_1, and the implications hold with oracle functions as well. This is because a collision in f forces decoding to fail on at least one of the two colliding inputs, while a point outside the range of g cannot be decoded correctly. It is therefore useful to study the (un)provability of these stronger principles in APX_1 and its extensions. Their unprovability proofs provide a starting point for our argument, while also revealing why the retraction principle requires additional ideas.

For this comparison, we introduce two extensions of APX_1 and their natural relativizations. The first one is the true universal theory of APX_1, denoted $\mathsf{True}_\forall\mathsf{APX}_1$.

Remark 1.6 (True relativized universal theories $\mathsf{True}_\forall\mathsf{APX}_1$). The theory $\mathsf{True}_\forall\mathsf{APX}_1(f,g)$ extends $\mathsf{PV}_1(\mathsf{P},f,g)$ by every universal sentence that holds in every standard interpretation of these symbols. Here a universal sentence is the universal closure of a quantifier-free formula. In a standard interpretation, f and g are arbitrary functions of the prescribed lengths, and P is any correct additive approximate counting oracle for circuits with access to f and g. The remaining function symbols are interpreted by their defining oracle algorithms. The four axioms of $\mathsf{APX}_1(f,g)$ are among these universal truths, so $\mathsf{True}_\forall\mathsf{APX}_1(f,g)$ extends $\mathsf{APX}_1(f,g)$. We define $\mathsf{True}_\forall\mathsf{APX}_1(f)$ and $\mathsf{True}_\forall\mathsf{APX}_1(g)$ similarly. See Definition 3.5 and Definition 3.6 for the precise definitions.

We also introduce a natural variant of APX_1, denoted by APX_1^*, designed to support a stronger form of approximate counting.

Remark 1.7 (Multiplicative approximate counting and APX_1^*). The theory APX_1^* strengthens the precision and local consistency axioms so that the intended counting estimates have relative error. More precisely, if $p(C)$ is the true acceptance probability of C, the intended guarantee is

$$(1-\delta)\,p(C) \le \mathsf{P}(C,\Delta) \le (1+\delta)\,p(C), \quad \text{where } \delta = \frac{1}{\|\Delta\|}.$$

This gives useful precision even when $p(C)$ is very small. The resulting axioms imply those of APX_1.

The relativized theories $\mathsf{APX}_1^*(f)$, $\mathsf{APX}_1^*(g)$, and $\mathsf{APX}_1^*(f,g)$ are defined by allowing the corresponding oracle gates in counting queries and oracle calls in the underlying computations, just as for APX_1.

We obtain the following positive and negative results.

Theorem 1.8 ((Un)provability of pigeonhole principles in extensions of APX_1)**.** *The following statements hold:*

1. $\mathsf{True}_\forall\mathsf{APX}_1(f,g) \vdash \mathsf{rwPHP}(f,g)$.
2. $\mathsf{True}_\forall\mathsf{APX}_1(g) \nvdash \mathsf{dwPHP}(g)$.
3. $\mathsf{True}_\forall\mathsf{APX}_1(f) \nvdash \mathsf{wPHP}(f)$.
4. $\mathsf{APX}_1^*(g) \vdash \mathsf{dwPHP}(g)$. *In particular,* $\mathsf{APX}_1^* \vdash \mathsf{dwPHP}$.
5. $\mathsf{APX}_1^*(f) \vdash \mathsf{wPHP}(f)$. *In particular,* $\mathsf{APX}_1^* \vdash \mathsf{wPHP}$.

The first item of Theorem 1.8 highlights a distinction essential to Theorem 1.5: adding true universal axioms already suffices to prove rwPHP. To see why, consider the oracle circuit $C^{f,g}$ that accepts precisely when $g(f(x)) \neq x$. Its acceptance probability is at least $1/2$, and the corresponding lower bound on its approximate count is a true universal statement. With this estimate available, $\mathsf{APX}_1(f, g)$ can prove the existence of an input x on which decoding fails. In contrast, the unprovability results for wPHP and dwPHP in the second and third items of Theorem 1.8 remain valid even after all true universal axioms are added.

A consequence is that, in the case of wPHP and dwPHP, the corresponding lower bound argument can provide genuine approximate counting answers as the adversary constructs the oracle showing unprovability. On the other hand, our adversary for rwPHP must preserve the axiom instances exposed during the interaction without requiring the counting answers to be accurate. Exploiting the distinction between the approximate counting axioms and their intended interpretation is a central difficulty in establishing Theorem 1.5.

The remaining items of Theorem 1.8 separate additive and multiplicative approximate counting in the relativized setting. The theory $\mathsf{APX}_1^*(f)$ proves $\mathsf{wPHP}(f)$, which is unprovable even in $\mathsf{True}_\forall\mathsf{APX}_1(f)$, and $\mathsf{APX}_1^*(g)$ proves $\mathsf{dwPHP}(g)$, which is unprovable even in $\mathsf{True}_\forall\mathsf{APX}_1(g)$. Since the multiplicative theory extends the additive theory, it follows that in the relativized setting, APX_1^* is strictly stronger than APX_1. We refer to Section 5.3 for some computational consequences of Theorem 1.8 that might be of independent interest.

The more delicate argument for rwPHP is part of a general connection between unprovability in APX_1 and the existence of infinite models. We turn to this connection next.

1.2.3 Finite Combinatorial Principles, Infinite Models, and the Limits of APX_1

Our next results connect the provability of finite combinatorial principles with their behavior on infinite structures. For a natural class of principles that hold in every non-empty finite structure, we show that their finite formulations are unprovable in relativized APX_1 exactly when they admit an infinite counterexample. When no such counterexample exists, the finite formulation is already provable in the corresponding relativization of PV_1. Thus, for this class of principles, the approximate counting axioms yield no additional proving power.

For example, an infinite counterexample to rwPHP is obtained on $\mathbb{N}$ by defining

$$f_0(x) = 2x, \qquad f_1(x) = 2x + 1, \qquad h(y) = \lfloor y/2 \rfloor.$$

The functions f_0 and f_1 encode two disjoint copies of $\mathbb{N}$ into a single copy: one uses the even numbers, and the other uses the odd numbers. Decoding recovers the original copy from the parity of the encoded value and the original element by applying h. Thus every input is recovered correctly. This gives an infinite counterexample to the retraction principle.

To formally state the results of this section, we first need to introduce the relevant terminology.

$\mathcal{L}$-Principles, $\widehat{\mathcal{L}}$-Translations, and $\mathsf{APX}_1(\widehat{\mathcal{L}})$. Let $\mathcal{L}$ be a non-empty finite language consisting of function and constant symbols and disjoint from the language of APX_1. Let $\mathcal{C}$ be its set of constant symbols and $\mathcal{F}$ its set of function symbols.

In order to discuss the provability of finite combinatorial principles expressed as $\mathcal{L}$-statements but restricted to a domain $[N]$, it will be convenient to replace each function symbol $f \in \mathcal{F}$ of arity d by a symbol $\widehat{f}(s, x_1, \ldots, x_d)$, and each constant $c \in \mathcal{C}$ by a unary symbol $\widehat{c}(s)$. Write $\widehat{\mathcal{L}}$ for the resulting language. Now define the following formula over the language $\widehat{\mathcal{L}} \cup \{<\}$:

$$\mathrm{Closed}_{\widehat{\mathcal{L}}}(N) \triangleq \bigwedge_{c \in \mathcal{C}} \widehat{c}(N) < N \ \wedge \ \bigwedge_{f \in \mathcal{F}} \forall u_1, \ldots, u_{\mathrm{arity}(f)} < N \ \widehat{f}(N, u_1, \ldots, u_{\mathrm{arity}(f)}) < N,$$

where $<$ is the PV_1 symbol corresponding to the order relation $<$. This formula will be useful when we consider $\mathcal{L}$-structures on $[N] = \{0, \ldots, N-1\}$.

Let $\mathsf{APX}_1(\widehat{\mathcal{L}})$ denote the relativized theory in which the symbols of $\widehat{\mathcal{L}}$ are available as oracles both to polynomial-time computations and to the circuits submitted to the approximate counting oracle $\mathsf{P}(C, \Delta)$, with the length axioms

$$\|\widehat{f}(s, \bar{x})\| \leq \|s\| \quad \text{and} \quad \|\widehat{c}(s)\| \leq \|s\|$$

included before admitting derived functions in the Cobham's style definition of relativized $\mathsf{PV}_1(\mathsf{P}, \widehat{\mathcal{L}})$. The notation $\mathsf{PV}_1(\widehat{\mathcal{L}})$ uses the same parameterized oracles and length axioms, with the counting oracle omitted.

Finally, for an $\mathcal{L}$-formula ψ, write $\psi^{[N]}$ for its *finitary translation at size* N (where N is a fresh variable), which is defined recursively by

$$x^{[N]} \triangleq x, \qquad c^{[N]} \triangleq \widehat{c}(N), \qquad \big(f(t_1, \ldots, t_d)\big)^{[N]} \triangleq \widehat{f}(N, t_1^{[N]}, \ldots, t_d^{[N]}),$$
$$\big(\forall x\ \phi\big)^{[N]} \triangleq \forall x < N\ \phi^{[N]}, \qquad \big(\exists x\ \phi\big)^{[N]} \triangleq \exists x < N\ \phi^{[N]}.$$

and commutes with equality and Boolean connectives. All references below to infinite structures concern the original language $\mathcal{L}$.

We are now ready to state the main result of this section. In the statement below, we think of a tuple $(x_1, \ldots, x_k)$ as a potential solution to the search problem encoded by ϕ.

Theorem 1.9 (Riis-style dichotomy for relativized APX_1)**.** *Let $\phi(x_1, \ldots, x_k)$ be a formula in the language $\mathcal{L}$ with equality. If $\forall x_1 \cdots \forall x_k\ \neg\phi(x_1, \ldots, x_k)$ has an infinite model, then*

$$\mathsf{APX}_1(\widehat{\mathcal{L}}) \nvdash \forall N \geq 1\ \Big(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N)\ \rightarrow\ \exists x_1, \ldots, x_k < N\ \phi^{[N]}(x_1, \ldots, x_k)\Big).$$

Theorem 1.9 is an analogue of Riis' finitisation theorems [Rii94] (see also [Mül21] for a more recent related work). For sentences expressed in an additional relational language, Riis showed that provability of their finite restrictions in the relativized theories T_2^1 or S_2^2 is equivalent to validity in all structures for that relational language. Crucially, the proof of the finite restriction may use all the axioms of the ambient arithmetic theory, whereas an infinite counterexample need only interpret the additional combinatorial symbols, i.e., it need not carry any arithmetic structure or satisfy the axioms of the theory. Thus a purely combinatorial infinite counterexample can obstruct an arithmetic proof. Theorem 1.9 establishes an obstruction of the same form in the context of relativized APX_1. Note that this phenomenon does not extend to relativized APC_1, which proves rwPHP despite the existence of an infinite counterexample.

Remark 1.10 (Example: The weak pigeonhole principle). A map $f\colon [2N] \to [N]$ can be represented by two maps $f_0, f_1\colon [N] \to [N]$, defined by $f_0(x) = f(x)$ and $f_1(x) = f(N + x)$. Conversely, any such pair determines a map $f\colon [2N] \to [N]$. A collision in f therefore corresponds either to a collision within one of f_0, f_1, or to an intersection of their ranges. To capture the relativized weak pigeonhole principle, let $\mathcal{L} = \{f_0, f_1\}$ consist of two unary function symbols, and define

$$\phi(x, y) \triangleq \big(x \neq y\ \wedge\ \big(f_0(x) = f_0(y) \vee f_1(x) = f_1(y)\big)\big)\ \vee\ f_0(x) = f_1(y).$$

The sentence $\exists x, y\ \ \phi(x, y)$ asserts that f_0 and f_1 cannot both be injective with disjoint ranges. It holds in every non-empty finite structure. However, it fails in the infinite structure with universe $\mathbb{N}$ and interpretations $f_0(x) = 2x$ and $f_1(x) = 2x + 1$. Thus Theorem 1.9 gives

$$\mathsf{APX}_1(\widehat{\mathcal{L}}) \nvdash \forall N \geq 1\ \Big(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N)\ \rightarrow\ \exists x, y < N\ \phi^{[N]}(x, y)\Big).$$

Under the encoding above, the slices $\widehat{f}_0(N, \cdot)$ and $\widehat{f}_1(N, \cdot)$ represent the two halves of a map $f\colon [2N] \to [N]$. The

displayed sentence is therefore the relativized weak pigeonhole principle asserting that this map has a collision.

In the next example, we explain how a principle that relies on a relation symbol, in addition to function and constant symbols, can be reformulated by replacing it with a function symbol.

Remark 1.11 (Example: The Herbrandized ordering principle [AT14]). The Herbrandized ordering principle (HOP) asserts that a finite strict linear order cannot have a total immediate predecessor function: at a least element, such a function must fail. In the vocabulary $\alpha = (\prec, h)$ of [AT14], where $\prec$ is a binary relation symbol and h is a unary function symbol, its natural formulation for $h\colon [N] \to [N]$ is

$$\mathrm{LinOrd}_{\prec}(N) \;\to\; \exists x < N \;\big(\neg(h(x) \prec x) \;\vee\; \exists y < N \;(h(x) \prec y \wedge y \prec x)\big)\,,$$

where $\mathrm{LinOrd}_{\prec}(N)$ asserts that $\prec$ is a strict linear order on $[N]$. Thus a solution witnesses either a violation of the order axioms or a failure of $h(x)$ to be the immediate predecessor of x.

To express this principle in our framework, let $\mathcal{L} = \{h, r, c_T\}$, where h is a unary function, r is a binary function, and c_T is a constant. Abbreviate $r(x, y) = c_T$ by $x \prec y$. Now define

$$\begin{aligned}\phi(x,y,z) \triangleq\; & x \prec x \vee \big(x \neq y \wedge \neg(x \prec y) \wedge \neg(y \prec x)\big) \\ & \vee \big(x \prec y \wedge y \prec z \wedge \neg(x \prec z)\big) \\ & \vee \neg(h(x) \prec x) \vee \big(h(x) \prec y \wedge y \prec x\big).\end{aligned}$$

The first three disjuncts witness a failure of $\prec$ to be a strict linear order, while the last two express the remaining conditions from HOP. Hence $\exists x, y, z\ \phi(x, y, z)$ holds in every non-empty finite $\mathcal{L}$-structure. In the formula translation, the abbreviation $x \prec y$ becomes $\widehat{r}(N, x, y) = \widehat{c}_T(N)$.

An infinite counterexample is obtained on $\mathbb{N}$ by taking $c_T = 0$, $h(x) = x + 1$, and $r(x, y) = 0$ if $x > y$, and 1 otherwise. Then $\prec$ is the reverse of the usual order and $h(x)$ is always the immediate predecessor of x. Consequently,

$$\mathsf{APX}_1(\widehat{\mathcal{L}}) \nvdash \forall N \geq 1 \;\Big(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N) \;\to\; \exists x, y, z < N \;\phi^{[N]}(x, y, z)\Big)\,.$$

Theorem 1.9 clarifies the weakness of the approximate counting axioms of APX_1. While they support a broad range of probabilistic arguments [CLOW26], APX_1 cannot distinguish the finite from the infinite.

As a consequence of Theorem 1.9, for combinatorial principles of the above form that hold in every non-empty finite $\mathcal{L}$-structure, provability of their translations in $\mathsf{APX}_1(\widehat{\mathcal{L}})$ is *equivalent* to the absence of an infinite counterexample. Indeed, one obtains the following characterization.

Corollary 1.12 (Finite relativized combinatorial principles provable in APX_1)**.** *Let $\phi(x_1, \ldots, x_k)$ be any formula in $\mathcal{L}$ with equality, and suppose that $\exists \bar{x}\, \phi(\bar{x})$ holds in every non-empty finite $\mathcal{L}$-structure. Write*

$$\Theta_\phi \triangleq \forall N \geq 1 \;\Big(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N) \;\to\; \exists x_1, \ldots, x_k < N \;\phi^{[N]}(x_1, \ldots, x_k)\Big)\,.$$

Then the following are equivalent:

1. $\mathsf{APX}_1(\widehat{\mathcal{L}}) \vdash \Theta_\phi$.
2. $\mathsf{PV}_1(\widehat{\mathcal{L}}) \vdash \Theta_\phi$.
3. *$\exists \bar{x}\, \phi(\bar{x})$ holds in every infinite $\mathcal{L}$-structure.*

Corollary 1.12 indicates that in terms of the provability of a large class of combinatorial principles, APX_1 behaves more like its deterministic base theory PV_1 than like APC_1.

1.3 Techniques

We now explain the main ideas underlying our most technical results (Theorem 1.5 and Theorem 1.9). As a warm-up, we provide an overview of the unprovability of dwPHP in $\mathsf{True}_\forall\mathsf{APX}_1$ in the relativized setting (Theorem 1.8 Item 2).

Unprovability of dwPHP (Theorem 1.8 Item 2). Assume towards a contradiction that $\mathsf{True}_\forall\mathsf{APX}_1$ proves dwPHP in the relativized setting. Applying the KPT witnessing theorem [KPT91], we obtain a winning strategy for the "inspector" in the following "dwPHP game". The game is defined in terms of functions $g\colon [N] \to [2N]$ and a finite number of true universal sentences $\Psi_1, \Psi_2, \ldots, \Psi_\ell$ (i.e., universal sentences in the language of $\mathsf{PV}_1(g, \mathsf{P})$ that hold in every interpretation of g and any valid interpretation of the approximate counting oracle P), and consists of $m = \operatorname{poly}\log(N)$ rounds. In each round, the inspector can either make a query to g, make a query to P, or make an inversion query. Here, a query to P consists of an oracle circuit $C^{(-)}$ of size at most $s = \operatorname{poly}\log(N)$, and the adversary must return a value $p \in [0,1]$ as the purported value of $\Pr_z[C^g(z) = 1]$ within additive error $\varepsilon = 1/\operatorname{poly}\log(N)$. An inversion query consists of an element $y \in [2N]$, and the adversary must return a preimage $x \in [N]$ such that $g(x) = y$; if the adversary is unable to do so, the inspector wins. Finally, if the inspector is able to find a violation of one of the axioms Ψ_i, then the inspector also wins.

We will construct a winning strategy for the adversary as long as m, s, ε^{-1} are sufficiently small compared to N (which leads to the contradiction). If there were no P queries, then the adversary could win by constructing a partial function g lazily: For each query $g(x)$ such that x is not already in the current domain of g, we assign $g(x) = 0 \in [2N]$; for each inversion query y, pick a random element $x \in [N]$ outside the current domain of g and assign $g(x) = y$. As long as $m < N$, the domain size of g will be less than N and the adversary will not run out of preimages for inversion queries.

It remains to handle the P oracle queries. The idea is to fix a small pseudorandom set $\mathcal{S}$ such that for every small circuit C and every "valid" g,

$$\Pr_z[C^g(z) = 1] \approx \Pr_{z \leftarrow \mathcal{S}}[C^g(z) = 1],$$

and return $\Pr_{z\leftarrow\mathcal{S}}[C^g(z) = 1]$ as the estimate. During this process, we need to evaluate $C^g(z)$ for every $z \in \mathcal{S}$. Again, whenever we make a query $g(x)$ such that x is not already in the current domain of g, we assign $g(x) = 0$. At first glance, it may appear that this is self-defeating: during the game, the domain size of g is upper bounded by $|\mathcal{S}| \cdot m \cdot s$, so we want a pseudorandom set of size $|\mathcal{S}|$ that is pseudorandom against every g with this domain size; on the other hand, a standard union bound only gives a pseudorandom set of large size that is pseudorandom against functions with a smaller description. However, notice that we lazily assigned 0 to all g-values except in the inversion queries, hence the number of preimages mapped to a non-zero element in $[2N]$ is at most m. We say g is "valid" if there are at most m preimages that map to a non-zero element. It is then easy to obtain a pseudorandom set of size $|\mathcal{S}| = \operatorname{poly}(m, s, 1/\varepsilon)$ that ε-fools every C^g for small circuit C and valid g. As long as the domain size of g is $\operatorname{poly}(m, |\mathcal{S}|) < N$, the adversary wins the game.

We note that in the above proof, the adversary replies with *honest* probabilities in the following sense. If we let g^{final} be the function g at the end of the game, then every $\mathsf{P}(C)$ query is answered with an ε-approximation of $\Pr_z[C^{g^{\text{final}}}(z) = 1]$. Hence, these answers will satisfy any universal axiom Ψ_i and the adversary wins the game. Looking ahead, to show the unprovability of rwPHP in APX_1, the adversary cannot reply with honest probabilities, as $\mathsf{True}_\forall\mathsf{APX}_1$ indeed proves rwPHP (Theorem 1.8 Item 1).

Unprovability of rwPHP (Theorem 1.5). We now explain at a high level the unprovability of the retraction weak pigeonhole principle (rwPHP), which asserts for PV_1 function symbols $f\colon \{0,1\}^{n+1} \to \{0,1\}^n$

and $g\colon \{0,1\}^n \to \{0,1\}^{n+1}$ that there exists $x \in \{0,1\}^{n+1}$ such that $g(f(x)) \neq x$. The general Riis-style dichotomy follows a very similar argument.

We will establish using witnessing methods that if APX_1 proves rwPHP in the relativized setting, then there is a winning strategy for the "inspector" in the following "rwPHP game" (we will actually prove a broad generalization of this in Section 6). The rwPHP game is defined in terms of functions $f\colon [2N] \to [N]$ and $g\colon [N] \to [2N]$. The game proceeds for $m = \operatorname{poly}\log(N)$ rounds and in each round, the inspector supplies a distribution $\mathcal{T}_i$ over decision trees T of height $h \leq \operatorname{poly}\log(N)$ which query the functions f, g on various inputs and output a value in $\{0,1\}$. The adversary must reply with a value $p_i \in [0,1]$. At the end of the game, the inspector outputs a value $x \in [2N]$, and wins the game if $g(f(x)) \neq x$. The values p_i are supposed to represent $\Pr_{T\sim\mathcal{T}_i}[T(f,g) = 1]$. However, the adversary is allowed to deviate from this "honest answer," as long as they obey the following "local consistency" rules for some error parameter $\epsilon \geq (\operatorname{poly}\log(N))^{-1}$:

1. (*Deterministic Consistency*) If the inspector supplies a distribution supported on a single decision tree T_i, then the adversary must reply with the true probability $\Pr[T_i(f,g) = 1] \in \{0,1\}$.
2. (*Mixture Consistency*) For any three round indices i_1, i_2, i_3 such that $\mathcal{T}_{i_1} = \frac{1}{2}\mathcal{T}_{i_2} + \frac{1}{2}\mathcal{T}_{i_3}$, if the adversary responded with values $p_{i_1}, p_{i_2}, p_{i_3}$, it must be that $\left|p_{i_1} - \frac{1}{2}p_{i_2} - \frac{1}{2}p_{i_3}\right| \leq \varepsilon$.

Our goal is to construct an adversary that always wins when $m, h, \epsilon^{-1} \leq \operatorname{poly}\log(N)$. As discussed above, this lower bound cannot be proven by an adversary that always gives honest answers: In this case the game amounts to a standard randomized decision tree which aims to solve the search problem associated with rwPHP, which is easy using binary search (using the fact that a random x satisfies $g(f(x)) \neq x$ with probability $\geq \frac{1}{2}$). This highlights the subtlety of the lower bound: to construct our adversary, we must answer "fake probabilities" in such a way that (1) they preclude the determination of any particular solution to the rwPHP search problem, and (2) they are indistinguishable from honest probabilities according to the two local consistency rules.

Our strategy will be as follows. Over the course of the game, the adversary will maintain a partial matching ρ_i after round i has been completed. When the next tree distribution $\mathcal{T}_i$ is supplied, the adversary will take s independent samples $T_{i,1}, \ldots, T_{i,s} \sim \mathcal{T}_i$ (for some parameter $s = \operatorname{poly}(h, m, \epsilon^{-1})$ to be determined later). It then runs over each of the trees $T_{i,j}$ in order and does the following. Starting at the root of $T_{i,j}$, if the tree queries $f(x)$ and x is matched in ρ_i, we traverse the tree as if the answer to the query was the mate of x in ρ_i. If x is unmatched in ρ_i, we choose a uniformly random unmatched element $y \in [N]$, extend ρ_i to add the pair (x, y), and traverse the tree according to $f(x) = y$. We perform the symmetric operation for queries of the form $g(y)$. We can always perform an appropriate move provided $|\rho_i| + sh \leq N$ which we will make sure to maintain throughout the game. When all trees $T_{i,1}, \ldots, T_{i,s}$ have been traversed in order, we reach a leaf in each tree and hence determine the corresponding output bits $b_{i,1}, \ldots, b_{i,s} \in \{0,1\}$ of these trees at those leaves. The adversary then responds with the value $p_i = \frac{1}{s}\sum_j b_{i,j}$, and sets $\rho_{i+1} \supseteq \rho_i$ to be the extended matching obtained from all the extensions made to ρ_i during these traversals.

Now, when we reach the end of the game and ρ_m is the final matching, we will guarantee by our choice of parameters that $|\rho_m| < N$. The inspector must decide on some $x_{\text{final}} \in [2N]$ to output as their final answer; we extend ρ_m to ρ_{final} such that x_{final} is matched in ρ_{final}, which is possible since $|\rho_m| < N$. We then choose any pair of functions f, g such that for all $(x, y) \in \rho_{\text{final}}$, $f(x) = y$ and $g(y) = x$. We want to argue that with high probability over the randomness of the adversary (corresponding to the random matching choices and random trees sampled), this pair (f, g) will witness the failure of the inspector. First observe that, since $g(f(x_{\text{final}})) = x_{\text{final}}$, the inspector failed to find an rwPHP solution. Second, since $f(x) = y$ and $g(y) = x$ for all $(x, y) \in \rho_{\text{final}}$, the deterministic consistency rule is obeyed with probability 1. So it remains only to show that with high probability the mixture consistency rule is obeyed by the adversary's answers p_i.

To prove mixture consistency, we consider an alternate hypothetical sequence of answers p_i^{final} defined as the probability of outputting $b = 1$ in the following experiment for each i. Sample a fresh $T_i \sim \mathcal{T}_i$ and randomly traverse through T_i starting from ρ_{final}: each query $f(x)$ is answered according to ρ_{final} if x is matched in ρ_{final}, and is otherwise set to a uniform element of $[N]$ which is unmatched in the current extension of ρ_{final} obtained thus far. Queries of the form $g(y)$ are handled similarly. We end up at a leaf of T_i with bit b and output b. We perform this process *in parallel* for each $i \leq m$ to define the values p_i^{final}, instead of iterating over the rounds in order and updating the matching each time (this is crucial). Now, for these values p_i^{final}, the mixture consistency rules are obeyed precisely: p_i^{final} is the probability of outputting 1 in some experiment in which the first step is to take a uniform sample from $\mathcal{T}_i$, so its value is linear in the distribution of $\mathcal{T}_i$. So it would suffice to show that with nonzero probability over the randomness of the adversary, $|p_i - p_i^{\text{final}}| \leq \epsilon/2$ for every $i \leq m$. From this, we immediately get mixture consistency for each relevant triple $p_{i_1}, p_{i_2}, p_{i_3}$.

To prove that $|p_i - p_i^{\text{final}}| \leq \epsilon/2$, we will consider two more sets of intermediate p values and proceed by a kind of hybrid argument. The key claim underlying each step of the hybrid argument is a general lemma (Lemma 6.5) which says the following. Consider two decision trees T_1, T_2, a partial matching ρ, and two experiments. In the first experiment, we randomly traverse T_1 followed by T_2, using the initial matching ρ and randomly answering each query $f(x)$ (resp. $g(y)$) by choosing a random element in $[N]$ (resp. $[2N]$) which has so far been unmatched. After the traversal of T_1 we reach some leaf $\ell_1 \in T_1$, and then after the traversal of T_2 we reach some $\ell_2 \in T_2$. The second experiment is defined identically except we first traverse T_2 and then T_1. Lemma 6.5 states that if the heights of the trees T_1, T_2 and the size of ρ are all much smaller than N, then the distribution of the leaves (ℓ_1, ℓ_2) we reach in these two experiments will be close in statistical distance. Roughly speaking, this allows us to undo the sequence of matching extensions used to determine p_i^{final} and show that p_i is close to p_i^{final}.

To prove Lemma 6.5 (and more generally, to make all of the probabilistic arguments easier during the analysis of the adversary), we define a global random variable ω which simultaneously determines the path we will reach in any tree T by such a traversal starting with any initial matching ρ. This random variable is defined by choosing, for each $x \in [2N]$ a uniformly random ordering of the elements of $[N]$ (and symmetrically for each $y \in [N]$ an ordering of $[2N]$). Traversals of decision trees are then defined by answering queries according to the current partial assignment or else by matching x/y to the least unpaired element in its associated ordering. In this terminology, Lemma 6.5 will say that with high probability over the global randomness ω, the pair of leaves ℓ_1, ℓ_2 obtained in the T_1, T_2 and T_2, T_1 experiments are the same (this is actually a stronger statement than the former statistical distance bound, and is what we will directly prove). The argument that $T_1 \circ T_2$ behaves as $T_2 \circ T_1$ with high probability regarding matching extensions and reached leaves proceeds by induction on the total depth $h_1 + h_2$ of these trees, ultimately relying on an elementary probabilistic analysis of the effect of a single swap of two basic matching operations.

We remark that the probabilities p_i (and p_i^{final}) that our adversary provides are indeed different from honest probabilities. They are evaluated over ρ_{final} where the f- and g-queries *always get matched*, which ensures that the decision tree queries never find any unmatched element. For example, if the inspector submits the randomized decision tree $\{T_z\}_z$ such that for some rwPHP instance (f', g'), $T_z = 1$ if and only if $g'(f'(z)) = z$, then our adversary outputs 1 while the true probability is always at most $1/2$.

Finally, we briefly mention how the aforementioned ideas can be extended to a substantially large class of principles.

Generalization to finite combinatorial principles (Theorem 1.9). The above proof strategy extends to any finite combinatorial principle that can be captured using quantifier-free first-order formulas and admits an infinite counter-model $\mathcal{M}$ (a weaker form of Theorem 1.9). In a bit more detail, instead of lazy-sampling a partial matching between $[N]$ and $[2N]$, we lazy-sample a partial matching between $[N]$ and $\mathcal{M}$. For this,

we define a global random variable ω that chooses a uniformly random ordering of $[N]$ for each element in $\mathcal{M}$, and a uniformly random ordering of (a finite subset of) $\mathcal{M}$ for each element in $[N]$. The partial injection from $[N]$ to $\mathcal{M}$ together with a natural adaptation of the argument described above guarantees the existence of a winning strategy for the adversary. In turn, this implies that there is no proof of the finitisation of the principle in relativized APX_1, since one can show that a winning inspector can be extracted from a candidate proof using witnessing and other ideas. We refer to Section 6 for more details. The argument can be extended to quantified formulas (as in the actual statement of Theorem 1.9) using Herbrandization, a standard technique from mathematical logic.

Our approach shares with Riis' forcing proof [Rii94] the use of partial maps to transfer information from an infinite counter-model. In the forcing approach, this construction takes place over a nonstandard model of arithmetic, and a central task is to ensure that it yields a model of the required arithmetic axioms (see Atserias and Müller [AM15] and Müller [Mül21] for related results and techniques). Here, witnessing replaces the construction of a model with a finite adversary argument. The additional challenge is to maintain the local consistency of approximate counting answers as the partial structure grows. As explained above, our construction and its analysis supply this consistency without requiring the answers to approximate the true probabilities of a completed finite structure.

Acknowledgements. We would like to thank Ondra Ježil for discussions about the provability of weak pigeonhole principles in bounded arithmetic. We also thank Moritz Müller for conversations about Riis' result.

Funding Information. This work received support from the UKRI Frontier Research Guarantee Grant EP/Y007999/1 and the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick. The authors would also like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme Logical Foundations of Complexity Theory (LFC), where part of the work on this paper was undertaken. The LFC programme was supported by EPSRC grant EP/Z000580/1. Jiatu Li is supported by the MIT Schwarzman College of Computing PhD Fellowship. Oliver Korten is supported by the National Science Foundation under Grant No. DMS-2424441.

AI Methodology. Following the recommendation of the "AI and TCS: The Next Six Months" white paper, we describe the usage of AI tools in this research project in detail.

Activity	**AI used**
Asking the research question	No
Coming up with the approach	No
Proof development	Yes
Writing and exposition	Yes
Checking for bugs	Yes
Other supporting tasks	Yes

All AI-generated content was independently verified and edited by the authors, who take full responsibility for the correctness of all claims and proofs in this paper.

The main technical contribution of LLM tools is in the proof of Theorem 1.5. The authors formulated a framework (similar to Section 6.1 but only for rwPHP) that reduces unprovability of rwPHP to lower bounds for a query problem related to randomized decision trees, and worked out proofs of partial cases (e.g. non-adaptive queries with "dishonest" adversaries or $n/\operatorname{poly}(\log n)$ adaptive queries with "honest" adversaries that provide genuine approximate count estimates). The authors then formulated a proof strategy for the general case (fully describing the behavior of the adversary) and reduced its correctness to a key "robustification lemma" which they could not prove; they asked GPT 5.6 (from OpenAI) to prove this robustification lemma, and it succeeded. The machine-generated proof involved a formulation and proof of a key order-swapping lemma (see Lemma 6.5). After confirming this original proof, the authors suggested a significant simplification to the adversary's strategy to GPT 5.6, and the machine subsequently proved that this simpler strategy worked (and gave an analysis of the strategy which was much simpler than the analysis in the original proof). This latter proof is the one given in the current manuscript; it still relies on the key order-swapping lemma devised by GPT 5.6 for the original proof. Finally, the authors worked out a mostly complete proof sketch of an extension of this rwPHP result to Theorem 1.9 and presented it to GPT 5.6, which filled in some minor missing details in this argument.

LLM tools are also used for routine mathematics, including drafting proofs (with the ideas given as part of the prompt), adapting proof techniques to a similar setting, calculating parameters, and fixing errors. This affects Corollary 1.12 (GPT 6), Theorem 4.2 (by Claude Fable 5), Section 5.1 (by GPT 6), Theorem 5.5 (by GPT 5.6 and Claude Fable 5), Theorem 5.12 (by Claude Opus 5.5), Theorem 5.14 (by Claude Opus 5.5), Lemma 6.11 (by GPT 6), and results in the Appendix (by Claude Fable 5, Claude Fable 5.1, and Claude Opus 5.5). In these cases, LLM tools have not provided substantial conceptual or technical contributions.

Moreover, LLM tools (primarily GPT 6) are used to improve the presentation of the introduction through interaction with the authors.

2 Preliminaries

2.1 Basic Notation

For a string $x \in \{0,1\}^*$, we use $\|x\|$ to denote the bit length of x (this is to distinguish from the absolute value $|\cdot|$). We use $\|v\|_1$ to denote the ℓ_1-norm of a vector $v \in \mathbb{R}^n$, and in particular, $\|x\|_1$ to denote the Hamming weight of a string $x \in \{0,1\}^n$.

We write $\mathsf{T} \vdash \phi$ to denote that ϕ is provable in first-order logic from T. We write $\vdash_{\mathrm{FO}} \phi$ to emphasise provability of the sentence in first-order predicate logic, i.e., when the sentence is logically valid.

2.2 Bounded Arithmetic and Theory PV_1

We recall the background in bounded arithmetic needed below. For further details about bounded arithmetic, we refer to [CLOW26] and the survey [Oli25].

Theory PV_1. Our base theory PV_1 is a first-order theory formalizing deterministic polynomial-time reasoning. We use a standard presentation over finite binary strings, with natural numbers, rational numbers, circuits, and finite sequences represented by suitable efficient encodings. The language of PV_1 contains function symbols for polynomial-time algorithms, introduced through composition and limited recursion on notation, together with axioms describing these constructions. In the standard model, terms therefore compute polynomial-time functions, and quantifier-free formulas define polynomial-time predicates. Conversely, every polynomial-time predicate can be represented by a quantifier-free formula using its characteristic function.

The theory PV_1 supports induction on notation for quantifier-free formulas. It also proves ordinary induction for such formulas, i.e., for every quantifier-free formula $\varphi(i, \bar{a})$,

$$\mathsf{PV}_1 \vdash \forall \bar{a}\, \forall b\ \big(\varphi(0,\bar{a}) \wedge \forall i < b\, \big(\varphi(i,\bar{a}) \to \varphi(i+1,\bar{a})\big)\ \to\ \varphi(b,\bar{a})\big)\,.$$

A further property essential to our applications is that PV_1 admits a universal axiomatization, meaning an equivalent presentation whose axioms are universal closures of quantifier-free formulas. We refer to [Li25] for a detailed presentation of PV_1.

A quantifier is *bounded* if it has the form $\exists y \le t(\bar{x})$ or $\forall y \le t(\bar{x})$, where the term t does not contain y. It is *sharply bounded* if its bound has the form $\|t(\bar{x})\|$ (recall $\|\cdot\|$ denotes bit length). The statements expressing totality of search problems that we consider typically have the form

$$\forall x\, \exists y \le t(x)\ R(x,y),$$

where R is quantifier-free in the language of PV_1. Such statements belong to the class of sentences $\forall\Sigma^b_1(\mathsf{PV}_1)$, which assert that every input has a witness of polynomially bounded bit length whose validity can be checked in polynomial time.

We write $n \in \mathsf{Log}$ to indicate that $n = \|u\|$ for an auxiliary string u. Formally, $\forall n \in \mathsf{Log}\ \psi(n)$ abbreviates $\forall u\, \psi(\|u\|)$. This convention supplies a parameter of length n, allowing constructions of size polynomial in n, including 1^n and the binary representation of 2^n. Similarly, $n \in \mathsf{LogLog}$ indicates $n = \|\|u\|\|$. For precision parameters, the notation $\delta^{-1} \in \mathsf{Log}$ means that $\delta = 1/\|\Delta\|$ for a string Δ.

We also use relativized theories $\mathsf{PV}_1(F)$, where F is a tuple of fresh constant and function symbols, typically with specified polynomial bounds on output length. These symbols and their length axioms are added before closing the language under composition and limited recursion on notation, with induction available for quantifier-free formulas in the resulting language. Terms then describe polynomial-time computations with oracle access to F. These theories likewise admit universal axiomatizations. The particular oracle symbols and length bounds used here are specified in Section 3.

2.3 Witnessing Theorems

We will need the following formulation of Herbrand's Theorem.

Theorem 2.1 (Herbrand's Theorem; finite instance form (see, e.g., [Bus98, Section 2.5.1]))**.** *Let T be a universal theory in a first-order language $\mathcal{K}$ with equality and at least one constant symbol. Let $\bar{c}$ be a (possibly empty) tuple of fresh parameter constants, and let $\chi(\bar{c},\bar{x})$ be a quantifier-free formula in $\mathcal{K}(\bar{c})$ whose free variables are among $\bar{x}$. If*

$$T \vdash \exists \bar{x}\, \chi(\bar{c},\bar{x}),$$

then there are finitely many ground instances $\theta_j(\bar{s}_j(\bar{c}))$, $1 \le j \le m$, of axioms $\forall \bar{u}_j\, \theta_j(\bar{u}_j)$ of T, and finitely many tuples of ground terms $\bar{t}_i(\bar{c})$, $1 \le i \le r$, of the same length as $\bar{x}$, such that

$$\vdash_{\mathrm{FO}}\ \bigwedge_{j=1}^{m} \theta_j(\bar{s}_j(\bar{c}))\ \longrightarrow\ \bigvee_{i=1}^{r} \chi(\bar{c},\bar{t}_i(\bar{c})).$$

Here $\vdash_{\mathrm{FO}}$ denotes derivability in first-order logic with equality, every formula θ_j is quantifier-free, and all terms belong to $\mathcal{K}(\bar{c})$.

We will also need to extract information from a proof of a sentence of higher quantifier complexity. The following result will be useful.

Theorem 2.2 (KPT Witnessing Theorem [KPT91])**.** *Let T be a universal first-order theory in a language $\mathcal{L}$, and let $\varphi(x, y, z)$ be a quantifier-free $\mathcal{L}$-formula. Suppose that*

$$T \vdash \forall x\, \exists y\, \forall z\ \varphi(x, y, z).$$

Then there exist a constant $r \geq 1$ and $\mathcal{L}$-terms $t_1(x), t_2(x, z_1), \ldots, t_r(x, z_1, \ldots, z_{r-1})$ such that

$$T \vdash \forall x\, \forall z_1 \cdots \forall z_r \bigvee_{i=1}^{r} \varphi\big(x, t_i(x, z_1, \ldots, z_{i-1}), z_i\big).$$

(*The variables x, y, z may also denote tuples, in which case each t_i is a corresponding tuple of terms.*)

2.4 Unrelativized Combinatorial Principles

2.4.1 Refuter(Yao)

We review the definition of the TFZPP problem called Refuter(Yao) [CLOW26]. Recall that Yao's distinguisher-to-predictor transformation [Yao82] shows that if a distribution $\mathcal{D}$ over $\{0,1\}^n$ is not ε-pseudorandom, i.e., there is a circuit $C : \{0,1\}^n \to \{0,1\}$ (called *distinguisher*) such that

$$\Big|\Pr[C(\mathcal{D})] - \Pr[C(\mathcal{U}_n)]\Big| > \varepsilon,$$

then there exists $i \in [n]$ and a *predictor* $P : \{0,1\}^{i-1} \to \{0,1\}$ such that

$$\Pr_{x \leftarrow \mathcal{D}}[P(x_{<i}) = x_i] \geq \frac{1}{2} + \frac{\varepsilon}{4n},$$

i.e., P predicts the i-th bit of $\mathcal{D}$ with advantage at least $\varepsilon/4n$.

We say that a discrete probability distribution $\mathcal{D}$ is *flat* if it is the uniform distribution over a multiset. The *size* of the distribution is the size of this multiset. We will represent flat distributions explicitly as a list of strings. In the subsequent discussions, we might tacitly assume that the relevant distribution is flat and explicitly represented.

Definition 2.3. The search problem Refuter(Yao) is defined as follows.

- (*Parameters*). Length of strings n, distribution size m, predictor size s, and advantage $\delta \in [0,1]$.
- (*Input*). A circuit $G : \{0,1\}^{nm} \to [n] \times \{0,1\}^s$ (called *predictor generator*).
- (*Solution*). Any explicit flat distribution $\mathcal{D} \in (\{0,1\}^n)^m$ of size m such that the following holds:

  Let $(i, P) \triangleq G(\mathcal{D})$, where $P : \{0,1\}^{i-1} \to \{0,1\}$ is parsed as a circuit of description length $\leq s$. Then

  $$\Pr_{x \leftarrow \mathcal{D}}[P(x_{<i}) = x_i] < \frac{1}{2} + \delta.$$

  In other words, P is not a predictor of the i-th bit of $\mathcal{D}$ with advantage δ.

At a high level, Refuter(Yao) asks to generate a distribution $\mathcal{D}$ that is unpredictable against a *given* predictor generator G – a deterministic algorithm that aims to output a predictor P for $\mathcal{D}$. The distribution $\mathcal{D}$ is not necessarily an unpredictable (or equivalently, pseudorandom) distribution against *small circuits*; it suffices to fool the given *deterministic* predictor generator G.

To ensure that Refuter(Yao) is a total search problem, we assume $(\delta^2/10) \cdot m \geq s + \lceil \log n \rceil + 1$ if it is not explicitly specified (see [CLOW26] for more discussions).

2.4.2 Pigeonhole Principles and Total Search Problems

Definition 2.4 (Lossy Code)**.** We define LossyCode as the following total search problem: Given circuits $C : \{0,1\}^n \to \{0,1\}^m$ and $D : \{0,1\}^m \to \{0,1\}^n$, where $m < n$, the solution is a string $x \in \{0,1\}^n$ such that $D(C(x)) \neq x$.

Definition 2.5 (rwPHP)**.** The retraction weak Pigeonhole Principle (rwPHP) is defined as the $\forall\Sigma_1$-sentence in the language of PV_1 stating that "LossyCode is total". Specifically: for every pair of circuits $C : \{0,1\}^n \to \{0,1\}^m$ and $D : \{0,1\}^m \to \{0,1\}^n$ satisfying $n, m \in \mathsf{Log}$, $m < n$, there exists $x \in \{0,1\}^n$ such that $D(C(x)) \neq x$.

3 APX_1, APX_1^*, and Related Theories

3.1 APX_1 and APX_1^*

We first recall the theory APX_1 introduced in [CLOW26]. The language of APX_1 is the relativized theory $\mathsf{PV}_1(\mathsf{P})$, where $\|\mathsf{P}(C,\Delta)\| \leq \|C\| + 2\|\Delta\|$ is intended to output a rational number $p \in \Pr_{x\in\mathcal{U}}[C(x) = 1] \pm 1/\|\Delta\|$. For simplicity, we use $\mathsf{P}_\delta(C)$ to denote $\mathsf{P}(C, 1^{\delta^{-1}})$ for $\delta^{-1} \in \mathsf{Log}$.

Besides the ordinary axioms for relativized PV_1 (where functions are interpreted as polynomial-time oracle algorithms), four axioms are introduced to govern the behavior of the oracle P:

P-ORACLE AXIOMS FOR APX_1

- (*Basic Axiom*). $\mathsf{P}_\delta(C) \in [0,1] \cap \mathbb{Q}$.
- (*Boundary Axiom*). If C is a circuit that does not read its input bits, then $\mathsf{P}_\delta(C) = \mathrm{Val}(C) \in \{0,1\}$, where $\mathrm{Val}(C)$ denotes the output of C.
- (*Precision Consistency Axiom*). For every circuit C and every $\delta_1^{-1}, \delta_2^{-1}, \beta^{-1} \in \mathsf{Log}$, $|\mathsf{P}_{\delta_1}(C) - \mathsf{P}_{\delta_2}(C)| \leq \delta_1 + \delta_2 + \beta$.
- (*Local Consistency Axiom*). For every circuit C and $\delta^{-1}, \beta^{-1} \in \mathsf{Log}$, let $\mathrm{Fix}_b(C)$ be the circuit obtained by fixing the rightmost input bit of C to b, then

$$\left|\mathsf{P}_\delta(C) - \frac{\mathsf{P}_\delta(\mathrm{Fix}_0(C)) + \mathsf{P}_\delta(\mathrm{Fix}_1(C))}{2}\right| \leq 2\delta + \beta.$$

Intuitively, the theory captures minimal mathematical reasoning capability corresponding to probabilistic polynomial time. It is shown that APX_1 provides a rich toolkit for probability theory, where probability is formalized using the oracle P.

The Theory APX_1^*. We define the theory APX_1^* corresponding to minimal theory for multiplicative-error approximate counting (i.e. the complexity class prSBP) by modifying the last two axioms to satisfy the behavior of a multiplicative-error approximate counting oracle: $\mathsf{P}_\delta(C) \in (1 \pm \delta) \cdot \Pr_{x\leftarrow\mathcal{U}}[C(x) = 1]$.

For simplicity, we introduce the notation $u \lesssim^* v$ for numbers u, v as the abbreviation for the formula "$\forall\beta^{-1} \in \mathsf{Log}\ u \leq (1+\beta)\cdot v$", and $u \approx^* v$ as the abbreviation of the formula "$u \lesssim^* v \wedge v \lesssim^* u$"; this hides the slack parameter β. We can then describe the axioms of APX_1^*:

P-ORACLE AXIOMS FOR APX_1^*

- (*Basic and Boundary Axioms*). Same as the corresponding axioms in APX_1.
- (*Precision Consistency Axiom*). For every circuit C and every $\delta_1^{-1}, \delta_2^{-1} \in \mathsf{Log}$,

$$(1-\delta_1)\mathsf{P}_{\delta_2}(C) \lesssim^* (1+\delta_2)\mathsf{P}_{\delta_1}(C)$$

- (*Local Consistency Axiom*). For every circuit C and every $\delta^{-1} \in \mathsf{Log}$, then

$$(1-\delta)p_M \lesssim^* (1+\delta)\mathsf{P}_\delta(C) \quad \text{and} \quad (1-\delta)\mathsf{P}_\delta(C) \lesssim^* (1+\delta)p_M$$

where $p_M \triangleq \frac{1}{2}(\mathsf{P}_\delta(\mathrm{Fix}_0(C)) + \mathsf{P}_\delta(\mathrm{Fix}_1(C)))$.

Note that an additive-error approximate counting oracle $\mathsf{P}_\delta^+(\cdot)$ can be implemented by a multiplicative-error approximate counting oracle $\mathsf{P}_\delta^*(\cdot)$, as $\mathsf{P}_\delta^*(C) \in (1 \pm \delta) \cdot p$ implies that $\mathsf{P}_\delta^*(C) \in p \pm \delta$. Accordingly, we can show that APX_1^* is indeed an extension of APX_1.

Proposition 3.1. *For every sentence ϕ in the language of $\mathsf{PV}_1(\mathsf{P})$, $\mathsf{APX}_1 \vdash \phi$ implies that $\mathsf{APX}_1^* \vdash \phi$.*

Proof. It suffices to show that axioms of APX_1 can be proved in APX_1^*. As the two theories share basic and boundary axioms, it suffices to verify the other two axioms. Fix the slack parameter $\beta^{-1} \in \mathsf{Log}$ of the target APX_1 axiom and let $\beta' \triangleq \beta/3$ (so that $\beta'^{-1} \in \mathsf{Log}$); below, the notation $\lesssim^*$ in the axioms of APX_1^* is instantiated at the slack β'.

- (PRECISION CONSISTENCY AXIOM FOR APX_1). We may assume that $\delta_1 + \delta_2 + \beta < 1$, as otherwise the axiom is immediate from the BASIC AXIOM FOR APX_1^*; in particular $\delta_1, \delta_2 < 1$. The PRECISION CONSISTENCY AXIOM FOR APX_1^* (with the roles of δ_1, δ_2 exchanged) gives $(1-\delta_2)\mathsf{P}_{\delta_1}(C) \le (1+\beta')(1+\delta_1)\mathsf{P}_{\delta_2}(C)$, and therefore

  $$\mathsf{P}_{\delta_1}(C) - \mathsf{P}_{\delta_2}(C) \le \big((1+\beta')(1+\delta_1) - 1\big) \cdot \mathsf{P}_{\delta_2}(C) + \delta_2 \cdot \mathsf{P}_{\delta_1}(C) \le \delta_1 + \delta_2 + 2\beta' \le \delta_1 + \delta_2 + \beta,$$

  where the second inequality uses $\mathsf{P}_{\delta_1}(C), \mathsf{P}_{\delta_2}(C) \in [0,1]$ (the BASIC AXIOM FOR APX_1^*) and $\delta_1 < 1$; the bound on $\mathsf{P}_{\delta_2}(C) - \mathsf{P}_{\delta_1}(C)$ is symmetric.
- (LOCAL CONSISTENCY AXIOM FOR APX_1). Similarly, we may assume $2\delta + \beta < 1$, so $\delta < 1/2$; also $p_M \in [0,1]$ by the BASIC AXIOM FOR APX_1^*. From $(1-\delta)\mathsf{P}_\delta(C) \le (1+\beta')(1+\delta)p_M$ we obtain

  $$\mathsf{P}_\delta(C) - p_M \le \big((1+\beta')(1+\delta) - 1\big) \cdot p_M + \delta \cdot \mathsf{P}_\delta(C) \le 2\delta + 2\beta' \le 2\delta + \beta,$$

  and the bound on $p_M - \mathsf{P}_\delta(C)$ follows from $(1-\delta)p_M \le (1+\beta')(1+\delta)\mathsf{P}_\delta(C)$ analogously.

This completes the proof. □

Standard Models and Admissible Models. Similar to [CLOW26], we now justify that the axioms for APX_1^* capture exactly the intended behavior of P by showing that standard models (i.e. P-oracle behaves as a multiplicative-error counting oracle) are exactly *admissible models* (i.e. P-oracle satisfying axioms).

Let $\mathbb{M}$ denote the standard model of PV_1: the universe is the set of all binary strings, and every function symbol of PV_1 is interpreted by the polynomial-time algorithm defining it. For a function $\hat{\mathsf{P}} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, let $\mathbb{M}(\hat{\mathsf{P}})$ denote the model of $\mathsf{PV}_1(\mathsf{P})$ obtained from $\mathbb{M}$ in which P is interpreted by $\hat{\mathsf{P}}$.

Definition 3.2 (Standard Models)**.** A function $\hat{\mathsf{P}} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a *correct multiplicative-error approximate counting function* if the following conditions hold:

1. for every circuit $C : \{0,1\}^n \to \{0,1\}$ and every Δ, $\hat{\mathsf{P}}(C,\Delta)$ is (the encoding of) a rational number $q \in [0,1]$ of length at most $\|C\| + 2\|\Delta\|$ such that $q \in (1 \pm 1/\|\Delta\|) \cdot p(C)$, where $p(C) \triangleq \Pr_{x\in\{0,1\}^n}[C(x) = 1]$; and
2. $\hat{\mathsf{P}}(C,\Delta) = \mathrm{Val}(C) \in \{0,1\}$ whenever C does not read its input bits.

We say that $\mathbb{M}(\hat{\mathsf{P}})$ is a *standard model* of APX_1^* if $\hat{\mathsf{P}}$ is a correct multiplicative-error approximate counting function.

Definition 3.3 (Admissible Models)**.** Let $\hat{\mathsf{P}} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a function. We say that $\mathbb{M}(\hat{\mathsf{P}})$ is an *admissible model* of APX_1^* if it satisfies all axioms of APX_1^*.

The following proposition establishes the equivalence between standard and admissible models. Its proof is essentially an adaption of [CLOW26, Theorem 2.5], and is deferred to Section A.1.

Proposition 3.4. *Let $\hat{\mathsf{P}} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be any function. Then $\mathbb{M}(\hat{\mathsf{P}})$ is a standard model of APX_1^* if and only if it is an admissible model of APX_1^*.*

3.2 Relativized Theories and Relativized Principles

To study the (un)provability of combinatorial principles such as wPHP, dwPHP and rwPHP in bounded theories, we will also work with relativized theories and principles, where the compression and decompression circuits are replaced by fresh function symbols, see, e.g., [Jeř07b].

First, we discuss relativization in the context of pigeonhole principles.

Relativized APX_1 and APX_1^* for Pigeonhole Principles. Let f, g be two new function symbols with the intended length bounds $\|f(u)\| = \|u\| - 1$ and $\|g(v)\| = \|v\| + 1$. We define $\mathsf{PV}_1(f,g)$ as the relativized theory of PV_1, where f and g are added as base functions and the length bounds are added as axioms. Function symbols in $\mathsf{PV}_1(f,g)$ are therefore polynomial-time oracle algorithms with oracle calls to f and g.

The relativized version of APX_1 further extends $\mathsf{PV}_1(f,g)$ by containing an approximate-counting symbol $\mathsf{P}(C,\Delta)$. The input C is parsed as an (f,g)-oracle Boolean circuit, and Δ is the precision parameter. The intended semantics of $\mathsf{P}(C,\Delta)$ is that

$$\mathsf{P}(C,\Delta) \in \Pr_{z\in\{0,1\}^\ell}\left[C^{f,g}(z) = 1\right] \pm \frac{1}{\|\Delta\|}. \tag{3.1}$$

where ℓ is the input length of C. Similar to the unrelativized theory APX_1, we do not allow the circuit C to contain P-oracles.

The axioms of $\mathsf{APX}_1(f,g)$ consist of the axioms of $\mathsf{PV}_1(f,g)$ and the counting axioms for APX_1 (see Section 3.1). The basic, precision and local consistency axioms are identical (with the change that the circuit can query f and g). The boundary axiom will be:

- (*Boundary Axiom*). If C does not read its input bits and $\delta^{-1} \in \mathsf{Log}$, then $\mathsf{P}_\delta(C) = \mathrm{Val}^{f,g}(C) \in \{0,1\}$, where $\mathrm{Val}^{f,g}(C)$ denotes the output of C, which may involve oracle queries to f and g.

Similarly, we can define the relativized theory $\mathsf{APX}_1^*(f,g)$, where the local and precision consistency axioms are replaced by that of APX_1^* (relativized). Note that we may also define the relativized theory with only f or g accordingly.

Relativized Pigeonhole Principles. The purpose of introducing f, g is to phrase combinatorial principles in a black-box fashion. The relativized retraction PHP, denoted by $\mathsf{rwPHP}(f, g)$, is defined as[4]

$$\mathsf{rwPHP}(f,g) \triangleq \forall n \in \mathsf{Log}\ \exists x \in \{0,1\}^n \big(\|f(x)\| \geq n \vee g(f(x)) \neq x\big). \tag{3.2}$$

These sentences can be defined in the language of $\mathsf{PV}_1(f, g)$ (and thus also $\mathsf{APX}_1(f, g)$). Similarly, we define the relativized versions of dwPHP and wPHP as

$$\mathsf{dwPHP}(g) \triangleq \forall n \in \mathsf{Log}\ \exists y \in \{0,1\}^{n+1}\ \forall x \in \{0,1\}^n\ g(x) \neq y \tag{3.3}$$

$$\mathsf{wPHP}(f) \triangleq \forall n \in \mathsf{Log}\ \exists x_1, x_2 \in \{0,1\}^n \big(x_1 \neq x_2 \wedge (f(x_1) = f(x_2) \vee \|f(x_1)\| \geq n \vee \|f(x_2)\| \geq n)\big). \tag{3.4}$$

These sentences are in the language of $\mathsf{PV}_1(g)$ and $\mathsf{PV}_1(f)$, respectively.

We also consider a relativized extension of APX_1 for Theorem 1.9, where the additional function symbols represent maps on a finite universe $[N]$. Unlike the functions used in the pigeonhole principles above, these maps need not satisfy an output length bound depending only on the lengths of their arguments, uniformly in N. For example, $f(0)$ may equal $N - 1$. To incorporate such functions into the underlying theory of polynomial-time computation, we introduce N as an explicit argument of each new symbol and impose the universal length bound

$$\|f(N, \bar{x})\| \leq \|N\|.$$

These bounds are included before closing the language of the theory under composition and limited recursion on notation. We give the precise definition next.

The Relativized Theory $\mathsf{APX}_1(\widehat{\mathcal{L}})$. Let $\mathcal{L}$ be a finite language consisting of function and constant symbols. For each d-ary function symbol $f \in \mathcal{L}$, introduce a function symbol $\widehat{f}$ of arity $d + 1$. An original constant symbol c is treated as the case $d = 0$, giving a unary symbol $\widehat{c}$. Write $\widehat{\mathcal{L}}$ for this parameterized language.

The primitive symbols satisfy the universal length axioms

$$\|\widehat{f}(s, x_1, \ldots, x_d)\| \leq \|s\|, \qquad f \in \mathcal{L},$$

including $\|\widehat{c}(s)\| \leq \|s\|$ for constants. The size argument s is an ordinary input and may vary between oracle calls. These are the only axioms imposed on the primitive symbols of $\widehat{\mathcal{L}}$.

We first define the base theory $B = \mathsf{PV}_1(\widehat{\mathcal{L}}, \mathsf{P})$. Start with the ordinary PV_1 functions and axioms, the primitive symbols $\widehat{\mathcal{L}} \cup \{\mathsf{P}\}$, their length axioms above, and

$$\|\mathsf{P}(C, \Delta)\| \leq \|C\| + 2\|\Delta\|.$$

As usual, generate further functions in stages by composition and limited recursion on notation. At each stage, include the defining axioms for the functions admitted so far and quantifier-free induction on notation, with parameters, in the current language.

The theory $\mathsf{APX}_1(\widehat{\mathcal{L}})$ extends B by the four approximate counting axioms, defined in the natural way. As before, $P(C, \Delta)$ can be applied to Boolean circuits with $\widehat{\mathcal{L}}$-oracle gates and no P gates.

For applications to finite $\mathcal{L}$-structures on $[N]$, the original symbols are represented by the slice $\widehat{f}(N, \bar{x})$ and $\widehat{c}(N)$. Closure of this slice under the functions, and membership of its constants in $[N]$, are hypotheses of the finite principles rather than axioms of $\mathsf{APX}_1(\widehat{\mathcal{L}})$. Note that every $\mathcal{L}$-structure on $[N]$ can be represented in this way: use its tables at the slice $s = N$ on arguments from $[N]$, and return zero elsewhere. The length axioms hold because every value is below N.

[4]Here we use $\{0,1\}^n$ instead of N for convenience. This difference is inessential in our results.

3.3 $\mathsf{True}_\forall\mathsf{APX}_1$: True Universal Axioms of (Relativized) APX_1

Definition 3.5 (Standard models)**.** A *standard model* of the language of $\mathsf{APX}_1(g)$ is a structure over the standard integers in which g is interpreted by an arbitrary family of functions $\{g_m : \{0,1\}^{m-1} \to \{0,1\}^m\}$, the symbol P is interpreted by a correct approximate-counting oracle for g (defined below), and every $\mathsf{PV}_1(g,\mathsf{P})$-function symbol is interpreted by its defining polynomial-time machine with oracle access to g and P. Here a function P is a *correct approximate-counting oracle* for g if for every g-oracle circuit $C : \{0,1\}^m \to \{0,1\}$ and every Δ:

1. $\mathsf{P}(C,\Delta) \in \mathbb{Q} \cap [0,1]$;
2. $\left|\mathsf{P}(C,\Delta) - \Pr_{z \leftarrow \{0,1\}^m}[C^g(z) = 1]\right| \leq 1/\|\Delta\|$;
3. $\mathsf{P}(C,\Delta) = \mathrm{Val}^g(C)$ whenever C does not read its random input.

Definition 3.6 (True universal axioms and $\mathsf{True}_\forall\mathsf{APX}_1$)**.** A *true universal axiom* of $\mathsf{APX}_1(g)$ is a universal sentence $\Psi = \forall \vec{u}\ \psi(\vec{u})$, with ψ quantifier-free in the language of $\mathsf{APX}_1(g)$, that holds in every standard model. The *true universal theory* of $\mathsf{APX}_1(g)$ is the set of all true universal axioms. We use $\mathsf{True}_\forall\mathsf{APX}_1(g)$ to denote the theory consisting of $\mathsf{PV}_1(P,g)$ extended with the set of all true universal axioms. We extend this definition to $\mathsf{True}_\forall\mathsf{APX}_1(f,g)$ in the natural way.

Specifically, the four $\mathsf{APX}_1(g)$ axioms are true universal axioms of $\mathsf{APX}_1(g)$. Moreover, the true universal theory includes combinatorial axioms that are unlikely to be provable in $\mathsf{APX}_1(g)$, as well as other mathematical statements; we provide two examples.

Example 3.7 ($\#\mathsf{rwPHP}$)**.** The approximate counting version of retraction weak Pigeonhole Principle, denoted $\#\mathsf{rwPHP}$, is the following statement:

- For every g-oracle circuits $C : \{0,1\}^n \to \{0,1\}^{n-1}$, $D : \{0,1\}^{n-1} \to \{0,1\}^n$, and every $\delta^{-1}, \beta^{-1} \in \mathsf{Log}$, let $T_{C,D} : \{0,1\}^n \to \{0,1\}$ be the circuit $T(x) \triangleq [D(C(x)) = x]$, $\mathsf{P}_\delta(T_{C,D}) \leq 1/2 + \delta + \beta$.

This is a true sentence for any correct approximate-counting oracle. By the Existence Lemma in $\mathsf{APX}_1(g)$ (see [CLOW26]), if $\mathsf{APX}_1(g) \vdash \#\mathsf{rwPHP}$, then $\mathsf{APX}_1(g)$ proves the ordinary retraction weak Pigeonhole Principle.

We record the following result (Theorem 1.8 Item 1), which follows from the discussion above using that $\#\mathsf{rwPHP}$ is a universal sentence that holds in every standard model.

Proposition 3.8. $\mathsf{True}_\forall\mathsf{APX}_1(f,g) \vdash \mathsf{rwPHP}(f,g)$.

Example 3.9 ($\mathsf{prBPP} = \mathsf{prP}$)**.** Assume that $\mathsf{prBPP} = \mathsf{prP}$ and let $\mathsf{capp}(C,\Delta)$ be the polynomial-time algorithm that outputs a number in $\Pr_x[C(x) = 1] \pm 1/\|\Delta\|$ for circuits C without oracle gates. The correctness of the algorithm can be formalized as the following universal statement:

- For every circuit $C : \{0,1\}^n \to \{0,1\}$ (without oracle gates), any $\delta, \beta \in \mathsf{Log}^{-1}$, it follows that $|\mathsf{P}_\delta(C) - \mathsf{capp}(C, 1^{\delta^{-1}})| \leq 2\delta + \beta$.

This is a true sentence (and thus a true universal axiom) if $\mathsf{prBPP} = \mathsf{prP}$ and capp is a correct algorithm for approximate counting. Note that this sentence does not involve the g-oracle.

4 LossyCode, Refuter(Yao), and Total Search Problems of APX_1

In this section, we prove Theorem 1.2 and Theorem 1.3.

4.1 Equivalence Between LossyCode (rwPHP) and Refuter(Yao)

Recall that [CLOW26] proved a witnessing theorem for the $\forall\Sigma_1$-fragment of the theory APX_1 via the search problem Refuter(Yao), namely, TFNP-problems that are provably total in APX_1 can be reduced to Refuter(Yao). Yet it is unclear whether Refuter(Yao) is provably total in APX_1.

Theorem 4.1 (Witnessing for APX_1 [CLOW26])**.** *Let $\varphi(x, y)$ be a quantifier-free formula in the language of PV_1. If $\mathsf{APX}_1 \vdash \forall x\, \exists y\, \varphi(x, y)$, there exists a deterministic polynomial-time Turing reduction from the search problem defined by φ to* Refuter(Yao).

We show that the answer is likely *no* by proving PV_1*-equivalence* between the totality of Refuter(Yao) and rwPHP, which is unlikely provable in APX_1 (as we will show later). In particular, we show that Refuter(Yao) and the search problem LossyCode are equivalent with respect to polynomial-time reduction.

Theorem 4.2. *There are polynomial-time mapping reductions between* Refuter(Yao) *and* LossyCode*. Moreover, the totality of* Refuter(Yao) *and* rwPHP *(i.e. totality of* LossyCode*), as $\forall\Sigma^b_1$-formulas in the language of PV_1, are equivalent with respect to PV_1.*

Proof. We note that the reduction from Refuter(Yao) to LossyCode is proved in [CLOW26, Theorem 5.4] and the idea originates from [Kor22]; the formalization in PV_1 is left as an exercise. In the rest of the proof, we construct the reduction from LossyCode to Refuter(Yao).

Let $C : \{0,1\}^n \to \{0,1\}^{n-1}$ be the compressor and $D : \{0,1\}^{n-1} \to \{0,1\}^n$ be the decompressor of a LossyCode instance; we assume without loss of generality that the stretch is 1. Let $\varepsilon \in (0, 1)$ be a parameter to be determined later. The reduction consists of the following two steps.

1. Construct a new instance where the decompressor is the truth table generator $\mathsf{TT}_{s,\ell} : \{0,1\}^s \to \{0,1\}^\ell$ that takes a circuit of size s and outputs the first ℓ-bits of its truth-table, where $s = \ell^\varepsilon$. More formally, we construct a polynomial-time mapping reduction from (C, D) to a new instance $(E, \mathsf{TT}_{s,\ell})$ of LossyCode.
2. We further reduce the instance $(E, \mathsf{TT}_{s,\ell})$ to Refuter(Yao) with desired parameters.

We note that for any $\varepsilon \in (0, 1)$, the first step can be achieved by the standard "tree-construction" [Kor21] (see, e.g., [Kor22, Kor25, LPT24]), and its formalization in PV_1 is straightforward (see, e.g., [Jeř07b, Theorem 3.1 (iii)]). We then focus on the second step.

Now let $(E, \mathsf{TT}_{s,\ell})$ be the instance of LossyCode with $\varepsilon \triangleq 1/20$ and $s = \ell^\varepsilon$. We assume without loss of generality that m is sufficiently large. Let $\ell = 2^\nu$, we construct a Refuter(Yao) instance with parameters

$$n' \triangleq \nu, \qquad m \triangleq 2^{\nu/10} = \ell^{1/10}, \qquad s' \triangleq s + \nu^3, \qquad \delta \triangleq 1/4, \tag{4.1}$$

which satisfy $(\delta^2/10) \cdot m \geq s' + \lceil \log \ell \rceil + 1$ since $s \leq \ell^{1/20}$.

The predictor generator $G : \{0,1\}^{\nu m} \to [\nu] \times \{0,1\}^{s'}$ parses its input as a list $\mathcal{D} = (x^{(1)}, \ldots, x^{(m)})$ of ν-bit strings. For $0 \leq j < \nu$, let $f_j : \{0,1\}^j \to \{0,1\}$ be the empirically optimal predictor of the $(j+1)$-st bit: $f_j(z)$ is the most common value of $(x^{(t)})_{j+1}$ among the samples with $(x^{(t)})_{1\ldots j} = z$, counted with multiplicity (0 to break tie). We say that f_j *errs* on $x^{(t)}$ if $f_j\big((x^{(t)})_{1\ldots j}\big) \neq (x^{(t)})_{j+1}$.

Claim. *For every $t \in [m]$, the number of $j \in \{0, \ldots, \nu - 1\}$ such that f_j errs on $x^{(t)}$ is at most* $\log m$.

Proof. Let $c_{t,j}$ be the number of indices $t' \in [m]$ such that $x^{(t')}_{\leq j} = x^{(t)}_{\leq j}$, so that $c_{t,0} = m$. The sequence $c_{t,j}$ is non-increasing in j, and $c_{t,\nu} \geq 1$. Suppose that f_j errs on $x^{(t)}$ and let $b \triangleq (x^{(t)})_{j+1}$. Among the $c_{t,j}$ samples sharing the prefix $(x^{(t)})_{\leq j}$, the number of those with $(j+1)$-st bit equal to b is at most $c_{t,j}/2$: otherwise b would be the strict majority value and f_j would output b instead. Hence $c_{t,j+1} \leq c_{t,j}/2$, and there are at most $\log m$ erring indices. □

Summing the claim above over $t \in [m]$ and using $\log m = \nu/10$,

$$\sum_{j=0}^{\nu-1} \left|\{t \in [m] : f_j \text{ errs on } x^{(t)}\}\right| \;\le\; m \log m \;=\; \frac{m\nu}{10}, \tag{4.2}$$

so there exists $j < \nu$ such that f_j errs on at most $m/10$ samples. The predictor generator G then:

- computes the smallest j such that f_j errs on at most $m/10$ samples;
- computes the truth table F_j of the function $\tilde{f}_j : \{0,1\}^\nu \to \{0,1\}$ defined as $\tilde{f}_j(w) \triangleq f_j(w_{1\dots j})$;
- computes $E(F_j) \in \{0,1\}^s$ and parses the output as a circuit $K : \{0,1\}^\nu \to \{0,1\}$; and
- outputs $(j+1, P)$, where $P(z) \triangleq K(z \circ 0^{\nu-j})$.

It can be verified that the instance map outputs G with the parameters in Equation (4.1). To see the correctness of the reduction, let $\mathcal{D}$ be any solution of G, and suppose toward a contradiction that $\mathsf{TT}_{s,\ell}(E(F_j)) = F_j$, i.e., $\mathrm{tt}(K) = F_j$. Then for every $z \in \{0,1\}^j$ we have $P(z) = f_j(z)$, so P errs on at most $m/10$ samples, and subsequently

$$\Pr_{x \leftarrow \mathcal{D}}\big[P(x_{<i}) = x_i\big] \;\ge\; \frac{9}{10} \;>\; \frac{1}{2} + \delta,$$

contradicting that $\mathcal{D}$ is a solution.

Formalization in PV_1. As mentioned above, it suffices to verify that the step 2 (the reduction from the LossyCode instance $(E, \mathsf{TT}_{s,\ell})$ to Refuter(Yao)) can be formalized in PV_1. Recall that rwPHP formalizes exactly LossyCode is total, and we formalize it with $n \in \mathsf{Log}$. It is worth noting that the construction of the truth table F_j is feasible as $\nu \in \mathsf{LogLog}$ (recall that $\nu = \log \ell$ and $\ell = \mathrm{poly}(n, |C|, |D|)$). The rest of the proof can be formalized straightforwardly, with some routine work that has been comprehensively described in literature (see, e.g., [Pic15, Kra19, Li25]). □

4.2 Witnessing Theorem for the $\forall\Sigma^b_1(\mathsf{PV}_1)$-Consequences of APX_1

Both APX_1 and APX^*_1 are universal theories in the language of relativized PV_1. Following a recent work of [Tha24], we derive witnessing theorems for the $\forall\Sigma^b_1$-fragments of these theories.

Definition 4.3 (Witness of Axiom Violation)**.** Let $L = \{(C_i, \Delta_i, p_i)\}_{i\in[k]}$ be a list of query-answer pairs to the P oracle, where each C_i is a circuit, Δ_i is a string, and $p_i \in [0,1]$ is a rational number with at most $\|C_i\| + 2\|\Delta_i\|$ bits. A *witness of axiom violation* for local consistency axiom of APX_1 is a tuple (u, v, w, B), $u, v, w \in [k]$, such that the followings hold.

- C_v, C_w are exactly C_u with last bit fixed to be 0 and 1, respectively.
- $|\Delta_u| = |\Delta_v| = |\Delta_w|$. Let $\delta \triangleq 1/|\Delta_u|$, and $\beta \triangleq 1/|B|$.
- $|p_u - (p_v + p_w)/2| > 2\delta + \beta$.

Accordingly, we can define witness of violation for other axioms of APX_1 and APX^*_1. Note that given the list L and τ, it can be verified in polynomial time whether τ is indeed a witness of axiom violation.

Definition 4.4 (Inspector-Adversary Game)**.** The inspector-adversary game for the theory APX_1 (resp. the theory APX^*_1) is defined as follows.

- In each round, the inspector outputs a circuit C_i and a string Δ_i (called inspector's *challenge*), and the adversary outputs a rational $p_i \in [0,1]$ with at most $\|C_i\| + 2\|\Delta_i\|$ bits (called adversary's *response*).
- Let $\{\mathsf{P}(C_i, \Delta_i) = p_i\}_{i\in[k]}$ be a transcript of k rounds. The inspector wins after k rounds, if it can output a valid witness of axiom violation for any axiom of APX_1 (resp. APX^*_1).

Definition 4.5 (Local)**.** The search problem $\mathsf{Local}(\mathsf{APX}_1)$, associated with the Inspector-Adversary Game for APX_1, is defined as follows: Given a query algorithm $Q = (M_1, M_2, \ldots, M_k, M_{k+1})$, where M_i is a circuit that takes the previous transcript as input and outputs the message in the i-th round, and in particular, M_{k+1} is intended to output a witness of axiom violation for any APX_1 axiom. The solution is a list of adversary's responses $p_1, \ldots, p_k$ given which the inspector does not win after k rounds.

Similarly, we can define the search problem $\mathsf{Local}(\mathsf{APX}_1^*)$ according to the Inspector-Adversary Game for APX_1^* (instead of APX_1).

We show that $\mathsf{Local}(\mathsf{APX}_1)$ (resp. $\mathsf{Local}(\mathsf{APX}_1^*)$) captures $\forall\Sigma_1^b$-consequences of APX_1 (resp. APX_1^*). First, Thapen [Tha24] shows that provably total TFNP problems reduce to their inspector-adversary game search problems; indeed, Thapen proves a more general result that for any theory $\mathsf{PV}_1(f)$ augmented with universal axioms about the f-oracle, provably total TFNP problems in the theory reduce to their corresponding Inspector-Adversary Game. In particular, we have:

Lemma 4.6 ([Tha24, Lemma 3.7])**.** *Let $R(x, y)$ be a PV_1-predicate. Then:*

- *if $\mathsf{APX}_1 \vdash \forall x\, \exists y\, R(x, y)$, then $R \leq_p \mathsf{Local}(\mathsf{APX}_1)$;*
- *if $\mathsf{APX}_1^* \vdash \forall x\, \exists y\, R(x, y)$, then $R \leq_p \mathsf{Local}(\mathsf{APX}_1^*)$;*

(both via single-query polynomial-time reducibility.)

The above lemma shows that each computational problem $\mathsf{Local}(\cdot)$ is strong enough for witnessing $\forall\Sigma_1^b$-consequences of the corresponding theory. The next lemma shows that each of them is also the weakest possible TFNP problem for witnessing.

Lemma 4.7. APX_1 *(resp.* APX_1^**) proves that* $\mathsf{Local}(\mathsf{APX}_1)$ *is total (resp.* $\mathsf{Local}(\mathsf{APX}_1^*)$ *is total).*

Proof. We work in APX_1 (resp. APX_1^*). We need to prove that for any given query algorithm $Q = (M_1, M_2, \ldots, M_k, M_{k+1})$, there exists a transcript given which the inspector does not win in k rounds. Indeed, we compute iterative as follows:

- Let $(C_1, \Delta_1) = M_1(\tau_1 := \varnothing)$, we define $p_1 \triangleq \mathsf{P}(C_1, \Delta_1)$
- Let τ_2 be the transcript before this step, and $(C_2, \Delta_2) = M_2(\tau_2)$, then $p_2 \triangleq \mathsf{P}(C_2, \Delta_2)$.
- $\ldots$
- Finally, $p_k \triangleq \mathsf{P}(C_k, \Delta_k)$.

Note that $p_1, \ldots, p_k$ can be computed via a P-oracle polynomial time algorithm, and can thus be defined. Moreover, with the axioms of APX_1 (resp. APX_1^*), we can conclude that the inspector must fail. □

5 Approximate Counting and Weak Pigeonhole Principles

In this section, we study upper bounds and (relativized) lower bounds for weak pigeonhole principles in bounded theories for approximate counting. In particular, we derive Theorem 1.5 and prove the remaining items of Theorem 1.8.

5.1 Unprovability of Relativized rwPHP in APX_1

In this section, we use Theorem 1.9 to show that $\mathsf{APX}_1(f, g) \nvdash \mathsf{rwPHP}(f, g)$ (Theorem 1.5). Recall that

$$\mathsf{rwPHP}(f, g) = \forall N\big(N > 1 \to \exists x < 2N\ (f(x) \geq N \vee g(f(x)) \neq x)\big).$$

As in the example following Theorem 1.9 in Section 1.2.3, we represent the two halves of the compression map by unary function symbols f_0 and f_1. Introduce one further unary function symbol h, and let $\mathcal{L} \triangleq \{f_0, f_1, h\}$. Define

$$\phi(x) \triangleq h(f_0(x)) \neq x \ \vee \ h(f_1(x)) \neq x \ \vee \ f_0(x) = f_1(x).$$

Thus $\forall x \, \neg\phi(x)$ asserts that h is a common left inverse of f_0 and f_1, and that $f_0(x) \neq f_1(x)$ for every x. This sentence has an infinite model with universe $\mathbb{N}$:

$$f_0(x) = 2x, \qquad f_1(x) = 2x + 1, \qquad h(y) = \lfloor y/2 \rfloor.$$

Consequently, Theorem 1.9 gives $\mathsf{APX}_1(\widehat{\mathcal{L}}) \nvdash \Theta_\phi$, where

$$\Theta_\phi \triangleq \forall N \geq 1 \ \left(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N) \ \rightarrow \ \exists x < N \ \phi^{[N]}(x) \right).$$

Finally, in order to obtain Theorem 1.5, it is enough to show that a proof of $\mathsf{rwPHP}(f, g)$ in $\mathsf{APX}_1(f, g)$ would yield a proof of Θ_ϕ in $\mathsf{APX}_1(\widehat{\mathcal{L}})$. For completeness, we describe the argument below.

If $\mathsf{APX}_1(f, g) \vdash \mathsf{rwPHP}(f, g)$ then $\mathsf{APX}_1(\widehat{\mathcal{L}}) \vdash \Theta_\phi$. For each $N > 1$, define

$$F_N(z) \triangleq \begin{cases} \widehat{f}_0(N, z), & z < N, \\ \widehat{f}_1(N, z - N), & N \leq z < 2N, \end{cases}$$

and

$$G_N(y) \triangleq \begin{cases} \widehat{h}(N, y), & \widehat{f}_0(N, \widehat{h}(N, y)) = y, \\ N + \widehat{h}(N, y), & \widehat{f}_0(N, \widehat{h}(N, y)) \neq y, \end{cases} \qquad (y < N).$$

Extend F_N and G_N by 0 outside $[2N]$ and $[N]$, respectively. These are $\mathsf{PV}_1(\widehat{\mathcal{L}})$ functions, defined uniformly in the parameter N. Under $\mathrm{Closed}_{\widehat{\mathcal{L}}}(N)$, they restrict to maps

$$F_N \colon [2N] \to [N], \qquad G_N \colon [N] \to [2N].$$

Given these definitions, it is not hard to check that the theory $\mathsf{PV}_1(\widehat{\mathcal{L}})$ proves that, for every $N > 1$, the following implication holds:

$$\begin{aligned} \mathrm{Closed}_{\widehat{\mathcal{L}}}(N) \ \wedge \ \exists z < 2N \ \big(F_N(z) \geq N \vee G_N(F_N(z)) \neq z \big) \\ \longrightarrow \quad \exists x < N \ \phi^{[N]}(x). \end{aligned}$$

Now suppose, towards a contradiction, that $\mathsf{APX}_1(f, g) \vdash \mathsf{rwPHP}(f, g)$. In this proof, replace calls to f and g by the $\mathsf{PV}_1(\widehat{\mathcal{L}})$ functions F_N and G_N, treating $N > 1$ as a parameter. Inside counting queries, replace each oracle circuit C by the oracle circuit C_N over $\widehat{\mathcal{L}}$ obtained by the same substitutions at its oracle gates. One can check by verifying axioms and induction (in the meta-language) on the complexity of proofs that this translation can be carried out so that the resulting proof is valid in $\mathsf{APX}_1(\widehat{\mathcal{L}})$. It therefore yields the N instance of $\mathsf{rwPHP}(F_N, G_N)$.

By the preceding reduction, we get that

$$\mathsf{APX}_1(\widehat{\mathcal{L}}) \vdash \forall N > 1 \ \left(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N) \ \rightarrow \ \exists x < N \ \phi^{[N]}(x) \right).$$

Finally, for $N = 1$, closure gives $\widehat{f}_0(1, 0) = \widehat{f}_1(1, 0) = 0$, so $\phi^{[1]}(0)$ holds. Together with the displayed consequence, this proves Θ_ϕ in $\mathsf{APX}_1(\widehat{\mathcal{L}})$, contradicting Theorem 1.9. Therefore $\mathsf{APX}_1(f, g) \nvdash \mathsf{rwPHP}(f, g)$, as required.

5.2 Unprovability of Relativized (d)wPHP in $\mathsf{True}_\forall\mathsf{APX}_1$

5.2.1 A Lower Bound for dwPHP

We first prove the following lower bound on inspector algorithms for dwPHP. Looking ahead, we will use this lower bound and the KPT witnessing theorem (Theorem 2.2) to show the unprovability of $\mathsf{dwPHP}(g)$ in the true universal theory of $\mathsf{APX}_1(g)$.

Lemma 5.1. *Let $k \in \mathbb{N}$ be a constant and let $\Psi_1, \dots, \Psi_k$ be true universal axioms of $\mathsf{APX}_1(g)$, where $\Psi_j = \forall \vec{u}\ \psi_j(\vec{u})$ with ψ_j quantifier-free. Let $t(n) \geq n$ be a non-decreasing function and $\mathcal{I}(1^n)$ be a (non-uniform) query algorithm of size $t = t(n)$, $g = \{g_n : \{0,1\}^{n-1} \to \{0,1\}^n\}$, $\mathsf{P} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that $\|\mathsf{P}(C,\Delta)\| \leq \|C\| + 2\|\Delta\|$. The algorithm (called inspector) may take one of the following three types of queries:*

- *(g queries): Given $x \in \{0,1\}^{m-1}$, the oracle outputs $g(x) \in \{0,1\}^m$.*
- *(P queries): Given an oracle circuit C and a string Δ, the oracle outputs $\mathsf{P}(C,\Delta)$.*
- *(inversion queries): Given $y \in \{0,1\}^n$, the oracle either outputs some string $x \in \{0,1\}^{n-1}$ such that $g(x) = y$, or $\perp$.*

Note that the algorithm $\mathcal{I}(1^n)$ can query the g oracle on input of any length $\leq t$, while for inversion queries it is only allowed to query y of length exactly *n. The inspector is said to succeed (resp. fail) if at least one (resp. neither) of the following two conditions holds:*

- *(range avoidance): the oracle outputs $\perp$ in an inversion query $y \in \{0,1\}^n$.*
- *(axiom violation): the inspector finds and outputs some $j \in [k]$ and a tuple $\vec{u}$ of strings such that the instance $\psi_j(\vec{u})$ evaluates to false by the oracle answers to g and P.*

Then for some constant $\varepsilon > 0$, if $t(n) \leq 2^{\varepsilon n}$ for all sufficiently large n, there are $\{g_n\}, \mathsf{P}$, and an inversion oracle such that for infinitely many n, $\mathcal{I}(1^n)$ fails.

Proof. Fix $k \in \mathbb{N}$ and axioms $\Psi_1, \dots, \Psi_k$, and let $d \geq 1$ be a constant such that for every $i \in [k]$, $\Psi_i = \forall \vec{u}\ \psi(\vec{u})$, $\psi(\vec{u})$ will make g-oracle queries and P-oracle queries on input length at most n^d when $n = \|\vec{u}\|$ is sufficiently large. Note that the constant d always exists as $\psi(\vec{u})$ are quantifier-free formulas in the language of $\mathsf{APX}_1(g)$ and are thus polynomial-time algorithms with (g, P)-oracles.

Let $\varepsilon > 0$ be a parameter to be determined, and let $N = \{n_1, n_2, \dots\}$ be a set of input lengths on which we will ensure that $\mathcal{I}$ does not succeed. Let $n \in N$, $t = t(n)$, $\tilde{t} = t^d$. Let $\ell = \ell(n)$ be a parameter to be determined. For $\vec{x} = (x_1, \dots, x_\ell), \vec{y} = (y_1, \dots, y_\ell)$, where x_i's are distinct and $\|y_i\| = \|x_i\| + 1 \leq \tilde{t}$ for every $i \in [\ell]$, we define $g_{\vec{x},\vec{y}}$ as

$$g_{\vec{x},\vec{y}}(x) \triangleq \begin{cases} y_i & x = x_i \\ 0^{\|x\|+1} & x \notin \{x_1, \dots, x_\ell\} \end{cases}$$

Let $D_n \subseteq \{0,1\}^{\tilde{t}}$ and let $D_n^{\leq m}$ be its projection to the first m bits. It is said to be pseudorandom if for every such $\vec{x}, \vec{y}$ and any oracle circuit $C : \{0,1\}^m \to \{0,1\}$ of size at most $\tilde{t}$, $D^{\leq m}$ $(1/\tilde{t})$-fools the circuit $C^{g_{\vec{x},\vec{y}}}$. Specifically,

$$\left| \Pr_{z \leftarrow D_n^{\leq m}}[C^{g_{\vec{x},\vec{y}}}(z) = 1] - \Pr_{z \leftarrow \{0,1\}^m}[C^{g_{\vec{x},\vec{y}}}(z) = 1] \right| \leq \frac{1}{2\tilde{t}}. \tag{5.1}$$

By the Hoeffding bound, we know that for any fixed $\vec{x}, \vec{y}$ and circuit C, if we sample a size-s multiset $D \subseteq \{0,1\}^{\tilde{t}}$ uniformly at random, $D_n^{\leq m}$ fools C with probability $1 - 2\exp(-s/(2\tilde{t}^2))$. There are at most $2^{2(\tilde{t}+1)\ell}$ choices of $(\vec{x}, \vec{y})$ and at most $2^{O(\tilde{t}\log\tilde{t})}$ oracle circuits of size at most $\tilde{t}$. Therefore, by union bound over $\vec{x}, \vec{y}, C$, there exists a pseudorandom set $D_n \subseteq \{0,1\}^{\tilde{t}}$ of size at most $s(n) \leq 100 \cdot t^{3d}\ell^2\sqrt{n}$ $(*)$.

Construction of Adversary. Let $n_0 \in \mathbb{N}$ be sufficiently large. We will construct the oracle g and P iteratively, starting with $g(z)$ and $\mathsf{P}(C, \Delta)$ being undetermined for any input z and C, Δ. In the i-th iteration, we will fix entries of g, P to force $\mathcal{I}(1^{n_i})$ fails; in addition, we will maintain the invariance that after the i-th iteration:

1. We fix g on all inputs of length at most $\tilde{t}(n_i)$ and P whenever $\|C\|, \|\Delta\| \leq \tilde{t}(n_i)$.
2. g agrees with $g_{\vec{x},\vec{y}}(x)$ on fixed strings for a list $\vec{x}, \vec{y}$ of length at most $\ell(n_i)$.
3. For any $\|C\|, \|\Delta\| \leq \tilde{t}(n_i)$, $|\mathsf{P}(C, \Delta) - \Pr_x[C^g(x) = 1]| \leq 1/\|\Delta\|$.
4. The procedure will not stall; see ($\diamond$) below.

For $i = 0$, we fix g to be all-zero on input length at most $\tilde{t}(n_0)$ and fix P to be the correct acceptance probability with respect to g. We now describe the i-th iteration of the procedure.

(*Iteration i*). Fix a pseudorandom set $D_n \subseteq \{0,1\}^{\tilde{t}}$. We run the algorithm $\mathcal{I}(1^{n_i})$ and answer queries if the corresponding entry for g and P are fixed. Otherwise we work as follows:

- For any g query on x, we fix $g(x) = 0^{\|x\|+1}$ and answer it.
- For any P query on circuit $C : \{0,1\}^m \to \{0,1\}$ and parameter Δ, we evaluate $C^g(z)$ for any $z \in D_n^{\leq m}$, during which any unfixed g-query will be fixed to output the all zero string. Let

$$p \triangleq \Pr_{z \leftarrow D_n^{\leq m}}[C^g(z) = 1], \tag{5.2}$$

  and fix and answer $\mathsf{P}(C, \Delta)$ to be any dyadic rational number that is $1/(2\|\Delta\|)$-close to p.
- For any inversion query $y \in \{0,1\}^{n_i}$, if we have set x such that $g(x) = y$, we answer x. Otherwise, we find any $x \in \{0,1\}^{n_i - 1}$ such that $g(x)$ has not been fixed, fix $g(x) = y$, and then answer x. We will ensure that such x always exists later ($\diamond$).

After that, we fix $g(x) = 0^{\|x\|+1}$ for any unfixed x with $\|x\| \leq \tilde{t}(n_i)$. Then for every circuit $C : \{0,1\}^m \to \{0,1\}$ of size at most $\tilde{t}(n_i)$ and any Δ of length at most $\tilde{t}(n_i)$, we fix $\mathsf{P}(C, \Delta)$ according to (5.2).

Let g, P be the union of all fixed g, P in the procedure for $i = 1, 2, \ldots$. We argue that for the functions g, P, it is always possible to answer inversion queries such that $\mathcal{I}(1^{n_i})$ fails for any i.

Proof of Correctness. We first show that the adversary will succeed if the four invariance properties hold. Notice that if $\mathcal{I}(1^{n_i})$ succeeds instead, it either outputs an instance $\vec{u}$ such that $\psi_j(\vec{u})$ is false for some $j \in [k]$, or finds a string y on which $g^{-1}(y) = \varnothing$. It can be verified that the latter is ruled out by the fourth property and by the construction procedure.

To rule out the former case, we first observe that after the i-th iteration, the functions g and P are fixed on input length $\leq \tilde{t}(n_i)$. Fix any $j \in [k]$. For any instance $\vec{u}$ provided by the inspector that intended to falsify Ψ_j, let $m = \|\vec{u}\|$, we know that $m \leq t(n_i)$ as the inspector runs in at most $t(n_i)$ time. Recall that we set $d \geq 1$ so that to evaluate $\psi_j(\vec{u})$, all g-oracle and P-oracle queries are of length at most $m^d \leq \tilde{t}(n_i)$, and thus are already fixed. Moreover, by the third invariant, we know that the P-oracle is a correct approximate-counting oracle. (Note that conditions (1) and (3) of Definition 3.5 follow immediately from the implementation of $\mathsf{P}(\cdot,\cdot)$.) Therefore, the instance $\vec{u}$ cannot falsify Ψ_j as it is true for any fixed g and correct approximate-counting oracle P.

It remains to verify the invariance properties.

(*Invariance 1*): This is straightforward from the construction procedure.

(*Invariance 2*): Note that $g(x)$ is set to be non-zero in the i-th iteration only if x was chosen to answer an inversion query made by $\mathcal{I}(1^{n_i})$. In the i-th iteration, we will answer at most $t(n_i)$ inversion queries from $\mathcal{I}(1^{n_i})$. Subsequently, the statement holds provided that $\ell(n_i) \geq t(n_i) + \ell(n_{i-1})$. Set $\ell(n_i) \triangleq n_i \cdot t(n_i)$ ($\star$).

(*Invariance 3*): Fix any $C \in \{0,1\}^m \to \{0,1\}$ and Δ of length at most $\tilde{t}(n_i)$. By invariance 2, we know that g agrees with $g_{\vec{x},\vec{y}}$ in the i-th iteration for a list $\vec{x}, \vec{y}$ of length $\ell(n_i)$. Recall that D_{n_i} is a pseudorandom set, and in particular $D_{n_i}^{\leq m}$ $(1/(2\tilde{t}(n_i)))$-fools any size-$\tilde{t}(n_i)$ oracle circuit with $g_{\vec{x},\vec{y}}$ to answer the oracle queries. Moreover, we know by the construction that whenever we fix P in the i-th iteration, it is fixed to be the empirical estimation over the set D_{n_i}. Subsequently, it agrees with the actual acceptance probability with error at most $1/(2\tilde{t}(n_i)) + 1/(2\|\Delta\|) \leq 1/\|\Delta\|$.

(*Invariance 4*): We will show that the number of fixed $g(x)$ for $x \in \{0,1\}^{n_i-1}$ during the run of $\mathcal{I}(1^{n_i})$ is at most 2^{n_i-2}; as a result, there is always an unfixed $x \in \{0,1\}^{n_i-1}$ when we answer inversion oracle queries. We set $n_i \geq 2\tilde{t}(n_{i-1})$ so $g(x)$ is unfixed for any $x \in \{0,1\}^{n_i-1}$ before the i-th iteration. In the i-th iteration, each g-oracle query or inversion query will fix at most 1 entry, and each P-oracle query will fix at most $s(n_i) \cdot t(n_i)$ entries of g. Therefore in total, the number of entries in g on input length $n_i - 1$ fixed by these three types of queries is at most

$$2 \cdot (t(n_i))^2 \cdot s(n_i) \leq 100 \cdot (t(n_i))^{5d} \cdot \ell(n_i)^2 \cdot \sqrt{n_i} \leq 100 \cdot (t(n_i))^{7d} \cdot n_i^{2.5},$$

where the first inequality follows from $(*)$ and the second inequality follows from $(\star)$. By setting $\varepsilon := 1/(10d)$ and the assumption $t(n_i) \leq 2^{\varepsilon n_i}$, this is at most $100 \cdot 2^{0.7n_i} \cdot n_i^{2.5} \leq 2^{n_i-2}$ for sufficiently large n_i. □

The following is then an immediate corollary of the KPT witnessing theorem (Theorem 2.2).

Theorem 5.2. $\mathsf{True}_\forall\mathsf{APX}_1(g) \nvdash \mathsf{dwPHP}(g)$. *In particular,* $\mathsf{APX}_1(g) \nvdash \mathsf{dwPHP}(g)$.

Proof. Let T be the set of true universal axioms of $\mathsf{APX}_1(g)$, and suppose that $\mathsf{PV}_1(\mathsf{P}, g) + \mathsf{T} \vdash \mathsf{dwPHP}(g)$. A proof invokes only finitely many axioms $\Psi_1, \ldots, \Psi_k \in \mathsf{T}$, so $\mathsf{PV}_1(\mathsf{P}, g) + \{\Psi_1, \ldots, \Psi_k\} \vdash \mathsf{dwPHP}(g)$. For simplicity, we assume that all axioms $\Psi_i = \forall u_i\ \psi(u_i)$ have only one variable. Then we know that $\mathsf{PV}_1(\mathsf{P}, g) \vdash \mathsf{dwPHP}(g) \vee \bigvee_{i\in[k]} \exists u_i\ \neg\psi_i(u_i)$. Subsequently we can see that:

$$\mathsf{PV}_1(\mathsf{P}, g) \vdash \forall n \in \mathsf{Log}\ \exists y \in \{0,1\}^n\ \exists (u_1, \ldots, u_k)\ \forall x \in \{0,1\}^{n-1}\ g(x) \neq y \vee \bigvee_{i\in[k]} \neg\psi_i(u_i).$$

By the KPT Witnessing Theorem (Theorem 2.2) applied to the (universal) theory $\mathsf{PV}_1(\mathsf{P}, g)$, there are a constant $r \in \mathbb{N}$ and a polynomial-size Inspector (or Student) that, on input 1^n, makes g and P oracle queries, that wins against any Adversary (or Teacher) in the following sense:

- In each round, the Inspector proposes (from previous transcript) a string $y \in \{0,1\}^n$ and $u_1, \ldots, u_k$. If one of u_i witnesses $\neg\psi(u_i)$ the Inspector wins.
- The Adversary outputs a string $x \in \{0,1\}^{n-1}$ such that $g(x) = y$; if impossible, the Inspector wins.

This matches the game in Lemma 5.1, and thus leads to a contradiction as the Inspector has complexity $\mathrm{poly}(n)$ and makes at most $r = O(1)$ inversion queries for g. □

Remark 5.3. The query lower bound and unprovability result supplement the results of [ILW23] (see also the follow up papers [CL24, Ila25b, RWZ26, LRZ26]), which shows the hardness of the range avoidance problem (see Definition 5.16) and the unprovability of dwPHP in PV_1 under cryptographic assumptions (see also [CLOW26] for an easy generalization to APX_1).

It is worth noting that since the function g is modelled as an oracle, the lower bound in Lemma 5.1 allows exponentially many inversion queries, whereas the hardness result of [ILW23, RWZ26] allows only $O(1)$ queries. In fact, assuming circuit lower bounds, there is an algorithm for solving range avoidance that makes only polynomially many queries (see [ILW23, Appendix A]).

5.2.2 A Lower Bound for wPHP

We now prove a similar lower bound for Weak PHP against APX_1. Standard models and true universal axioms of $\mathsf{APX}_1(f)$ are defined as in Definitions 3.5 and 3.6, with g replaced by the length-decreasing oracle $f = \{f_n : \{0,1\}^n \to \{0,1\}^{n-1}\}$.

Lemma 5.4. *Let $k \in \mathbb{N}$ be a constant and let $\Psi_1, \ldots, \Psi_k$ be true universal axioms of $\mathsf{APX}_1(f)$, where $\Psi_j = \forall \vec{u}\ \psi_j(\vec{u})$ with ψ_j quantifier-free. Let $t(n) \geq n$ be a non-decreasing function and $\mathcal{I}(1^n)$ be a (non-uniform) query algorithm in size $t = t(n)$, $f = \{f_n : \{0,1\}^n \to \{0,1\}^{n-1}\}$, $\mathsf{P} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that $\|\mathsf{P}(C, \Delta)\| \leq \|C\| + 2\|\Delta\|$. The algorithm (called inspector) may take one of the following two types of queries:*

- *(f queries): Given $x \in \{0,1\}^m$, the oracle outputs $f(x) \in \{0,1\}^{m-1}$.*
- *(P queries): Given an oracle circuit C and a string Δ, the oracle outputs $\mathsf{P}(C, \Delta)$.*

The algorithm $\mathcal{I}(1^n)$ can query the f oracle on inputs of any length $\leq t$. It is said to succeed (resp. fail) if at least one (resp. neither) of the following two conditions holds:

- *(collision): the inspector outputs distinct $x, x' \in \{0,1\}^n$ such that $f(x) = f(x')$.*
- *(axiom violation): the inspector outputs some $j \in [k]$ and a tuple $\vec{u}$ of strings such that the instance $\psi_j(\vec{u})$ evaluates to false by the oracle answers.*

Then for some constant $\varepsilon > 0$, if $t(n) \leq 2^{\varepsilon n}$ for all sufficiently large n, there are $\{f_n\}$ and P such that for infinitely many n, $\mathcal{I}(1^n)$ fails.

Proof. Fix $k \in \mathbb{N}$ and axioms $\Psi_1, \ldots, \Psi_k$, and let $d \geq 1$ be a constant such that for every $j \in [k]$, the evaluation of $\psi_j(\vec{u})$ makes f-oracle and P-oracle queries on inputs of length at most n^d when $n = \|\vec{u}\|$ is sufficiently large; such d exists as $\psi_j(\vec{u})$ are quantifier-free formulas in the language of $\mathsf{APX}_1(f)$ and are thus polynomial-time algorithms with (f, P)-oracles. Let $\tilde{t}(n) \triangleq (t(n))^d$. Let $N = \{n_1, n_2, \ldots\}$ be a set of input lengths with $n_i \geq 2\tilde{t}(n_{i-1})$ and n_0 sufficiently large. We fix f and P iteratively, maintaining that:

1. After the i-th iteration, f is fixed on all inputs of length $\leq \tilde{t}(n_i)$, P is fixed when $\|C\|, \|\Delta\| \leq \tilde{t}(n_i)$.
2. Every value of P fixed in the i-th iteration is within $1/\tilde{t}(n_i) \leq 1/\|\Delta\|$ of the acceptance probability $\Pr_z[C^f(z) = 1]$ with respect to the final f.

For $i = 0$, we fix f on all input lengths at most $\tilde{t}(n_0)$ to be the truncation function $x \mapsto x_1 \cdots x_{\|x\|-1}$ and fix P to be the correct acceptance probabilities with respect to f.

(*Iteration i*). Write $t = t(n_i)$ and $\tilde{t} = \tilde{t}(n_i)$. On every new input length $m \leq \tilde{t}$ other than n_i, fix f_m to be the truncation function. On input length n_i, choose a uniformly random function $f_{n_i} : \{0,1\}^{n_i} \to \{0,1\}^{n_i-1}$. Independently, sample a multiset $D \subseteq \{0,1\}^{\tilde{t}}$ of size $s \triangleq \tilde{t}^4$ uniformly at random.

Now f is fixed on all lengths at most $\tilde{t}$; for every pair (C, Δ) with $\|C\|, \|\Delta\| \leq \tilde{t}$ that is not fixed yet, where $C : \{0,1\}^m \to \{0,1\}$, we fix $\mathsf{P}(C, \Delta)$ to be any dyadic rational that is $1/(2\|\Delta\|)$-close to $\Pr_{z \leftarrow D^{\leq m}}[C^f(z) = 1]$ as in (5.2). Note that a circuit of size at most $\tilde{t}$ queries f only on inputs of length at most $\tilde{t}$, so these values are determined at this point and are not affected by later iterations. We claim that with positive probability over the choice of (f_{n_i}, D), the following two events hold simultaneously.

(*Honesty*): every value of P fixed in this iteration is within $1/\tilde{t}$ of $\Pr_z[C^f(z) = 1]$. Fix any choice of the function f as mentioned above. For every circuit C of size at most $\tilde{t}$, the empirical estimate over $s = \tilde{t}^4$ uniform samples deviates from the acceptance probability by more than $1/(2\tilde{t})$ with probability at most $2\exp(-2s/\tilde{t}^2) = 2\exp(-2\tilde{t}^2)$ by the Hoeffding bound. If this does not happen, then $P(C, \Delta)$ deviates

from the acceptance probability by at most $1/(2\tilde{t}) + 1/(2\|\Delta\|) \le 1/\|\Delta\|$. Taking a union bound over the at most $2^{O(\tilde{t}\log\tilde{t})}$ oracle circuits of size at most $\tilde{t}$, the event (*Honesty*) fails with probability at most $1/6$.

(*No collision*): $\mathcal{I}(1^{n_i})$ does not output distinct $x, x' \in \{0,1\}^{n_i}$ with $f(x) = f(x')$. Fix any D. We may assume that $\mathcal{I}$ queries both x and x' before it halts, as this only adds two f-queries; it then suffices to show that with probability at least $5/6$, no two queried entries of length n_i receive the same value. The answers received by the inspector are determined by the values of f at its at most t direct f-queries, together with the at most $s \cdot \tilde{t}$ f-queries when evaluating each of its at most t queried circuits over the samples of D. Hence the run of $\mathcal{I}(1^{n_i})$ queries at most

$$B \triangleq 2 + t + t \cdot s \cdot \tilde{t} \le 2\tilde{t}^6$$

entries of f; moreover, f is deterministic except on entries of length n_i.

For the analysis, we implement the answers to the entries of length n_i by *lazy sampling*: upon a query $x \in \{0,1\}^{n_i}$ that was not queried before, the oracle answers with an independent uniformly random string $v(x) \in \{0,1\}^{n_i - 1}$; repeated queries are answered according to v. The joint distribution of the answers is identical to that of the function f_{n_i} sampled upfront. By the union bound, the probability that any two queried entries receive the same value is at most $2B^2/2^{n_i}$. By setting $\varepsilon := 1/(24d)$ and the assumption $t(n_i) \le 2^{\varepsilon n_i}$, we have $B \le 2t^{6d} \le 2^{n_i/3}$ for sufficiently large n_i, and thus the event (*No collision*) fails with probability at most $2B^2/2^{n_i} \le 1/6$.

Since (*Honesty*) fails with probability at most $1/6$ for every fixed f_{n_i}, and (*No collision*) fails with probability at most $1/6$ for every fixed D, with positive probability both events hold; fix such a pair (f_{n_i}, D) and proceed to the next iteration.

Finally, suppose that $\mathcal{I}(1^{n_i})$ succeeds. It cannot output a collision by (*No collision*). It cannot output an instance $(j, \vec{u})$ falsifying Ψ_j either: since $\|\vec{u}\| \le t(n_i)$ and by the choice of d, all f-oracle and P-oracle queries made to evaluate $\psi_j(\vec{u})$ concern lengths at most $\|\vec{u}\|^d \le \tilde{t}(n_i)$ and thus are already fixed, and the pair (f, P) is a standard model of $\mathsf{APX}_1(f)$: every fixed value of P is within $1/\|\Delta\|$ of the acceptance probability with respect to the final f, is a rational in $[0,1]$, and equals $\mathrm{Val}^f(C)$ exactly whenever C does not read its random input (in which case all evaluations over $D^{\le m}$ agree). Therefore $\mathcal{I}(1^{n_i})$ fails for every i. □

The following is then an immediate corollary of the Herbrand witnessing theorem.

Theorem 5.5. $\mathsf{True}_\forall\mathsf{APX}_1(f) \nvdash \mathsf{wPHP}(f)$. *In particular,* $\mathsf{APX}_1(f) \nvdash \mathsf{wPHP}(f)$.

Proof. Let T be the set of true universal axioms of $\mathsf{APX}_1(f)$. Suppose that $\mathsf{PV}_1(\mathsf{P}, f) + \mathsf{T} \vdash \mathsf{wPHP}(f)$. A proof invokes only finitely many axioms $\Psi_1, \dots, \Psi_k \in \mathsf{T}$, and for simplicity we assume that all axioms $\Psi_i = \forall u_i\, \psi_i(u_i)$ have only one variable. As in the proof of Theorem 5.2, we can see that:

$$\mathsf{PV}_1(\mathsf{P}, f) \vdash \forall n \in \mathsf{Log}\, \exists x, x' \in \{0,1\}^n\, \exists (u_1, \dots, u_k)\, \big(x \neq x' \wedge f(x) = f(x')\big) \vee \bigvee_{i \in [k]} \neg\psi_i(u_i).$$

The Herbrand witnessing theorem applied to the universal theory $\mathsf{PV}_1(\mathsf{P}, f)$ yields a *single* polynomial-size inspector that, for every f and P, outputs a tuple $(x, x', u_1, \dots, u_k)$ satisfying the displayed disjunction: the inspector evaluates the finitely many Herbrand candidates extracted from the proof and outputs the first one that verifies. This inspector succeeds on every input length in the sense of Lemma 5.4, which contradicts Lemma 5.4 as $\mathrm{poly}(n) \le 2^{\varepsilon n}$ for all sufficiently large n. □

Remark 5.6. This unprovability result is closely related to an early result of Thapen [Tha05] that $S^1_2 + \mathsf{dwPHP} \nvdash \mathsf{wPHP}$ (and therefore $\mathsf{APC}_1 \nvdash \mathsf{wPHP}$) in the relativized setting. Thapen's result relies on the witnessing theorem for $S^1_2 + \mathsf{dwPHP}$ (due to Wilkie [Kra95]) that if $S^1_2 + \mathsf{dwPHP} \vdash \forall x\, \exists y\, \phi(x, y)$ for a quantifier-free ϕ, then there is a *zero-error* algorithm that finds such y given x. It then suffices to use the standard fact that a random function is likely collision-resistant. Therefore, the unprovability result naturally generalizes to the theory that contains

all true $\forall\Sigma^b_0(\mathsf{PV}_1)$ consequences of $S^1_2 + \mathsf{dwPHP}$, for which the witnessing theorem still applies.

Theorem 5.5 is technically incomparable as the true universal theory $\mathsf{APX}_1(f)$ could contain sentences that are not in $S^1_2 + \mathsf{dwPHP}$ (or even its extension with all true $\forall\Sigma^b_0(\mathsf{PV}_1)$ sentences). Nevertheless, the proof follows exactly the same route: For each fixed f, any $\forall\Sigma^b_1(\mathsf{PV}_1(f))$-consequence, including $\mathsf{wPHP}(f)$ admits a zero-error witnessing algorithm (i.e. answering P-oracle queries by empirical estimation). Then take f to be a random function, which is likely collision-resistant.

5.3 Provability of (d)wPHP in APX^*_1

We now show both wPHP and dwPHP are provable in APX^*_1, the bounded theory for multiplicative-error approximate counting. We note that a computational version of these upper bounds, specifically

$$\mathsf{WeakPigeon} \in \mathsf{FP}^{\mathsf{prSBP}} \qquad \text{and} \qquad \mathsf{Avoid} \in \mathsf{FP}^{\mathsf{prSBP}},$$

is probably folklore (see [KP24, HV26] for discussions of the latter inclusion). Our main contribution is the proof complexity upper bounds of the principles in the (seemingly weak) theory APX^*_1. Moreover, by Thapen's witnessing theorem (see Lemma 4.6), we obtain (possibly) better upper bounds for PWPP, which might be of independent interest (see Theorem 5.14).

5.3.1 Basic Counting Functionalities in APX^*_1

We first develop basic counting functionalities in the theory APX^*_1, analogous to [CLOW26, Section 3] for APX_1. These results follow from essentially the same idea in [CLOW26]; we defer the proofs of Lemmas 5.7 to 5.9 to Section A.2.

Lemma 5.7 (Global Consistency for APX^*_1)**.** *APX^*_1 proves the following statement. Let $n, \delta^{-1} \in \mathsf{Log}$ and $C_1, C_2 : \{0,1\}^n \to \{0,1\}$ be circuits such that $C_1(x) \le C_2(x)$ for every $x \in \{0,1\}^n$. Then*

$$(1-\delta)\cdot \mathsf{P}_\delta(C_1) \lesssim^* (1+\delta)\cdot \mathsf{P}_\delta(C_2).$$

In particular, if $C_1(x) = C_2(x)$ for every $x \in \{0,1\}^n$, then both $(1-\delta)\cdot \mathsf{P}_\delta(C_1) \lesssim^* (1+\delta)\cdot \mathsf{P}_\delta(C_2)$ and $(1-\delta)\cdot \mathsf{P}_\delta(C_2) \lesssim^* (1+\delta)\cdot \mathsf{P}_\delta(C_1)$.

We will need the fact that if the acceptance probability of a circuit is noticeably smaller than $(1-\delta)\cdot 2^{-n}$, then it is unsatisfiable. This can be viewed as a multiplicative-error counterpart of [CLOW26, Lemma 3.7] on the acceptance probability of the "less-than-t" circuit.

Lemma 5.8 (Non-Existence Lemma for APX^*_1)**.** *APX^*_1 proves the following statement. Let $n, \delta^{-1} \in \mathsf{Log}$ satisfying $\delta < 1$ and $C : \{0,1\}^n \to \{0,1\}$ be a circuit. If there is $x \in \{0,1\}^n$ such that $C(x) = 1$, then $(1-\delta)\cdot 2^{-n} \lesssim^* \mathsf{P}_\delta(C)$. (Or equivalently, if $(1+\beta)\cdot \mathsf{P}_\delta(C) < (1-\delta)\cdot 2^{-n}$ for some $\beta^{-1} \in \mathsf{Log}$, then $C(x) = 0$ for every $x \in \{0,1\}^n$.)*

We will need the following generic averaging argument, which implies several useful facts. The counterpart for APX_1 is in [CLOW26, Section 3.2.3].

Lemma 5.9 (Averaging on Linear Combination for APX^*_1)**.** *APX^*_1 proves the following statement. Let $n, m, \delta^{-1} \in \mathsf{Log}$, $C_1, \ldots, C_m : \{0,1\}^n \to \{0,1\}$ be circuits, and $\lambda^+_1, \ldots, \lambda^+_m, \lambda^-_1, \ldots, \lambda^-_m \in \mathbb{Q}$ be non-negative. Let*

$$\mu^+ \triangleq \sum_{i\in[m]} \lambda^+_i \cdot \mathsf{P}_\delta(C_i) \qquad \text{and} \qquad \mu^- \triangleq \sum_{i\in[m]} \lambda^-_i \cdot \mathsf{P}_\delta(C_i).$$

Then for every $\beta^{-1} \in \mathsf{Log}$, there is $z \in \{0,1\}^n$ such that

$$\frac{(1+\beta)(1+\delta)\cdot\sum_{i\in[m]}\lambda_i^+\cdot C_i(z)}{(1-\delta)\cdot\sum_{i\in[m]}\lambda_i^-\cdot C_i(z)} \geq \frac{\mu^+}{\mu^-},$$

where $a/b \geq c/d$ is an abbreviation of $ad \geq bc$. Moreover, if $\mu^+ > 0$ and $\mu^- > 0$, then z can be chosen such that the inequality is strict, i.e., $ad > bc$.

Corollary 5.10 (Existence Lemma for APX_1^*)**.** APX_1^* *proves the following statement. Let $n, \delta^{-1} \in \mathsf{Log}$, $\delta < 1$, and $C : \{0,1\}^n \to \{0,1\}$ be a circuit. If $\mathsf{P}_\delta(C) > 0$, then there is $x \in \{0,1\}^n$ such that $C(x) = 1$.*

Proof. We work in APX_1^*. Fix $n, \delta^{-1} \in \mathsf{Log}$. Let $\mu \triangleq \mathsf{P}_\delta(C) > 0$ and $\beta = 0.01$. By Averaging on Linear Combination for APX_1^* with circuits C and the constant-1 circuit $O : \{0,1\}^n \to \{0,1\}$ (that does not read the input and outputs 1) and coefficients $\lambda_C^+ = \lambda_O^- = 1$, $\lambda_C^- = \lambda_O^+ = 0$, there exists $x \in \{0,1\}^n$ such that

$$\frac{(1+\beta)(1+\delta)C(x)}{(1-\delta)O(x)} \geq \frac{\mu}{\mathsf{P}_\delta(O)} = \frac{\mu}{1}.$$

Note that $O(x) = 1$, and the equality follows from the Boundary Axiom for APX_1^*. This further implies that $C(x) \geq \frac{(1-\delta)\mu}{(1+\beta)(1+\delta)} > 0$; subsequently, $C(x) = 1$. □

Note that the existence lemma for APX_1 in [CLOW26] requires $\mathsf{P}_\delta(C) > \delta + \beta$ for some $\beta^{-1} \in \mathsf{Log}$, whereas in APX_1^* it suffices that $\mathsf{P}_\delta(C)$ is positive.

Corollary 5.11 (Inclusion-Exclusion for APX_1^*)**.** APX_1^* *proves the following statement. Let $n, \delta^{-1} \in \mathsf{Log}$ and $C_1, C_2, C_\vee, C_\wedge : \{0,1\}^n \to \{0,1\}$ be circuits such that $C_\vee(x) = C_1(x) \vee C_2(x)$ and $C_\wedge(x) = C_1(x) \wedge C_2(x)$ for every $x \in \{0,1\}^n$. Then*

$$(1-\delta)\cdot\big(\mathsf{P}_\delta(C_\vee) + \mathsf{P}_\delta(C_\wedge)\big) \lesssim^* (1+\delta)\cdot\big(\mathsf{P}_\delta(C_1) + \mathsf{P}_\delta(C_2)\big) \quad \textit{and} \tag{5.3}$$

$$(1-\delta)\cdot\big(\mathsf{P}_\delta(C_1) + \mathsf{P}_\delta(C_2)\big) \lesssim^* (1+\delta)\cdot\big(\mathsf{P}_\delta(C_\vee) + \mathsf{P}_\delta(C_\wedge)\big). \tag{5.4}$$

In particular, if $C_1(x) \wedge C_2(x) = 0$ for every $x \in \{0,1\}^n$, then $(1-\delta)\cdot\mathsf{P}_\delta(C_\vee) \lesssim^* (1+\delta)\cdot\big(\mathsf{P}_\delta(C_1)+\mathsf{P}_\delta(C_2)\big)$ and $(1-\delta)\cdot\big(\mathsf{P}_\delta(C_1) + \mathsf{P}_\delta(C_2)\big) \lesssim^* (1+\delta)\cdot\mathsf{P}_\delta(C_\vee)$.

Proof. We work in APX_1^*. Fix $n, \delta^{-1} \in \mathsf{Log}$ and circuits $C_1, C_2, C_\vee, C_\wedge : \{0,1\}^n \to \{0,1\}$. We only prove (5.4); other parts are left as an exercise.[5] Let $\lambda_1^+ = \lambda_2^+ = \lambda_\vee^- = \lambda_\wedge^- = 1$ and $\lambda_\vee^+ = \lambda_\wedge^+ = \lambda_1^- = \lambda_2^- = 0$. We apply Averaging on Linear Combination for APX_1^* to circuits $C_1, C_2, C_\vee, C_\wedge$ and coefficients λ_i^+, λ_i^-. Let

$$\mu^+ \triangleq \mathsf{P}_\delta(C_1) + \mathsf{P}_\delta(C_2), \qquad \mu^- \triangleq \mathsf{P}_\delta(C_\vee) + \mathsf{P}_\delta(C_\wedge).$$

We may assume that $\mu^+ > 0$, as otherwise (5.4) is trivial. We may also assume that $\mu^- > 0$. This is because if $\mu^- = 0$, then $\mathsf{P}_\delta(C_\vee) = \mathsf{P}_\delta(C_\wedge) = 0$ by the Basic Axiom for APX_1^*. Since $C_i(x) \leq C_\vee(x)$ for every $x \in \{0,1\}^n$ and $i \in \{1,2\}$, Global Consistency for APX_1^* gives $(1-\delta)\cdot\mathsf{P}_\delta(C_i) \lesssim^* (1+\delta)\cdot\mathsf{P}_\delta(C_\vee) = 0$, i.e., $(1-\delta)\cdot\mathsf{P}_\delta(C_i) \leq 0$; therefore $(1-\delta)\cdot\mu^+ \leq 0$, and (5.4) holds.

Assume that $\mu^+, \mu^- > 0$. Then (the strict form of) Averaging on Linear Combination for APX_1^* implies that for every $\beta^{-1} \in \mathsf{Log}$, there exists $z \in \{0,1\}^n$ such that

$$\frac{\mu^+}{\mu^-} < \frac{(1+\beta)(1+\delta)(C_1(z)+C_2(z))}{(1-\delta)(C_\vee(z)+C_\wedge(z))}.$$

The inequality means that $(1-\delta)(C_\vee(z)+C_\wedge(z))\cdot\mu^+ < (1+\beta)(1+\delta)(C_1(z)+C_2(z))\cdot\mu^-$. Since the LHS is non-negative, $C_1(z)+C_2(z) > 0$. We can then cancel the common positive factor $C_1(z)+C_2(z) = C_\vee(z)+C_\wedge(z)$, which gives (5.4). □

[5] For (5.3), the case $\mathsf{P}_\delta(C_1) = \mathsf{P}_\delta(C_2) = 0$ additionally needs the Non-Existence Lemma for APX_1^*, which implies that C_1 and C_2, and hence $C_\vee$, are identically 0.

5.3.2 An Upper Bound for dwPHP

Theorem 5.12. $\mathsf{APX}_1^*(g) \vdash \mathsf{dwPHP}(g)$. *Consequently,* $\mathsf{APX}_1^* \vdash \mathsf{dwPHP}$.

Indeed, we will prove a slightly stronger version: there exists a (g, P)-oracle algorithm $A(1^n)$ that finds $y \in \{0,1\}^n$ that is not in the range of g. This immediately implies Theorem 5.12 by fixing $y \triangleq A(1^n)$.

Theorem 5.13. *Let g be a function symbol with intended output length $\|g(u)\| = \|u\| + 1$. There is a $\mathsf{PV}_1(g, \mathsf{P})$ function A such that the following is provable in $\mathsf{APX}_1^*(g)$: for every $n \in \mathsf{Log}$, $A(1^n)$ outputs $y \in \{0,1\}^n$ such that for every $x \in \{0,1\}^{n-1}$, $g(x) \neq y$.*

Consequently, there is a $\mathsf{PV}_1(\mathsf{P})$ function B such that APX_1^* proves the following: for every $n \in \mathsf{Log}$ and circuit $C : \{0,1\}^{n-1} \to \{0,1\}^n$, $B(C)$ outputs $y \in \{0,1\}^n$ such that for any $x \in \{0,1\}^{n-1}$, $C(x) \neq y$.

Proof. We argue in $\mathsf{APX}_1^*(g)$. Let $n \in \mathsf{Log}$ (we may assume $n \geq 1$) and $\eta^{-1} \in \mathsf{Log}$ be a parameter to be determined (where $\eta \leq 1/100$), and $\kappa \triangleq (1-\eta)/(1+\eta)^2$. For $y \in \{0,1\}^{\leq n}$, let $T_y : \{0,1\}^{n-1} \to \{0,1\}$ be the g-oracle circuit such that $T_y(x) = 1$ iff y is a prefix of $g(x)$, and $p(y) \triangleq \mathsf{P}_\eta(T_y)$. In particular, for $y = \varepsilon$ being the empty string, T_y is a circuit that is functionally equivalent to the constant-1 circuit. By the Boundary Axiom for APX_1^* and Global Consistency for APX_1^*, we can show that $p(\varepsilon) \leq (1+\eta)^2/(1-\eta) = \kappa^{-1}$.

Consider the (g, P)-oracle algorithm $A(1^n)$ that will find a string $y \in \{0,1\}^n$ that is not in the range of g. The algorithm starts with $y \leftarrow \varepsilon$, and iteratively constructs y such that $p(y)$ is sufficiently small.

- For $\|y\| < n$, it can be proved in PV_1 that $T_{y0}(x) \wedge T_{y1}(x) = 0$ for every $x \in \{0,1\}^{n-1}$, and $T_y(x) = T_{y0}(x) \vee T_{y1}(x)$ (using the length axiom $\|g(x)\| = n$). By the Inclusion-Exclusion for APX_1^* with $C_1 \triangleq T_{y0}, C_2 \triangleq T_{y1}$, and $C_\vee \triangleq T_y$, we can prove that

$$p(y0) + p(y1) = \mathsf{P}_\eta(T_{y0}) + \mathsf{P}_\eta(T_{y1}) \leq \frac{(1+\eta)^2}{1-\eta} \cdot \mathsf{P}_\eta(T_y) = \kappa^{-1} \cdot p(y).$$

- When $\|y\| < n$, the algorithm chooses $b \in \{0,1\}$ such that $p(yb) \leq \frac{1}{2\kappa} \cdot p(y)$, and sets $y \leftarrow yb$.

By induction on a quantifier-free formula, we can prove that the algorithm $A(1^n)$ outputs $y \in \{0,1\}^n$ such that $p(y) \leq \kappa^{-1} \cdot (2\kappa)^{-n} = 2^{-n} \cdot \kappa^{-(n+1)}$.

We now set the parameters. Let $\eta \triangleq 1/(100n) \leq 1/100$, so we have $\kappa \geq 1 - 5\eta$ and $\kappa^{n+1} > 0.8$. This means that

$$(1+\eta) \cdot p(y) \leq (1+\eta) \cdot \kappa^{-(n+1)} \cdot 2^{-n} < 1.3 \cdot 2^{-n} < (1-\eta) \cdot 2^{-(n-1)}.$$

Since T_y has $n-1$ input bits, by the Non-Existence Lemma for APX_1^* (with $\delta \triangleq \eta$ and $\beta \triangleq \eta$), we know that $T_y(x) = 0$ for every $x \in \{0,1\}^{n-1}$. This completes the proof, as $T_y(x) = 0$ means that y is not a prefix of $g(x)$, and in particular $g(x) \neq y$. □

5.3.3 An Upper Bound for wPHP

Theorem 5.14. $\mathsf{APX}_1^*(f) \vdash \mathsf{wPHP}(f)$. *Consequently,* $\mathsf{APX}_1^* \vdash \mathsf{wPHP}$.

Proof. We argue in $\mathsf{APX}_1^*(f)$. Let $n \in \mathsf{Log}$ and $\eta^{-1} \in \mathsf{Log}$ be a parameter to be determined (where $\eta \leq 1/100$), and $\kappa \triangleq (1-\eta)/(1+\eta)^2$. For $y \in \{0,1\}^{\leq n-1}$, let $T_y : \{0,1\}^n \to \{0,1\}$ be the f-oracle circuit such that $T_y(x) = 1$ iff y is a prefix of $f(x)$, and $p(y) \triangleq \mathsf{P}_\eta(T_y)$. In particular, for $y = \varepsilon$ being the empty string, T_y is a circuit that is functionally equivalent to the constant-1 circuit. By the Boundary Axiom for APX_1^* and Global Consistency for APX_1^*, we can show that $p(\varepsilon) \geq (1-\eta)/(1+\eta)^2 = \kappa$.

Consider the (f, P)-oracle algorithm $A(1^n)$ that will find a string $y \in \{0,1\}^{n-1}$ such that $|f^{-1}(y)| \geq 2$. The algorithm starts with $y \leftarrow \varepsilon$, and iteratively constructs y such that $p(y)$ is sufficiently large.

- It can be proved that $T_{y0}(x) \wedge T_{y1}(x) = 0$ for every $x \in \{0,1\}^n$, and $T_y(x) = T_{y0}(x) \vee T_{y1}(x)$. By the Inclusion-Exclusion for APX_1^* with $C_1 \triangleq T_{y0}, C_2 \triangleq T_{y1}$, and $C_\vee \triangleq T_y$, we can prove that

$$p(y0) + p(y1) = \mathsf{P}_\eta(T_{y0}) + \mathsf{P}_\eta(T_{y1}) \geq \frac{1-\eta}{(1+\eta)^2} \cdot \mathsf{P}_\eta(T_y) = \kappa \cdot p(y).$$

- When $\|y\| < n-1$, the algorithm chooses $b \in \{0,1\}$ such that $p(yb) \geq \frac{\kappa}{2} \cdot p(y)$, and sets $y \leftarrow yb$.

By induction on a quantifier-free formula, we can prove that the algorithm $A(1^n)$ outputs $y \in \{0,1\}^{n-1}$ such that $p(y) \geq 2^{-(n-1)} \cdot \kappa^n$.

For $x \in \{0,1\}^{\leq n}$, let $T_{y,x} : \{0,1\}^{n-\|x\|} \to \{0,1\}$ be the circuit such that $T_{y,x}(z) = 1$ iff $f(z \circ x) = y$. In particular, $T_{y,\varepsilon}$ is functionally equivalent to T_y. Let $q_y(x) \triangleq \mathsf{P}_\eta(T_{y,x})$; by Global Consistency for APX_1^*, we know that $q_y(\varepsilon) \geq \kappa \cdot p(y)$. Next, we consider the following (f,P)-oracle algorithm $E(1^n, y)$ that outputs $x \in \{0,1\}^{<n}$ such that $q_y(0x) > 0$ and $q_y(1x) > 0$. The algorithm starts with $x \leftarrow \varepsilon$ and constructs x iteratively as follows.

- Note that for $b \in \{0,1\}$, $T_{y,bx} \triangleq \mathrm{Fix}_b(T_{y,x})$. Therefore by the Local Consistency Axiom for APX_1^*, we know that

$$\frac{q_y(0x) + q_y(1x)}{2} = \frac{\mathsf{P}_\eta(T_{y,0x}) + \mathsf{P}_\eta(T_{y,1x})}{2} \geq \frac{1-\eta}{(1+\eta)^2} \cdot \mathsf{P}_\eta(T_{y,x}) = \kappa \cdot q_y(x).$$

- In each iteration, the algorithm computes $q_y(0x)$ and $q_y(1x)$. If both are positive, it succeeds and outputs x. Otherwise, it sets $x \leftarrow bx$ for the $b \in \{0,1\}$ maximizing $q_y(bx)$, so $q_y(bx) \geq 2\kappa \cdot q_y(x)$.

By induction on a quantifier-free formula, we can prove that in the i-th iteration, the algorithm maintains $x \in \{0,1\}^i$ such that $q_y(x) \geq (2\kappa)^i \cdot q_y(\varepsilon) \geq (2\kappa)^i \cdot 2^{-(n-1)} \cdot \kappa^{n+1}$. If the algorithm does not succeed after n iterations, we will construct $x \in \{0,1\}^n$ such that $q_y(x) \geq (2\kappa)^n \cdot 2^{-(n-1)} \cdot \kappa^{n+1} = 2\kappa^{2n+1}$.

We now set the parameters. Let $\eta \triangleq 1/(100n) \leq 1/100$, so we have $\kappa \geq 1 - 5\eta$ and $\kappa^{2n+1} > 0.8$. This means if the algorithm does not succeed, we will have $q_y(x) > 1$, which violates the Basic Axiom for APX_1^*. Subsequently, the algorithm $E(1^n, y)$ must succeed in finding $x \in \{0,1\}^{<n}$ such that $q_y(0x), q_y(1x) > 0$. Then by the Existence Lemma for APX_1^*, there exist z_0, z_1 such that

$$T_{y,0x}(z_0) = 1, \text{ i.e., } f(z_0 \circ 0 \circ x) = y, \qquad T_{y,1x}(z_1) = 1, \text{ i.e., } f(z_1 \circ 1 \circ x) = y.$$

This completes the proof, as $x_0 \triangleq z_0 \circ 0 \circ x$ and $x_1 \triangleq z_1 \circ 1 \circ x$ are distinct strings of length n such that $f(x_0) = f(x_1)$. □

Complexity-theoretic implication. It is worth mentioning that Theorem 5.14 has an immediate consequence in complexity theory by the witnessing theorems of Thapen [Tha24] (see Lemma 4.6). Recall that $\mathsf{Local}(\mathsf{APX}_1)$ (resp. $\mathsf{Local}(\mathsf{APX}_1^*)$) is the total search problem that, given a non-uniform query algorithm representing a strategy of the inspector in the Inspector-Adversary game for the theory APX_1 (resp. APX_1^*), outputs a transcript on which the inspector fails (see Definitions 4.4 and 4.5). By definition, $\mathsf{Local}(\mathsf{APX}_1)$ and $\mathsf{Local}(\mathsf{APX}_1^*)$ are in TFNP as the standard models of $\mathsf{APX}_1, \mathsf{APX}_1^*$ give a winning strategy for the adversary.

We first recall the definition of some search problems.

Definition 5.15 (WeakPigeon; see, e.g., [Jeř16])**.** WeakPigeon is defined as the following search problem in TFNP: Given a circuit $C : \{0,1\}^n \to \{0,1\}^{n-1}$, a solution is a pair of distinct strings $x_1, x_2 \in \{0,1\}^n$ such that $C(x_1) = C(x_2)$. The complexity class PWPP of search problems is defined as the TFNP problems that are polynomial-time reducible to WeakPigeon.

Definition 5.16 (Avoid; see, e.g., [Kor25] and references therein)**.** Avoid is defined as the following search problem in $\mathsf{TF\Sigma_2}$: Given a circuit $C : \{0,1\}^{n-1} \to \{0,1\}^n$, a solution is a string $y \in \{0,1\}^n$ such that for every $x \in \{0,1\}^{n-1}$, $C(x) \neq y$.

Theorem 5.17. $\mathsf{WeakPigeon} \leq_p \mathsf{Local}(\mathsf{APX}_1^*)$ *via single-query polynomial-time reducibility.*

Proof. This follows directly from the unrelativized version of Theorem 5.14 and Lemma 4.6. □

We note that Theorem 5.12 or Theorem 5.13 cannot be used to prove $\mathsf{Avoid} \leq_p \mathsf{Local}(\mathsf{APX}_1^*)$ as Avoid is not a TFNP problem. In fact, under plausible assumptions, $\mathsf{Avoid} \not\leq_p \mathsf{TFNP}$ since otherwise there will be an NP search algorithm for Avoid (see [CL24, Ila25b, RWZ26, LRZ26]).

6 Finite Combinatorial Principles, Infinite Models, and the Limits of APX_1

In this section, we establish Theorem 1.9 and Corollary 1.12. The proof of Theorem 1.9 is a combination of two steps. First, we introduce a certain game played between an inspector and an adversary. We show in Section 6.1 that an infinite model provides a winning strategy for the adversary in this game. In contrast, we show in Section 6.3 that a proof of the relevant sentence in the theory yields an inspector that wins the game. Combining these two results implies the unprovability in Theorem 1.9.

Theorem 1.9 (Riis-style dichotomy for relativized APX_1)**.** *Let $\phi(x_1, \ldots, x_k)$ be a formula in the language $\mathcal{L}$ with equality. If $\forall x_1 \cdots \forall x_k \, \neg\phi(x_1, \ldots, x_k)$ has an infinite model, then*

$$\mathsf{APX}_1(\widehat{\mathcal{L}}) \nvdash \forall N \geq 1 \ \left(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N) \ \to \ \exists x_1, \ldots, x_k < N \ \phi^{[N]}(x_1, \ldots, x_k)\right).$$

For simplicity, we work with quantifier-free $\mathcal{L}$-formulas in Section 6.1 below. In Section 6.2 and Section 6.3, we describe how standard techniques allow us to reduce the case of quantified formulas to the simpler setting of quantifier-free formulas.

6.1 Winning Strategies from Infinite Models

In this section, we fix some finite first-order language $\mathcal{L}$ consisting of function and constant symbols, and a *quantifier-free* formula $\phi(v_1 \ldots, v_k)$ over $\mathcal{L}$ with equality and exactly k free variables $v_1, \ldots, v_k$. We assume that ϕ exhausts all the symbols in $\mathcal{L}$, so that ϕ determines $\mathcal{L}$ (this is for notation, so that various parameters with dependence on $\mathcal{L}, \phi$ can be defined only in terms of ϕ). We will consider a game $\mathcal{G}_\phi(N)$ played over finite universes of size N, where ϕ is fixed and N grows to infinity. We start by introducing some terminology necessary to describe the game:

Definition 6.1 (Partial Structures, witnesses and decision trees)**.** A partial $\mathcal{L}$-structure ρ on $[N]$ assigns a value c^ρ in $[N]$ to some subset of the constant symbols and assigns a partial function $f^\rho : [N]^d \to [N]$ to each d-ary function symbol f. For two partial structures ρ, ρ' we say $\rho' \supseteq \rho$ ("ρ' extends ρ") if every constant symbol defined in ρ is defined and has the same value in ρ', and every partial function in ρ' extends the corresponding partial function in ρ. For an assignment $x_1, \ldots, x_k \in [N]$ to the variables $v_1, \ldots, v_k$ of ϕ, we say that $\rho \models \phi[\bar{x}]$ if each term in $\phi[\bar{x}]$ has a well-defined value in $[N]$ under the partial structure ρ and the assignment $\bar{x}$ (according to the natural interpretation[6]), and after substituting these values for the terms, the sentence $\phi[\bar{x}]$ is true. We say that a partial structure is "ϕ-witnessing" if there exists some $(x_1, \ldots, x_k) \in [N]^k$ such that $\rho \models \phi[x_1, \ldots, x_k]$.

[6] A formal definition involves an induction on the depth of terms. If $t = v_i$ its value is x_i; if $t = c$ for a constant symbol c, its value is c^ρ if c^ρ is defined, otherwise it is undefined. If $t = f(t_1, \ldots, t_d)$ for f a function symbol of arity d, and each t_i has a defined value $z_i \in [N]$, and $f^\rho(\bar{z})$ is defined and equals y, then the value of t is y; in all other cases it is undefined.

An (N, ϕ) decision tree T is a finite rooted tree. Each internal node queries either a constant c or a "table entry" $f(\bar{x})$, where $\bar{x} \in [N]^{\mathrm{arity}(f)}$; this node has one child for every possible answer in $[N]$. Every leaf $\ell \in T$ is labeled with a bit value $b_\ell \in \{0, 1\}$. For two such trees T_1, T_2, we define $T_1 \circ T_2$ to be the tree obtained by attaching a copy of T_2 to each leaf of T_1 (the root of each copy is identified with its associated leaf). We will not bother to define the bit values b_ℓ for $T_1 \circ T_2$ since they have no relevance to us. The height of the tree T is the maximum number of queries on a root-to-leaf path. An (N, ϕ) RDT (randomized decision tree) is a probability distribution over (N, ϕ) decision trees; its height is the maximum height of a tree in its support. When N, ϕ are clear from context, we simply refer to them as decision trees and RDTs respectively.

For a partial structure ρ and a tree T, we say that ρ "determines a path through T to a leaf" if, starting at the root of T and answering each query according to ρ, no query ever yields an undefined value (and hence we trace a path to some leaf).

Definition 6.2 (Game $\mathcal{G}_\phi(N; m, h, \epsilon)$)**.** For ϕ as above, $N, m, h \in \mathbb{N}$, $\epsilon \in (0, 1]$, we define a game $\mathcal{G}_\phi(N; m, h, \epsilon)$ as follows. The players are called the "inspector" and the "adversary." The game proceeds for m rounds. In round i, the inspector supplies an (N, ϕ) RDT $\mathcal{T}_i$ of height at most h, and the adversary responds with a value $p_i \in [0, 1]$. After all rounds are complete, the adversary supplies a partial $\mathcal{L}$-structure ρ on $[N]$. The adversary wins when all of the following conditions hold:

1. ρ is not ϕ-witnessing.
2. (*Deterministic Consistency*). If $\mathrm{supp}(\mathcal{T}_i) = \{T\}$ for some deterministic tree T, then ρ determines a path in T ending at a leaf $\ell \in T$ with $b_\ell = p_i$; in particular, $p_i \in \{0, 1\}$.
3. (*Mixture Consistency*). For any three round indices i_1, i_2, i_3 such that $\mathcal{T}_{i_1} = \frac{1}{2}\mathcal{T}_{i_2} + \frac{1}{2}\mathcal{T}_{i_3}$, we have $\left|p_{i_1} - \frac{1}{2}p_{i_2} - \frac{1}{2}p_{i_3}\right| \leq \varepsilon$.

We are now ready to state the main theorem of this section:

Theorem 6.3 (Winning strategy)**.** *Suppose there is an infinite $\mathcal{L}$-structure $\mathcal{M}$ satisfying*

$$\mathcal{M} \models \forall v_1 \cdots \forall v_k \, \neg\phi(v_1, \ldots, v_k).$$

Let a be the maximum arity of a function symbol in $\mathcal{L}$ ($a = 0$ if $\mathcal{L}$ only has constant symbols). The adversary has a winning strategy in $\mathcal{G}_\phi(N; m, h, \varepsilon)$ whenever

$$N \;\geq\; 10^5 (a+1)^2 m^2 h^2 \varepsilon^{-4} \big(\ln(2m)\big)^2. \tag{6.1}$$

Let $\mathcal{U}$ be the (infinite) universe of $\mathcal{M}$. Our argument will center around the construction of a certain partial injection $\pi : [N] \to \mathcal{U}$. For any such partial injection we can naturally associate a partial $\mathcal{L}$ structure ρ_π on $[N]$: for a constant c, if $c^{\mathcal{M}} = X$ and $\pi^{-1}(X)$ is defined, we set $c^{\rho_\pi} = \pi^{-1}(X)$. Similarly, if f is a function symbol of arity d, $f^{\mathcal{M}}(\bar{X}) = Y$, and $\pi^{-1}(X_1), \ldots, \pi^{-1}(X_d), \pi^{-1}(Y)$ are all defined, then we set $f^{\rho_\pi}(\pi^{-1}(X_1), \ldots, \pi^{-1}(X_d)) = \pi^{-1}(Y)$. In all other cases ρ_π is undefined. Observe that if $\pi' \supseteq \pi$ then $\rho_{\pi'} \supseteq \rho_\pi$, and for all π, ρ_π is not ϕ-witnessing. We use $|\pi|$ as shorthand for $|\{x \in [N] \mid \pi \text{ is defined at } x\}|$. We will describe a randomized strategy for the adversary; before the i^{th} round, the adversary will have generated some partial injection π_{i-1}, and after seeing the inspector's next move $\mathcal{T}_i$, the adversary will choose an extension $\pi_i \supseteq \pi_{i-1}$ (using randomness) and some answer p_i for that round. At the end, the adversary will supply ρ_{π_m} as the final partial $\mathcal{L}$-structure. We will argue that for any inspector strategy, with positive probability over the internal randomness of our adversary, the resulting play-sequence will be winning for the adversary. Since ρ_{π_m} will always be nonwitnessing, we will only have to confirm the other two win conditions; in fact, once our strategy is defined it will be clear also that condition (2) for

singleton RDTs is handled automatically, so the entire crux of the proof will be in demonstrating that the affine constraints in (3) are satisfied.

We will now define one of the two sources of randomness used by the adversary. Fix an arbitrary set $\mathcal{S} \subseteq \mathcal{U}$ with $|\mathcal{S}| = N$. Define a "priority oracle" ω to consist of a total ordering ω_x on $\mathcal{S}$ for each $x \in [N]$, as well as a total ordering ω_X on $[N]$ for each $X \in \mathcal{U}$. Let Ω be the set of all priority oracles, equipped with the product of the uniform measures on the component orderings; $\omega \sim \Omega$ denotes a sample from this distribution. For a priority oracle ω, a decision tree T of height h, and a partial injection $\pi : [N] \to \mathcal{U}$ satisfying $|\pi| + (a+1)h \leq N$, we define $T^\omega(\pi)$ to be the extension of π obtained by the following traversal. Start at the root of T with $\pi' = \pi$. At each internal node querying $f(x_1, \ldots, x_d)$, perform the following steps, treating a constant query c as a zero-arity query with value $c^{\mathcal{M}}$:

1. For $i = 1, \ldots, d$ in order, if $x_i \notin \operatorname{dom}(\pi')$, extend π' by setting $\pi'(x_i)$ to the ω_{x_i}-least element of $\mathcal{S} \setminus \operatorname{range}(\pi')$.
2. Set $X = f^{\mathcal{M}}(\pi'(x_1), \ldots, \pi'(x_d))$. If $X \notin \operatorname{range}(\pi')$, extend π' by setting $\pi'(y) = X$, where y is the ω_X-least element of $[N] \setminus \operatorname{dom}(\pi')$.
3. Follow the edge in T labeled $(\pi')^{-1}(X)$.

Upon reaching a leaf, set $T^\omega(\pi) = \pi'$. Let $\operatorname{out}^\omega(T, \pi)$ denote the bit labeling the leaf reached in this traversal.

We are now ready to describe the adversary's strategy. At the start, the adversary samples once and for all $\omega \sim \Omega$, and initializes $\pi_0 = \emptyset$. Fix some parameter s depending on N, m, h, ϵ which we will set later. Say that round $i-1$ is complete and the current injection is π_{i-1}. In round i, if the inspector supplies $\mathcal{T}_i$, the adversary uses fresh independent randomness to sample $T_{i,1}, \ldots T_{i,s} \sim \mathcal{T}_i$, sets $\pi_i = (T_{i,1} \circ \cdots \circ T_{i,s})^\omega(\pi_{i-1})$, and sets $p_i = \frac{1}{s} \sum_{j \leq s} \operatorname{out}^\omega(T_{i,j}, \pi_i)$. At the end of the last round m, the adversary outputs ρ_{π_m} as the final partial structure. It is now easy to observe (as promised earlier) that winning condition (2) will always be satisfied for the adversary. Since $\pi_m \supseteq \pi_i$, whenever $\mathcal{T}_i$ is a singleton supported on a fixed tree T_i, we have $p_i = \operatorname{out}^\omega(T_i, \pi_i) = \operatorname{out}^\omega(T_i, \pi_m)$. It remains to argue that with nonzero probability, the final state of the game satisfies condition (3).

Given a transcript of the game (including all the sampled randomness used in the adversary's strategy), we define 3 more sequences of numbers $(p_i^{\text{emp}}, p_i^{\text{true}}, p_i^{\text{final}} \in [0,1])_{i \leq m}$ which will be used in the analysis of the game's outcome. These values are given by:

$$p_i^{\text{emp}} = \frac{1}{s} \sum_{j \leq s} \operatorname{out}^\omega(T_{i,j}, \pi_{i-1}), \quad p_i^{\text{true}} = \mathbb{E}_{T_i \sim \mathcal{T}_i}[\operatorname{out}^\omega(T_i, \pi_{i-1})], \quad p_i^{\text{final}} = \mathbb{E}_{T_i \sim \mathcal{T}_i}[\operatorname{out}^\omega(T_i, \pi_m)].$$

Observe that whenever i_1, i_2, i_3 are such that $\mathcal{T}_{i_1} = \frac{1}{2}\mathcal{T}_{i_2} + \frac{1}{2}\mathcal{T}_{i_3}$, we must have $p_{i_1}^{\text{final}} = \frac{1}{2}p_{i_2}^{\text{final}} + \frac{1}{2}p_{i_3}^{\text{final}}$ exactly. We will argue that with nonzero probability over ω and the additional subsampling randomness of the adversary, the following holds for all i:

$$p_i = p_i^{\text{emp}}, \quad |p_i^{\text{emp}} - p_i^{\text{true}}| \leq \frac{\epsilon}{4}, \quad |p_i^{\text{true}} - p_i^{\text{final}}| \leq \frac{\epsilon}{4}$$

This will imply that for all such i_1, i_2, i_3, $|p_{i_1} - (\frac{1}{2}p_{i_2} + \frac{1}{2}p_{i_3})| \leq \epsilon$ and hence the game state is winning for the adversary (recall that the other two win conditions are satisfied by design).

We need two technical claims before we can proceed with the proof of Theorem 6.3. We first define a "fresh state" which consists of a pair $(\pi, \widetilde{\Omega})$ with $\pi : [N] \to \mathcal{U}$ a partial injection and $\widetilde{\Omega}$ a distribution of priority oracles satisfying the following: all of the orders $\widetilde{\Omega}_x, \widetilde{\Omega}_X$ with $x \in [N], X \in \mathcal{U}$ unmatched in π are jointly independent and uniformly distributed ("unmatched" means $\pi(x)$ is undefined, and $X \notin \operatorname{range}(\pi)$). The first technical claim is the simple observation that, if we start in a fresh state, then run a decision tree and

condition on its outcome, we end up in another fresh state. In other words, the priority orders of unmatched elements remain independent and uniform, since consulting an element's order immediately matches that element, leaving the orders of all remaining unmatched elements untouched.

Observation 6.4. *Let $(\pi, \widetilde{\Omega})$ be a fresh state and let T be a decision tree with $|\pi| + (a+1)\mathrm{height}(T) \leq N$ (so that $T^\omega(\pi)$ is well-defined). Let $\pi' \supseteq \pi$ be such that $\Pr_{\omega\sim\widetilde{\Omega}}[T^\omega(\pi) = \pi'] > 0$, and let Ω' be the distribution of $\omega \sim \widetilde{\Omega}$ conditioned on $T^\omega(\pi) = \pi'$. Then (π', Ω') is a fresh state.*

The main technical lemma underlying the proof of Theorem 6.3:

Lemma 6.5 (Swapping Orders)**.** *Fix a fresh state $(\pi, \widetilde{\Omega})$ and two decision trees T_1, T_2 of heights at most h_1, h_2, respectively. If $|\pi| + (a+1)(h_1+h_2) < N$, then*

$$\Pr_{\omega\sim\widetilde{\Omega}}[(T_1 \circ T_2)^\omega(\pi) \neq (T_2 \circ T_1)^\omega(\pi)] \leq \frac{2(a+1)^2 h_1 h_2}{N - |\pi| - (a+1)(h_1+h_2)}.$$

Proof. We will start by proving the lemma assuming it holds in the special case $h_1 = h_2 = 1$. At the very end we prove it in this special case. Define a "schedule" to be a string $\sigma \in \{1,2\}^{h_1+h_2}$ which has exactly h_1 1's and h_2 2's. We say that σ, σ' are adjacent schedules if there is some index $i < h_1 + h_2$ such that $(\sigma_i, \sigma_{i+1}) = (\sigma'_{i+1}, \sigma'_i)$ and $\sigma_j = \sigma'_j$ for all $j \notin \{i, i+1\}$. For a schedule σ, we define a decision tree T_σ as follows. T_σ starts by performing the first query of T_{σ_1}, and maintains the node reached in T_{σ_1} after that query. It then performs the next query of T_{σ_2} (if $\sigma_1 = \sigma_2$ this is the second step in the same tree, otherwise it is the first step in the other tree) and maintains the node reached in that tree; this continues on for each $i \in [h_1 + h_2]$. If a leaf in one tree T_b is reached before the $h_1 + h_2$ phases are complete, then on all subsequent steps i with $\sigma_i = b$ we perform no further step in round i. We will argue that, whenever σ, σ' are adjacent schedules, we must have

$$\Pr_{\omega\sim\widetilde{\Omega}}[T^\omega_\sigma(\pi) \neq T^\omega_{\sigma'}(\pi)] \leq \frac{2(a+1)^2}{N - |\pi| - (a+1)(h_1+h_2)} \tag{6.2}$$

This will immediately yield the lemma, since $T_1 \circ T_2 = T_{1^{h_1}2^{h_2}}$, $T_2 \circ T_1 = T_{2^{h_2}1^{h_1}}$, and the schedule $1^{h_1}2^{h_2}$ can be transformed into the schedule $2^{h_2}1^{h_1}$ by passing through a sequence of adjacent schedules of length $h_1 h_2 + 1$. We now claim that (6.2) follows from the special case of the current lemma with $h_1 = h_2 = 1$. To see this, say that σ, σ' are adjacent along $i, i+1$. If we can show that after the first $i+1$ steps the partial injections generated by $T^\omega_\sigma(\pi)$ and $T^\omega_{\sigma'}(\pi)$ are identical then it will obviously hold for the remaining steps (corresponding to indices where σ, σ' are equal); we can hence prune the remaining steps, or equivalently, assume $i = h_1 + h_2 - 1$. So assume now $i = h_1 + h_2 - 1$, and let Q be the truncation of T_σ to the first $h_1 + h_2 - 2$ phases of the joint σ-scheduled simulation of T_1, T_2 and let Ω' be the distribution of $\omega \sim \widetilde{\Omega}$ conditioned on some fixing of the value $\pi' = Q^\omega(\pi)$. We see that there is a pair of trees Q_1, Q_2 of height at most one, so that $T^\omega_\sigma(\pi) = (Q_1 \circ Q_2)^\omega(\pi')$ and $T^\omega_{\sigma'}(\pi) = (Q_2 \circ Q_1)^\omega(\pi')$. For this conditioning, we have that (π', Ω') is a fresh state, and hence we may apply the height-one case of the current lemma and conclude

$$\begin{aligned}\Pr_{\omega\sim\Omega'}[T^\omega_\sigma(\pi) \neq T^\omega_{\sigma'}(\pi)] &\leq \Pr_{\omega\sim\Omega'}\left[(Q_1 \circ Q_2)^\omega(\pi') \neq (Q_2 \circ Q_1)^\omega(\pi')\right] \\ &\leq \frac{2(a+1)^2}{N - |\pi'| - 2(a+1)} \leq \frac{2(a+1)^2}{N - |\pi| - (a+1)(h_1+h_2)}\end{aligned}$$

using $|\pi'| \leq |\pi| + (a+1)(h_1 + h_2 - 2)$. Averaging this bound over our conditioning establishes (6.2) and hence finishes the proof of the general case (assuming the special case).

It remains to prove the special case $h_1 = h_2 = 1$. Say T_1, T_2 each have height exactly 1 (the lemma is trivial if one has height zero). Say that T_1 makes a query to $f_1(\bar{x})$ for a function symbol f_1 of arity

d_1, $\bar{x} \in [N]^{d_1}$, and T_2 makes a query to $f_2(\bar{z})$ for a function symbol f_2 of arity d_2, $\bar{z} \in [N]^{d_2}$ (querying constant symbols is treated as the special case of an arity zero function). In the execution of $T_1 \circ T_2$, there are $(d_1+1)+(d_2+1)$ "basic matching operations" that must be performed: first we must pair each of $x_1, \ldots, x_{d_1} \in [N]$ with elements in $\mathcal{S}$, then pair $f_1^{\mathcal{M}}(\pi(x_1), \ldots, \pi(x_{d_1}))$ with an element of $[N]$, and then repeat the same process for f_2. We will apply the same reasoning here as we did in the general case above: if we can analyze the process of swapping two adjacent basic matching operations, and prove that the probability of such a swap starting from a current matching π will change the outcome with probability at most $\frac{2}{N-|\pi|}$, then we can consider a sequence of $(d_1+1)(d_2+1) \leq (a+1)^2$ of these swaps taking us from $T_1 \circ T_2$ to $T_2 \circ T_1$ and apply a union bound. Clearly freshness is preserved by these basic matching operations. At each step, the matching $\pi' \supseteq \pi$ we obtain after applying some basic matching operations satisfies $|\pi'| \leq |\pi| + d_1 + d_2 + 2 \leq |\pi| + 2(a+1)$; so overall we get the desired bound

$$\frac{2(d_1+1)(d_2+1)}{N-|\pi|-2(a+1)} \leq \frac{2(a+1)^2}{N-|\pi|-2(a+1)}$$

on the probability that $(T_1 \circ T_2)^{\omega}(\pi) \neq (T_2 \circ T_1)^{\omega}(\pi)$.

To finish the proof we need to analyze a single swap of two basic matching operations. Say that π is a current matching, and the next two moves demand that we first extend π to pair α, and then further extend it to pair β, for some $\alpha, \beta \in [N] \cup \mathcal{U}$. Obviously we may assume both α, β are currently unpaired in π. If $\alpha, \beta \in [N]$ or $\alpha, \beta \in \mathcal{U}$ and $\alpha \neq \beta$ then the swap will only have consequence if, starting from the initial matching π, both α and β would prefer the same partner γ. This requires that $\omega_\alpha, \omega_\beta$ have the same least element unpaired in π; by the freshness of the state $(\pi, \widetilde{\Omega})$ this happens with probability at most $\frac{1}{N-|\pi|}$. If $\alpha = \beta$ then clearly the swap has no effect. On the other hand, say that $\alpha \in [N], \beta \in \mathcal{U}$; let $Y \in \mathcal{U}, y \in [N]$ be their preferred mates according to ω (computed with respect to π); then a swap will only have an effect when either $\beta = Y$ or $\alpha = y$. Each of these events occurs with probability at most $\frac{1}{N-|\pi|}$, and hence the swap will have an effect with probability at most $\frac{2}{N-|\pi|}$. The case $\alpha \in \mathcal{U}, \beta \in [N]$ is identical. □

At this point we are ready to prove Theorem 6.3.

Proof of Theorem 6.3. As per the discussion above, it suffices to prove that with nonzero probability, the following inequalities hold for all $i \leq m$:

$$(1)\ p_i = p_i^{\text{emp}} \qquad (2)\ |p_i^{\text{emp}} - p_i^{\text{true}}| \leq \frac{\epsilon}{4} \qquad (3)\ |p_i^{\text{true}} - p_i^{\text{final}}| \leq \frac{\epsilon}{4}$$

We will show that for each i, each inequality holds with probability $> 1 - \frac{1}{3m}$ and then take a union bound. For the remainder of the argument, we will fix some particular inspector strategy once and for all. For a single such choice of strategy, we have that $\mathcal{T}_i$ is a deterministic function of ω and the subsampling randomness used through the first $i-1$ rounds.

We start with (2) which is easiest; this does not require Lemma 6.5 and instead follows directly from a concentration bound for an appropriate choice of s. Condition on ω and the choice of subsampling randomness through the first $i-1$ rounds. The summands defining p_i^{emp} are independent Bernoulli variables with mean p_i^{true}; by Hoeffding, we have $\Pr[|p_i^{\text{emp}} - p_i^{\text{true}}| > \frac{\epsilon}{4}] \leq 2e^{-s\epsilon^2/8}$. Setting $s = \lceil 8\epsilon^{-2} \ln(12m) \rceil$, we have $2e^{-s\epsilon^2/8} \leq \frac{1}{6m} < \frac{1}{3m}$. We also note that for this setting of s and our assumption on N, we have $N \geq (a+1)(ms+1)h_{\max}$, where $h_{\max} = (ms+1)h$ upper bounds the maximum height of any decision tree T considered in the remainder of this proof; this will guarantee that for all trees T and matchings π we consider, $T^{\omega}(\pi)$ is well-defined.

We now turn to (1). Condition on π_{i-1} and all subsampling randomness used through round i; we claim that this conditioning amounts to some fresh state $(\pi_{i-1}, \widetilde{\Omega})$. This follows from the fact that the

play of the game across the first $i-1$ rounds (once the subsampling randomness has been fixed) amounts precisely to the execution of some decision tree $Q_{<i}$ of depth $\leq (i-1)hs$ on $\omega \sim \Omega$ starting from π_0[7] using Observation 6.4. It will suffice to show that for each $j \leq s$, with probability $> 1 - \frac{1}{3ms}$ over $\omega \sim \widetilde{\Omega}$, $\mathrm{out}^\omega(T_{i,j}, \pi_{i-1}) = \mathrm{out}^\omega(T_{i,j}, \pi_i)$; taking a union bound over $j \leq s$ will then give (1). We consider the two trees

$$T = T_{i,1} \circ \cdots \circ T_{i,s}, \qquad T' = T_{i,j} \circ T_{i,1} \circ \cdots T_{i,j-1} \circ T_{i,j+1} \circ \cdots T_{i,s}$$

If we can prove that $\Pr_{\omega \sim \widetilde{\Omega}}[T^\omega(\pi_{i-1}) \neq T'^\omega(\pi_{i-1})] < \frac{1}{3ms}$ then we are done. This follows directly from Lemma 6.5; we are simply swapping the order of $T_{i,1} \circ \cdots \circ T_{i,j-1}$ and $T_{i,j}$, and the remaining suffix $T_{i,j+1} \circ \cdots \circ T_{i,s}$ has no effect. We can apply this lemma provided $N > (a+1)msh + 6(a+1)^2 ms(s-1)h^2$ which follows from the Theorem's assumptions and our choice of s.

We now turn to (3); the argument here is similar. Fix all subsampling randomness used throughout all m rounds; we will argue that the bound in (3) holds for every conditioning of the subsampling randomness, and hence on average over it. Now, as in (1) we condition on π_{i-1}; such conditioning again amounts to a fresh state $(\pi_{i-1}, \widetilde{\Omega})$. By the same argument used above to define $Q_{<i}$, we can determine that there is a decision tree $Q_{\geq i}$ (depending on our particular choice of π_{i-1}) of height at most $(m-i+1)sh$ such that $\pi_m = Q^\omega_{\geq i}(\pi_{i-1})$. Observe that $\mathcal{T}_i$ is completely determined (since π_{i-1} has been fixed). Now, our goal is to show that with probability $> 1 - \frac{1}{3m}$ over $\omega \sim \widetilde{\Omega}$, we have:

$$|\mathbb{E}_{T_i \sim \mathcal{T}_i}[\mathrm{out}^\omega(T_i, \pi_{i-1})] - \mathbb{E}_{T_i \sim \mathcal{T}_i}[\mathrm{out}^\omega(T_i, \pi_m)]| \leq \frac{\epsilon}{4}$$

By Markov, to prove this it will suffice to show:

$$\Pr_{\omega \sim \widetilde{\Omega}, T_i \sim \mathcal{T}_i}[\mathrm{out}^\omega(T_i, \pi_{i-1}) \neq \mathrm{out}^\omega(T_i, \pi_m)] < \frac{\epsilon}{12m}$$

where π_m implicitly depends on ω in the above expression. We show the above for every fixed choice of T_i. Applying Lemma 6.5, we know that $(T_i \circ Q_{\geq i})^\omega(\pi_{i-1}) = (Q_{\geq i} \circ T_i)^\omega(\pi_{i-1})$ holds with probability $> 1 - \frac{\epsilon}{12m}$ over $\omega \sim \widetilde{\Omega}$, provided $N > (a+1)(ms+1)h + 24(a+1)^2 m^2 sh^2 \epsilon^{-1}$ which follows from the Theorem's assumptions and our choice of s. This gives the desired result, since if $(T_i \circ Q_{\geq i})^\omega(\pi_{i-1}) = (Q_{\geq i} \circ T_i)^\omega(\pi_{i-1}) = \pi$, then π determines the path traversed in T_i^ω during both executions, and hence this path must agree in both. In the first execution, we run T_i directly on π_{i-1} using ω and hence trace a path to a leaf in T_i labeled $\mathrm{out}^\omega(T_i, \pi_{i-1})$; in the second, we first run $Q_{\geq i}$ on π_{i-1} using ω, which yields π_m, and then trace a path through T_i starting at π_m using ω, which reaches a leaf labeled $\mathrm{out}^\omega(T_i, \pi_m)$. Hence we must have $\mathrm{out}^\omega(T_i, \pi_{i-1}) = \mathrm{out}^\omega(T_i, \pi_m)$. □

6.2 Auxiliary Results for the Reduction to Quantifier-Free Formulas

We first recall some basic definitions. Let $\phi(x_1, \ldots, x_k)$ be a formula over the language $\mathcal{L}$. The *Herbrand normal form* of ϕ is defined as follows. We first convert ϕ into prenex normal form, say

$$\vdash_{\mathrm{FO}} \phi(\overline{x}) \leftrightarrow \forall y_1 \, \exists y_2 \, \ldots \, \forall y_\ell \, \psi(\overline{x}, y_1, \ldots, y_\ell),$$

where ψ is a quantifier-free $\mathcal{L}$-formula. Let $f_1, f_3, \ldots, f_\ell$ be new function symbols. The Herbrand normal form of ϕ, denoted $\mathsf{HNF}(\phi)$, is defined as the formula

$$\exists y_2 \, \exists y_4 \, \ldots \, \exists y_{\ell-1} \, \psi(\overline{x}, f_1(\overline{x}), y_2, f_3(\overline{x}, y_2), \ldots, f_\ell(\overline{x}, y_2, y_4, \ldots, y_{\ell-1}))$$

[7] To define this tree $Q_{<i}$ formally, we start with the first round, in which the inspector supplies some $\mathcal{T}_1$; since the subsampling randomness has been fixed, this completely determines a sequence $T_{1,1}, \ldots, T_{1,s}$ of decision trees. The first part of $Q_{<i}$ will consist of $T_{1,1} \circ \cdots \circ T_{1,s}$; after running these some value p_1 is produced, which then determines $\mathcal{T}_2$, and so on. We halt this process after the first $i-1$ rounds to obtain $Q_{<i}$; the partial matching reached from π_0 using ω is precisely $\pi_{i-1} = Q^\omega_{<i}(\pi_0)$.

in the language $\mathcal{L}_{\mathsf{HNF}} \triangleq \mathcal{L} \cup \{f_1, f_3, \ldots, f_\ell\}$. We will need the following basic properties of first-order logic.

Proposition 6.6. *$\forall \overline{x}\ \neg\phi(\overline{x})$ has a model over the universe $\mathcal{U}$ iff $\forall \overline{x}\ \neg\mathsf{HNF}(\phi)$ has a model over the universe $\mathcal{U}$. In particular, $\forall \overline{x}\ \neg\phi(\overline{x})$ has an infinite model iff $\forall \overline{x}\ \neg\mathsf{HNF}(\phi)$ has an infinite model.*

Proof. We may assume without loss of generality that ϕ is in prenex normal form, as logically equivalent formulas have the same truth value over any model.

The (*if*) direction is straightforward: the model $\mathcal{M}$ for $\forall \overline{x}\ \neg\mathsf{HNF}(\phi)$ is also a model for $\forall \overline{x}\ \neg\phi(\overline{x})$ by removing the interpretations for $f_1, f_3, \ldots, f_\ell$ on the same universe. For the (*only if*) direction, we consider the simplified case where $\phi(\overline{x}) \triangleq \forall y_1\ \exists y_2\ \psi(\overline{x}, y_1, y_2)$, and the general case can be proved analogously. Assume that $\mathcal{M}$ is a model for $\forall \overline{x}\ \neg\phi(\overline{x})$. Then, by definition, for every $\overline{x}^*$ in the universe $\mathcal{U}$, there is an $y_1^* \in \mathcal{U}$ such that for every $y_2^* \in \mathcal{U}$, $\psi^{\mathcal{M}}(\overline{x}^*, y_1^*, y_2^*)$ does not hold. Let f_1^* be any function that fixes some $y_1^* \in \mathcal{U}$ satisfying the above given $\overline{x}^*$. Then, by definition, $\mathcal{M}$ augmented with the interpretation $f_1 \mapsto f_1^*$ is a model for $\forall \overline{x}\ \neg\mathsf{HNF}(\phi)$. This completes the proof. □

Proposition 6.7 ([Koh08, Remark 2.15])**.** $\vdash_{\mathrm{FO}} \phi \to \mathsf{HNF}(\phi)$.

Let $\phi(x_1, \ldots, x_k)$ be any formula in the language $\mathcal{L}$ with equality. Following the notation from Section 1.2.3, we define

$$\Theta_\phi \triangleq \forall N \geq 1\ \left(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N)\ \to\ \exists x_1, \ldots, x_k < N\ \phi^{[N]}(x_1, \ldots, x_k)\right).$$

Proposition 6.8. $\vdash_{\mathrm{FO}} \Theta_\phi \to \Theta_{\mathsf{HNF}(\phi)}$.

Proof. Let $\mathcal{L}_{\mathsf{HNF}} \triangleq \mathcal{L} \cup \{f_1, f_3, \ldots, f_\ell\}$ be the language of $\mathsf{HNF}(\phi)$. Assume, towards a contradiction, that there is a model $\mathcal{M} = (\mathcal{U}, \mathcal{I})$ such that $\mathcal{M} \vDash \Theta_\phi$ but $\mathcal{M} \nvDash \Theta_{\mathsf{HNF}(\phi)}$. Consider any $N^* \in \mathcal{M}$ such that $\mathcal{M} \vDash N^* \geq 1$ and

$$\mathcal{M} \vDash \mathrm{Closed}_{\widehat{\mathcal{L}_{\mathsf{HNF}}}}(N^*)\ \wedge\ \forall x_1, \ldots, x_k < N^*\ \neg\mathsf{HNF}(\phi)^{[N^*]}(x_1, \ldots, x_k).$$

Since $\mathrm{Closed}_{\widehat{\mathcal{L}_{\mathsf{HNF}}}}(N^*)$ implies $\mathrm{Closed}_{\widehat{\mathcal{L}}}(N^*)$, the assumption $\mathcal{M} \vDash \Theta_\phi$ gives witnesses $x_1^*, \ldots, x_k^* <^{\mathcal{I}} N^*$ such that $\mathcal{M} \vDash \phi^{[N^*]}(x_1^*, \ldots, x_k^*)$.

Let $\mathcal{U}_{N^*} \triangleq \{x \in \mathcal{U} \mid x <^{\mathcal{I}} N^*\}$. It can be verified that this set is nonempty: if $k \geq 1$, it contains x_1^*; if $k = 0$, the first Herbrand symbol f_1 is a constant, and expanded closure gives $\widehat{f_1}^{\mathcal{M}}(N^*) <^{\mathcal{I}} N^*$. Let $\mathcal{I}_{N^*} : f(\cdot) \mapsto \widehat{f}^{\mathcal{I}}(N^*, \cdot)$, where constants are treated with functions with no input. By the definition of finitary translation, $\mathcal{M}_{N^*} \triangleq (\mathcal{U}_{N^*}, \mathcal{I}_{N^*})$ satisfies both $\forall \overline{x}\ \neg\mathsf{HNF}(\phi)(\overline{x})$ and $\phi(x_1^*, \ldots, x_k^*)$. This leads to a contradiction with Proposition 6.7. □

Corollary 6.9. *Let $\mathcal{L}_{\mathsf{HNF}} \triangleq \mathcal{L} \cup \{f_1, f_3, \ldots, f_\ell\}$ be the language of $\mathsf{HNF}(\phi)$, and let $\mathcal{T} \in \{\mathsf{PV}_1, \mathsf{APX}_1\}$. If $\mathcal{T}(\widehat{\mathcal{L}}) \vdash \Theta_\phi$, then $\mathcal{T}(\widehat{\mathcal{L}_{\mathsf{HNF}}}) \vdash \Theta_{\mathsf{HNF}(\phi)}$.*

6.3 Unprovability in $\mathsf{APX}_1(\mathcal{L})$

In this section, we complete the proof of Theorem 1.9. Let $\phi(x_1, \ldots, x_k)$ be a formula in the language $\mathcal{L}$ (with equality) of arbitrary quantifier complexity, and write

$$\Theta_\phi \triangleq \forall N \geq 1\ \left(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N)\ \to\ \exists x_1, \ldots, x_k < N\ \phi^{[N]}(x_1, \ldots, x_k)\right).$$

Assume that $\forall x_1 \cdots \forall x_k\ \neg\phi(x_1, \ldots, x_k)$ has an infinite model.

Lemma 6.10. *Assume that Theorem 1.9 holds for all quantifier-free formulas ϕ. Then Theorem 1.9 holds.*

Proof. Let $\mathsf{HNF}(\phi) \triangleq \exists \overline{y}\ \beta(\overline{x}, \overline{y})$ be the Herbrand normal form of ϕ. Assume that $\mathsf{APX}_1(\widehat{\mathcal{L}}) \vdash \Theta_\phi$. We know by Corollary 6.9 that $\mathsf{APX}_1(\widehat{\mathcal{L}_{\mathsf{HNF}}}) \vdash \Theta_{\mathsf{HNF}(\phi)}$. By the definition of the finitary translation, we can see that $\Theta_{\mathsf{HNF}(\phi)}$ and Θ_β are logically equivalent, and thus $\mathsf{APX}_1(\widehat{\mathcal{L}_{\mathsf{HNF}}}) \vdash \Theta_\beta$.

Since by assumption Theorem 1.9 holds for any quantifier-free formula, applying it to the formula β shows that $\forall \overline{x}\ \forall \overline{y}\ \neg\beta(\overline{x}, \overline{y})$ has no infinite model, or equivalently, $\forall \overline{x}\ \neg\mathsf{HNF}(\phi)$ has no infinite model. By Proposition 6.6, $\forall \overline{x}\ \neg\phi$ also has no infinite model. □

In the rest of the proof, we consider the case where ϕ is a quantifier-free formula. We prove Theorem 1.9 (for quantifier-free ϕ) by showing that a proof of Θ_ϕ in $\mathsf{APX}_1(\widehat{\mathcal{L}})$ yields a *winning inspector*. In more detail, we call a play of the game $\mathcal{G}_\phi$ (Definition 6.2) *consistent* if it satisfies the adversary's conditions (2) and (3), i.e., *deterministic consistency* and *mixture consistency*, respectively. Our inspector forces every consistent play to end with a partial $\mathcal{L}$-structure witnessing ϕ, thereby violating condition (1). Since this is in contradiction with Theorem 6.3, it follows that $\mathsf{APX}_1(\widehat{\mathcal{L}}) \nvdash \Theta_\phi$, as desired.

Formally, we will prove the following result.

Lemma 6.11 (Proofs yield winning inspectors)**.** *Let ϕ be a quantifier-free formula. If $\mathsf{APX}_1(\widehat{\mathcal{L}}) \vdash \Theta_\phi$, there is a polynomial $Q(n)$ depending only on the proof such that the inspector wins*

$$\mathcal{G}_\phi\big(2^n; Q(n), Q(n), 1/Q(n)\big)$$

for every $n \geq 1$.

First, we derive Theorem 1.9 from the lemma.

Proof of Theorem 1.9. By Lemma 6.10, we may assume that ϕ is quantifier-free. Suppose $\mathsf{APX}_1(\widehat{\mathcal{L}}) \vdash \Theta_\phi$, and take Q from Lemma 6.11. Let a be the maximum arity in the original language $\mathcal{L}$, with $a = 0$ if it contains only constants. Choose n sufficiently large such that

$$2^n \geq 10^5 (a+1)^2 Q(n)^8 \big(\ln(2Q(n))\big)^2.$$

The infinite model and Theorem 6.3 give a winning adversary in $\mathcal{G}_\phi(2^n; Q(n), Q(n), 1/Q(n))$, contradicting the lemma. This shows that $\mathsf{APX}_1(\widehat{\mathcal{L}}) \nvdash \Theta_\phi$. □

The remainder of this section is dedicated to the proof of the lemma.

Proof of Lemma 6.11. Roughly speaking, we use Herbrand's theorem to construct an inspector from a proof of Θ_ϕ in $\mathsf{APX}_1(\widehat{\mathcal{L}})$ and prove that the inspector wins the game. More details follow:

Step I: Applying Herbrand's theorem. Consider the theory $\mathsf{APX}_1(\widehat{\mathcal{L}})$ augmented with a fresh constant ν and the following axioms:

$$\nu \geq 1 \qquad \text{and} \qquad \mathrm{Closed}_{\widehat{\mathcal{L}}}(\nu).$$

Intuitively, the constant ν denotes the model size $N = 2^n$ we are working with. Since $\mathsf{APX}_1(\widehat{\mathcal{L}})$ proves Θ_ϕ, the augmented theory proves $\exists \bar{x}\, W(\nu, \bar{x})$ where

$$W(N, \bar{x}) \triangleq \left(\bigwedge_{\ell=1}^{k} x_\ell < N \right) \wedge \phi^{[N]}(\bar{x}).$$

Since the augmented theory is a universal theory, by Herbrand's theorem (Theorem 2.1), the following *finite* implication holds:

$$\vdash_{\mathrm{FO}} \bigwedge_{j=1}^{m} \theta_j(\bar{s}_j(\nu)) \longrightarrow \bigvee_{i=1}^{r} W(\nu, \bar{t}_i(\nu)). \tag{6.3}$$

where each θ_j is a ground instance of a universal axiom $\forall \bar{u}_j\, \theta_j(\bar{u}_j)$ of the augmented theory (with $\theta_j(\bar{u}_j)$ quantifier-free), and $\bar{s}_j$ and $\bar{t}_i$ are tuples of ground terms in its language.

Step II: From the finite implication to an inspector. We define an inspector that evaluates every antecedent $\theta_j(\bar{s}_j(\nu))$ and every succedent $W(\nu, \bar{t}_i(\nu))$ in (6.3). Given a total $\mathcal{L}$-structure ρ on $[N]$, define its interpretation $\widehat{\rho}$ by

$$\widehat{f^{\rho}}(s, \bar{u}) = \begin{cases} f^{\rho}(\bar{u}), & s = N \text{ and } \bar{u} \in [N]^a, \\ 0, & \text{otherwise}, \end{cases} \tag{6.4}$$

for each function f with arity a (constants have arity $a = 0$). Then the inspector evaluates every term and atomic subformula occurring in the antecedents $\theta_j(\bar{s}_j(\nu))$ and succedents $W(\nu, \bar{t}_i(\nu))$ in (6.3) on the interpretation $\widehat{\rho}$. During this procedure, the inspector needs to evaluate $\mathcal{L}$-functions and P-oracles. Whenever this occurs, we translate them into (randomized) decision trees and submit them as queries to the adversary.

In more detail, fix a large enough polynomial $P(n)$ that upper bounds the running time of each $\theta_j(\bar{s}_j(\nu))$ and $W(\nu, \bar{t}_i(\nu))$. Note that $P(n)$ also upper bounds their query complexity, the precision parameters $\delta^{-1} = \|\Delta\|$ of each invocation of P, and the inverse of slack parameters β^{-1} in each approximate counting axioms mentioned in (6.3). We set $Q(n) := (m + r) \cdot \lceil \log N \rceil \cdot P(n) \leq \mathrm{poly}(n)$ and consider the game $\mathcal{G}_{\phi}(N; Q(n), Q(n), 1/Q(n))$.

- **Case I: evaluating a function f.** A query to $f(\bar{u})$ is translated into $\lceil \log N \rceil$ queries to the adversary, each corresponding to one output bit of $f(\bar{u})$. That is, for each output bit $i \in [\lceil \log N \rceil]$, let B_i denote the decision tree that outputs the i-th bit of $f(\bar{u})$; the inspector submits the singleton distribution concentrated at B_i. Denote the adversary's response by p_i. If one of the numbers p_i is non-Boolean, then the *(Deterministic Consistency)* of the adversary already fails and the inspector wins the game. Otherwise, we have $f(\bar{u}) = \sum_i 2^i p_i$.

- **Case II: evaluating a P oracle.** A query to $\mathsf{P}(C, \Delta)$, where C is a valid circuit and $1 \leq \|\Delta\| \leq P(n)$, is translated into a randomized decision tree query to the adversary. For each input z of C (recall that z is treated as "randomness"), the computation of $C(z)$ can be expressed as a decision tree that queries $\mathcal{L}$-functions; we denote this decision tree as $T_{C,z}$. Let $\mathcal{T}_C$ denote the randomized decision tree that is the uniform distribution of $T_{C,z}$ (over z). The inspector submits $\mathcal{T}_C$ as a randomized decision tree to the adversary and obtains an answer $p_C \in [0, 1]$. We round p_C to the closest integer multiple of $2^{-\lceil \log \|\Delta\| \rceil}$ as the answer to this query.

  Note that if C does not read its input (hence $\mathcal{T}_C$ can be written as a singleton distribution) and p_C is non-Boolean, then the *(Deterministic Consistency)* of the adversary fails and the inspector wins the game. Also, the inspector stores a cache of already returned answers p_C so that if the same circuit C occurs later, the inspector does not issue any query and instead returns (the rounding of) p_C as stored.

The game proceeds in at most $(m + r) \cdot \log N \cdot P(n) \leq Q(n)$ rounds and each randomized decision tree submitted by the inspector has height at most $P(n) \leq Q(n)$.

Step III: Proving the inspector wins. We now show that the inspector wins the game; this is an immediate corollary of (6.3).

Start with the standard arithmetic structure on binary strings. Recall that at the end of the game, the adversary returns a partial $\mathcal{L}$-structure ρ on $[N]$. We extend ρ to a total structure ρ' by arbitrarily assigning unassigned constants and function values in ρ. Every $\widehat{\mathcal{L}}$-function symbol $\widehat{f}$ is interpreted as the function $\widehat{f}^{\rho'}$ as in (6.4). The symbol ν is interpreted as N. We also interpret $\mathsf{P}(C, \Delta)$ as p_C, rounded to the nearest integer multiple of $2^{-\lceil \log \|\Delta\| \rceil}$, whenever the inspector queried $P(C, \Delta)$ during the game, and 0 otherwise. Finally, we interpret every $\mathsf{PV}_1(\widehat{\mathcal{L}}, \mathsf{P})$-function by its defining oracle program. Since (6.3) holds in this structure, one of the following is true:

- One of the succedents $W(\nu, \bar{t}_i(\nu))$ is true. We show that this implies ρ is ϕ-witnessing. Indeed, let $\bar{x}$ be the value of $\bar{t}_i(N)$ in the completed arithmetic structure, then $\bar{x} < N$ and $\phi^{[N]}(\bar{x})$ holds there. Every term of $\phi^{[N]}(\bar{x})$ was evaluated, and one can use induction to show that $\rho \models \phi^{[N]}[\bar{x}]$. It follows that ρ is ϕ-witnessing and the inspector wins the game.
- One of the antecedents $\theta_j(\bar{s}_j(\nu))$ is false. Among all axioms of $\mathsf{APX}_1(\widehat{L})$, the axioms $\nu \geq 1$, $\mathrm{Closed}_{\widehat{\mathcal{L}}}(\nu)$, and axioms of $\mathsf{PV}_1(\widehat{L}, \mathsf{P})$ are clearly true under our interpretation. Thus it suffices to consider the case that $\theta_j(\bar{s}_j(\nu))$ is an approximate counting axiom.

  1. *Basic Axiom Instances.* Since the adversary always returns a number in $[0, 1]$, the basic axiom instance $\mathsf{P}_\delta(C) \in [0, 1] \cap \mathbb{Q}$ always holds.
  2. *Boundary Axiom Instances.* It is easy to see that either the adversary violates the *(Deterministic Consistency)* or any Boundary Axiom instance appearing as some $\theta_j(\bar{s}_j(\nu))$ is satisfied.
  3. *Precision Consistency Axiom Instances.* Let C be a circuit and $\delta_1^{-1}, \delta_2^{-1}, \beta^{-1} \in \mathsf{Log}$ such that the axiom $|\mathsf{P}_{\delta_1}(C) - \mathsf{P}_{\delta_2}(C)| \leq \delta_1 + \delta_2 + \beta$ appears as some $\theta_j(\bar{s}_j(\nu))$. Since both $\mathsf{P}_{\delta_1}(C)$ and $\mathsf{P}_{\delta_2}(C)$ are the rounding of the same stored response p_C, and the roundings have errors at most δ_1 and δ_2 respectively, the axiom is satisfied.
  4. *Local Consistency Axiom Instances.* Finally, it suffices to show that if C is a circuit, $\delta^{-1}, \beta^{-1} \in \mathsf{Log}$, and the Local Consistency Axiom

$$\left| \mathsf{P}_\delta(C) - \frac{\mathsf{P}_\delta(\mathrm{Fix}_0(C)) + \mathsf{P}_\delta(\mathrm{Fix}_1(C))}{2} \right| \leq 2\delta + \beta \tag{6.5}$$

  appeared as some $\theta_j(\bar{s}_j(\nu))$ and is falsified, then the adversary violates *(Mixture Consistency)* and loses the game. Note that $\mathcal{T}_C = \frac{1}{2}\mathcal{T}_{\mathrm{Fix}_0(C)} + \frac{1}{2}\mathcal{T}_{\mathrm{Fix}_1(C)}$. However, if (6.5) is false, then

$$\left| p_C - \frac{p_{\mathrm{Fix}_0(C)} + p_{\mathrm{Fix}_1(C)}}{2} \right| > \beta \geq 1/Q(n),$$

  meaning the adversary violates *(Mixture Consistency)*. □

6.4 Finite Combinatorial Principles Provable in APX_1

In this section, we provide the proof of Corollary 1.12. We will use the following result.

Proposition 6.12. *If* $\vdash_{\mathrm{FO}} \exists \overline{x}\, \phi(\overline{x})$, *then* $\vdash_{\mathrm{FO}} (\forall N \geq 1\ \exists x\ x < N) \to \Theta_\phi$.

Proof. Suppose for a contradiction that there is a model $\mathcal{M} = (\mathcal{U}, \mathcal{I})$ such that $\mathcal{M} \vDash \forall N \geq 1\ \exists x\ x < N$ but $\mathcal{M} \nvDash \Theta_\phi$. Let $N^* \in \mathcal{M}$ be an element such that $\mathcal{M} \vDash N^* \geq 1$, and

$$\mathcal{M} \vDash \left(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N^*) \ \wedge\ \forall x_1, \ldots, x_k < N^*\ \neg \phi^{[N^*]}(x_1, \ldots, x_k) \right).$$

Let $\mathcal{U}_{N^*} \triangleq \{x \in \mathcal{M} \mid x < N^*\}$, and let $\mathcal{I}_{N^*} \colon f(\cdot) \mapsto \mathcal{I}(\widehat{f}(N^*, \cdot))$. Since $\mathcal{M} \vDash \forall N \geq 1\ \exists x\ x < N$, we know that $\mathcal{U}_{N^*}$ is non-empty. By the definition of the finitary translation, we know that $\mathcal{M}_{N^*} \triangleq (\mathcal{U}_{N^*}, \mathcal{I}_{N^*})$ is a model of $\forall x_1, \ldots, x_k\ \neg\phi$. This leads to a contradiction with the assumption that $\vdash_{\mathrm{FO}} \exists \overline{x}\, \phi(\overline{x})$. □

Corollary 1.12 (Finite relativized combinatorial principles provable in APX_1)**.** *Let $\phi(x_1, \ldots, x_k)$ be any formula in $\mathcal{L}$ with equality, and suppose that $\exists \bar{x}\, \phi(\bar{x})$ holds in every non-empty finite $\mathcal{L}$-structure. Write*

$$\Theta_\phi \triangleq \forall N \geq 1 \left(\mathrm{Closed}_{\widehat{\mathcal{L}}}(N) \;\rightarrow\; \exists x_1, \ldots, x_k < N \; \phi^{[N]}(x_1, \ldots, x_k) \right).$$

Then the following are equivalent:

1. $\mathsf{APX}_1(\widehat{\mathcal{L}}) \vdash \Theta_\phi$.
2. $\mathsf{PV}_1(\widehat{\mathcal{L}}) \vdash \Theta_\phi$.
3. *$\exists \bar{x}\, \phi(\bar{x})$ holds in every infinite $\mathcal{L}$-structure.*

Proof. The implication $(2) \Rightarrow (1)$ is immediate, since $\mathsf{APX}_1(\widehat{\mathcal{L}})$ extends $\mathsf{PV}_1(\widehat{\mathcal{L}})$. For $(1) \Rightarrow (3)$, if (3) fails, then $\forall \bar{x}\, \neg\phi(\bar{x})$ has an infinite model. Theorem 1.9 therefore gives $\mathsf{APX}_1(\widehat{\mathcal{L}}) \nvdash \Theta_\phi$, contradicting (1).

It remains to prove $(3) \Rightarrow (2)$. Assume (3). Together with the hypothesis concerning finite structures, this implies that $\vdash_{\mathrm{FO}} \exists \bar{x}\, \phi(\bar{x})$. By Proposition 6.12, $\vdash_{\mathrm{FO}} (\forall N \geq 1\, \exists x\, x < N) \rightarrow \Theta_\phi$. Then the provability in $\mathsf{PV}_1(\widehat{\mathcal{L}})$ follows from the simple fact that $\mathsf{PV}_1 \vdash \forall N \geq 1\, \exists x\, x < N$. □

References

[ABM25] Albert Atserias, Sam Buss, and Moritz Müller. On the consistency of circuit lower bounds for non-deterministic time. *J. Math. Log.*, 25(3):2450023:1–2450023:41, 2025.

[AM15] Albert Atserias and Moritz Müller. Partially definable forcing and bounded arithmetic. *Arch. Math. Log.*, 54(1-2):1–33, 2015.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[AS16] Noga Alon and Joel H. Spencer. *The probabilistic method*. John Wiley & Sons, 2016.

[AT14] Albert Atserias and Neil Thapen. The ordering principle in a fragment of approximate counting. *ACM Trans. Comput. Log.*, 15(4):29:1–29:11, 2014.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993.

[Bus98] Samuel R. Buss. An introduction to proof theory. In *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 1–78. Elsevier, 1998.

[CJLR26] Liyan Chen, Abhishek Jain, Jiatu Li, and Lawrence Roy. Cryptography from dream compression. Manuscript, 2026.

[CL24] Yilei Chen and Jiatu Li. Hardness of range avoidance and remote point for restricted circuits via cryptography. In *Symposium on Theory of Computing* (STOC), pages 620–629, 2024.

[CLO24] Lijie Chen, Jiatu Li, and Igor C. Oliveira. Reverse mathematics of complexity lower bounds. In *Symposium on Foundations of Computer Science* (FOCS), pages 505–527, 2024.

[CLOW26] Lijie Chen, Jiatu Li, Igor C. Oliveira, and Ryan Williams. A theory for probabilistic polynomial-time reasoning. In *Symposium on Theory of Computing* (STOC), pages 1326–1337, 2026.

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

[dRGR22] Susanna F. de Rezende, Mika Göös, and Robert Robere. Guest column: Proofs, circuits, and communication. *SIGACT News*, 53(1):59–82, 2022.

[FGJ+26] Noah Fleming, Stefan Grosser, Siddhartha Jain, Jiawei Li, Hanlin Ren, Morgan Shirley, and Weiqiang Yuan. Total search problems in ZPP. In *Innovations in Theoretical Computer Science Conference* (ITCS), volume 362 of *LIPIcs*, pages 60:1–60:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026.

[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.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

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

[HJLM26a] Yao-Ching Hsieh, Abhishek Jain, Jiatu Li, and Surya Mathialagan. Overcoming padding in cryptography via the hardness of certifying random strings. In *Symposium on Foundations of Computer Science* (FOCS), 2026.

[HJLM26b] Yao-Ching Hsieh, Abhishek Jain, Jiatu Li, and Surya Mathialagan. SNARGs for NP from unprovability of mathematical theorems (Or: How to use the simplicity of cryptographic reasoning). In *Symposium on Theory of Computing* (STOC), pages 31–41, 2026.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[HV26] Edward A. Hirsch and Ilya Volkovich. Upper and lower bounds for the linear ordering principle. In *Symposium on Theoretical Aspects of Computer Science* (STACS), volume 364, pages 52:1–52:19, 2026.

[Ila25a] Rahul Ilango. Gödel in cryptography: Effectively zero-knowledge proofs for NP with no interaction, no setup, and perfect soundness. In *Symposium on Foundations of Computer Science* (FOCS), pages 1100–1127, 2025.

[Ila25b] Rahul Ilango. The oracle derandomization hypothesis is false (and more) assuming no natural proofs. *Electron. Colloquium Comput. Complex.*, 2025.

[ILW23] Rahul Ilango, Jiatu Li, and Ryan Williams. Indistinguishability obfuscation, range avoidance, and bounded arithmetic. In *Symposium on Theory of Computing* (STOC), pages 1076–1089. ACM, 2023.

[Jeř04] Emil Jeřábek. Dual weak pigeonhole principle, Boolean complexity, and derandomization. *Annals of Pure and Applied Logic*, 129(1-3):1–37, 2004.

[Jeř07a] Emil Jeřábek. Approximate counting in bounded arithmetic. *Journal of Symbolic Logic*, 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.

[Jeř16] Emil Jeřábek. Integer factoring and modular square roots. *Journal of Computer and System Sciences*, 82(2):380–394, 2016.

[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.

[Kor22] Oliver Korten. Derandomization from time-space tradeoffs. In *Computational Complexity Conference* (CCC), pages 37:1–37:26, 2022.

[Kor25] Oliver Korten. Range avoidance and the complexity of explicit constructions. *Bull. EATCS*, 145, 2025.

[KP24] Oliver Korten and Toniann Pitassi. Strong vs. weak range avoidance and the linear ordering principle. In *Symposium on Foundations of Computer Science* (FOCS), pages 1388–1407, 2024.

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

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

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

[LC12] Dai Tri Man Lê and Stephen A. Cook. Formalizing randomized matching algorithms. *Logical Methods in Computer Science*, 8(3), 2012. Article 5, 25 pages.

[Li25] Jiatu Li. An introduction to feasible mathematics and bounded arithmetic for computer scientists. *Electron. Colloquium Comput. Complex.*, TR25-086, 2025.

[LLR26] Jiawei Li, Yuhao Li, and Hanlin Ren. Finding bugs in short proofs: The metamathematics of resolution lower bounds. In *Symposium on Theory of Computing* (STOC), pages 788–798, 2026.

[LO23] Jiatu Li and Igor C. Oliveira. Unprovability of strong complexity lower bounds in bounded arithmetic. In *Symposium on Theory of Computing* (STOC), pages 1051–1057, 2023.

[LPT24] Jiatu Li, Edward Pyne, and Roei Tell. Distinguishing, predicting, and certifying: On the long reach of partial notions of pseudorandomness. In *Symposium on Foundations of Computer Science* (FOCS), pages 1500–1512, 2024.

[LRZ26] Xin Li, Hanlin Ren, and Yan Zhong. Many proof complexity generators inside one demi-bits generator. In *Symposium on Foundations of Computer Science* (FOCS), 2026.

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

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Mül21] Moritz Müller. Typical forcings, NP search problems and an extension of a theorem of Riis. *Ann. Pure Appl. Log.*, 172(4):102930, 2021.

[Oli25] Igor C. Oliveira. Meta-mathematics of computational complexity theory. *SIGACT News*, 56(1):41–68, 2025.

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

[Pic15] Ján Pich. Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic. *Logical Methods in Computer Science*, 11(2), 2015.

[PS26] Ján Pich and Rahul Santhanam. Learning algorithms versus automatability of Frege systems. *J. Math. Log.*, 26(2):2550002:1–2550002:35, 2026.

[Rii94] Søren Riis. Finitisation in bounded arithmetic. BRICS Report Series RS-94-23, Department of Computer Science, University of Aarhus, 1994.

[RWZ26] Hanlin Ren, Yichuan Wang, and Yan Zhong. Hardness of range avoidance and proof complexity generators from demi-bits. In *Innovations in Theoretical Computer Science Conference* (ITCS), volume 362, pages 111:1–111:25, 2026.

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Tha02] Neil Thapen. *The weak pigeonhole principle in models of bounded arithmetic*. PhD thesis, University of Oxford, 2002.

[Tha05] Neil Thapen. Structures interpretable in models of bounded arithmetic. *Ann. Pure Appl. Log.*, 136(3):247–266, 2005.

[Tha24] Neil Thapen. How to fit large complexity classes into TFNP. *CoRR*, abs/2412.09984, 2024.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *Symposium on Foundations of Computer Science* (FOCS), pages 80–91, 1982.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation* (EUROSAM), volume 72 of *Lecture Notes in Computer Science*, page 216–226. Springer, 1979.

A Auxiliary Results and Missing Formalizations

A.1 Standard Models and Admissible Models

Proposition 3.4. *Let* $\hat{\mathsf{P}} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ *be any function. Then* $\mathbb{M}(\hat{\mathsf{P}})$ *is a standard model of* APX_1^* *if and only if it is an admissible model of* APX_1^*.

Proof. ($\Rightarrow$): it can be verified by straightforward calculation. ($\Leftarrow$): Suppose that $\mathbb{M}(\hat{\mathsf{P}})$ is admissible. The BASIC AXIOM FOR APX_1^* and the length bound imposed on P in the language $\mathsf{PV}_1(\mathsf{P})$ ensure condition 1 of Definition 3.2 except for the requirement $q \in (1 \pm 1/\|\Delta\|) \cdot p(C)$, and the BOUNDARY AXIOM FOR APX_1^* ensures condition 2. Suppose, towards a contradiction, that there are a circuit $C : \{0,1\}^n \to \{0,1\}$ and a string Δ such that $q \triangleq \hat{\mathsf{P}}(C,\Delta) \notin (1 \pm 1/\|\Delta\|) \cdot p(C)$, where $p(C)$ denotes the acceptance probability of C. Note that both C and Δ are fixed finite strings and $n \in \mathbb{N}$ is a standard integer. Then there are two cases:

(Case 1: $q > (1 + 1/\|\Delta\|) \cdot p(C)$). Let $\varepsilon \in (0,1)$ such that $q > (1+\varepsilon)(1 + 1/\|\Delta\|) \cdot p(C)$. Let $\Xi \triangleq 1^{64(n+1)/\varepsilon}$ and $\xi \triangleq 1/\|\Xi\| = \varepsilon/(64(n+1))$, and instantiate the notation $\lesssim^*$ in every axiom below at the slack $\beta \triangleq \xi$. By the PRECISION CONSISTENCY AXIOM FOR APX_1^* (with $\delta_1 \triangleq \xi$ and $\delta_2 \triangleq 1/\|\Delta\|$),

$$\hat{\mathsf{P}}(C,\Xi) \;\geq\; \frac{(1-\xi)\,q}{(1+\xi)(1+1/\|\Delta\|)} \;>\; (1+\varepsilon)\cdot\frac{1-\xi}{1+\xi}\cdot p(C) \;\geq\; (1+\varepsilon)(1-2\xi)\cdot p(C).$$

We construct a sequence of circuits $C^{(0)} \triangleq C$ and $C^{(i+1)} \triangleq \mathrm{Fix}_{\sigma_i}(C^{(i)})$ maintaining the invariant

$$\hat{\mathsf{P}}(C^{(i)},\Xi) \;>\; t_i \cdot p(C^{(i)}), \qquad t_0 \triangleq (1+\varepsilon)(1-2\xi), \quad t_{i+1} \triangleq (1-3\xi)\cdot t_i;$$

note that when $p(C^{(i)}) = 0$ the invariant asserts $\hat{\mathsf{P}}(C^{(i)},\Xi) > 0$. For the inductive step, the LOCAL CONSISTENCY AXIOM FOR APX_1^* at precision ξ gives

$$\frac{\hat{\mathsf{P}}(\mathrm{Fix}_0(C^{(i)}),\Xi) + \hat{\mathsf{P}}(\mathrm{Fix}_1(C^{(i)}),\Xi)}{2} \;\geq\; \frac{1-\xi}{(1+\xi)^2}\cdot\hat{\mathsf{P}}(C^{(i)},\Xi) \;>\; t_{i+1}\cdot\frac{p(\mathrm{Fix}_0(C^{(i)})) + p(\mathrm{Fix}_1(C^{(i)}))}{2},$$

where the last inequality follows from $(1-3\xi)(1+\xi)^2 \le 1-\xi$ and $p(C^{(i)}) = (p(\mathrm{Fix}_0(C^{(i)})) + p(\mathrm{Fix}_1(C^{(i)})))/2$. By averaging, there is $\sigma_i \in \{0,1\}$ with $\hat{\mathsf{P}}(\mathrm{Fix}_{\sigma_i}(C^{(i)}),\Xi) > t_{i+1}\cdot p(\mathrm{Fix}_{\sigma_i}(C^{(i)}))$. After n steps, the circuit $C^{(n)}$ does not read any input bit, and

$$t_n \ge (1+\varepsilon)(1-3\xi)^{n+1} \ge (1+\varepsilon)\big(1-3(n+1)\xi\big) = (1+\varepsilon)(1-3\varepsilon/64) > 1.$$

By the BOUNDARY AXIOM FOR APX_1^*, $\hat{\mathsf{P}}(C^{(n)},\Xi) = p(C^{(n)}) \in \{0,1\}$, while both cases contradict $\hat{\mathsf{P}}(C^{(n)},\Xi) > p(C^{(n)})$.

(*Case 2:* $q < (1-1/\|\Delta\|)\cdot p(C)$). This case is symmetric and essentially similar to Case 1. Specifically, we construct $C^{(i)}$ for $i = 0,1,\ldots,n$ such that $\hat{\mathsf{P}}(C^{(i)},\Xi) < s_i \cdot p(C^{(i)})$; we set Ξ to be sufficiently large so $s_n < 1$ and it leads to a contradiction. □

A.2 Basic Counting Functionalities in APX_1^*

In this subsection, we prove the lemmas deferred from Section 5.3.1.

Lemma 5.7 (Global Consistency for APX_1^*)**.** APX_1^* *proves the following statement. Let* $n, \delta^{-1} \in \mathsf{Log}$ *and* $C_1, C_2 : \{0,1\}^n \to \{0,1\}$ *be circuits such that* $C_1(x) \le C_2(x)$ *for every* $x \in \{0,1\}^n$*. Then*

$$(1-\delta)\cdot \mathsf{P}_\delta(C_1) \lesssim^* (1+\delta)\cdot \mathsf{P}_\delta(C_2).$$

In particular, if $C_1(x) = C_2(x)$ *for every* $x \in \{0,1\}^n$*, then both* $(1-\delta)\cdot \mathsf{P}_\delta(C_1) \lesssim^* (1+\delta)\cdot \mathsf{P}_\delta(C_2)$ *and* $(1-\delta)\cdot \mathsf{P}_\delta(C_2) \lesssim^* (1+\delta)\cdot \mathsf{P}_\delta(C_1)$.

Proof. We follow the bit-by-bit fixing trick as in [CLOW26], with changes to take care of the multiplicative error in approximate counting. We argue in APX_1^*. Assume, towards a contradiction, that there exist $C_1, C_2 : \{0,1\}^n \to \{0,1\}$ such that $C_1(x) \le C_2(x)$ for every $x \in \{0,1\}^n$, and for some $\beta^{-1} \in \mathsf{Log}$,

$$(1-\delta)\cdot \mathsf{P}_\delta(C_1) > (1+\beta)\cdot(1+\delta)\cdot \mathsf{P}_\delta(C_2). \tag{A.1}$$

We may assume that $\delta < 1$ as the other case is trivial. Let $\eta^{-1} \in \mathsf{Log}$ be determined later (satisfying $\eta \le 1/16$) and let $\kappa \triangleq (1+\eta)^2/(1-\eta)$. (Note that every occurrence of $\lesssim^*$ in the axioms below is instantiated at the slack η.)

By the PRECISION CONSISTENCY AXIOM FOR APX_1^*, we have

$$\mathsf{P}_\delta(C_1) \le \frac{(1+\eta)(1+\delta)}{1-\eta}\cdot \mathsf{P}_\eta(C_1) \le \kappa\cdot(1+\delta)\cdot \mathsf{P}_\eta(C_1),$$
$$\mathsf{P}_\delta(C_2) \ge \frac{1-\delta}{(1+\eta)^2}\cdot \mathsf{P}_\eta(C_2) \ge \kappa^{-1}\cdot(1-\delta)\cdot \mathsf{P}_\eta(C_2),$$

where the last inequality in each line follows from $(1+\eta)/(1-\eta) \le \kappa$ and $(1+\eta)^2 \le \kappa$. By combining these two inequalities with (A.1), we have

$$\kappa^2\cdot \mathsf{P}_\eta(C_1) > (1+\beta)\cdot \mathsf{P}_\eta(C_2). \tag{A.2}$$

Let $\mathrm{Fix}_b(C)$ be the circuit obtained by fixing the rightmost input bit of C to b. Similarly, we define $\mathrm{Fix}_z(C)$ as the circuit that fixes the last $\|z\|$ input bits of C to z. By the LOCAL CONSISTENCY AXIOM FOR APX_1^*, for every circuit D with at least one input bit, we have

$$\kappa^{-1}\cdot\frac{\mathsf{P}_\eta(\mathrm{Fix}_0(D)) + \mathsf{P}_\eta(\mathrm{Fix}_1(D))}{2} \le \mathsf{P}_\eta(D) \le \kappa\cdot\frac{\mathsf{P}_\eta(\mathrm{Fix}_0(D)) + \mathsf{P}_\eta(\mathrm{Fix}_1(D))}{2}. \tag{A.3}$$

Consider the algorithm that iteratively fixes the rightmost input bit of both circuits: starting from $D_1^{(0)} \triangleq C_1$ and $D_2^{(0)} \triangleq C_2$, in round $j = 0, 1, \ldots, n-1$ it computes a bit b_j (specified below) and sets $D_i^{(j+1)} \triangleq \mathrm{Fix}_{b_j}(D_i^{(j)})$ for $i \in \{1, 2\}$. The algorithm maintains the invariant

$$\kappa^{2(j+1)} \cdot \mathsf{P}_\eta(D_1^{(j)}) > (1+\beta) \cdot \mathsf{P}_\eta(D_2^{(j)}), \tag{A.4}$$

which holds for $j = 0$ by (A.2). Suppose that it holds for some $j < n$, and let $S_i \triangleq \mathsf{P}_\eta(\mathrm{Fix}_0(D_i^{(j)})) + \mathsf{P}_\eta(\mathrm{Fix}_1(D_i^{(j)}))$ for $i \in \{1, 2\}$. Applying (A.3) to $D_1^{(j)}$ and $D_2^{(j)}$, we obtain

$$\kappa^{2(j+1)} \cdot \kappa \cdot \frac{S_1}{2} \geq \kappa^{2(j+1)} \cdot \mathsf{P}_\eta(D_1^{(j)}) > (1+\beta) \cdot \mathsf{P}_\eta(D_2^{(j)}) \geq (1+\beta) \cdot \kappa^{-1} \cdot \frac{S_2}{2}.$$

Multiplying by $2\kappa > 0$ gives $\kappa^{2(j+2)} \cdot S_1 > (1+\beta) \cdot S_2$. Subsequently, there exists $b \in \{0, 1\}$ such that $\kappa^{2(j+2)} \cdot \mathsf{P}_\eta(\mathrm{Fix}_b(D_1^{(j)})) > (1+\beta) \cdot \mathsf{P}_\eta(\mathrm{Fix}_b(D_2^{(j)}))$. The algorithm sets $b_j \triangleq b$ so the invariant (A.4) holds for $j+1$. By induction on $j \leq n$ on a quantifier-free formula, (A.4) also holds for $j = n$.

Let $z \triangleq b_{n-1} \cdots b_1 b_0 \in \{0, 1\}^n$, so that $D_i^{(n)} = \mathrm{Fix}_z(C_i)$ for $i \in \{1, 2\}$. Since $\mathrm{Fix}_z(C_i)$ does not read its input and outputs $C_i(z)$, the BOUNDARY AXIOM FOR APX_1^* gives $\mathsf{P}_\eta(D_i^{(n)}) = C_i(z)$, so (A.4) for $j = n$ means that $\kappa^{2(n+1)} \cdot C_1(z) > (1+\beta) \cdot C_2(z)$. Since the RHS is non-negative, $C_1(z) = 1$, we know that $C_2(z) = 1$ (as $C_1(z) \leq C_2(z)$). Therefore, $\kappa^{2(n+1)} > 1 + \beta$.

Finally, we set $\eta \triangleq \beta/(16(n+1)) \leq 1/16$ to derive a contradiction. Since $\eta \leq 1/5$, we have $(1+\eta)^2 \leq (1+4\eta)(1-\eta)$, i.e., $\kappa \leq 1 + 4\eta$. We use the inequality $(1+s)^k \leq 1 + 2ks$ for $s \geq 0$ and $ks \leq 1/2$, which can be proved in PV_1. Applying it with $s = 4\eta$ and $k = 2(n+1)$ (where $ks = \beta/2$), we obtain $\kappa^{2(n+1)} \leq (1+4\eta)^{2(n+1)} \leq 1 + 16(n+1)\eta = 1 + \beta$, which leads to a contradiction. □

Lemma 5.8 (Non-Existence Lemma for APX_1^*)**.** APX_1^* *proves the following statement. Let* $n, \delta^{-1} \in \mathsf{Log}$ *satisfying* $\delta < 1$ *and* $C : \{0,1\}^n \to \{0,1\}$ *be a circuit. If there is* $x \in \{0,1\}^n$ *such that* $C(x) = 1$*, then* $(1-\delta) \cdot 2^{-n} \lesssim^* \mathsf{P}_\delta(C)$*. (Or equivalently, if* $(1+\beta) \cdot \mathsf{P}_\delta(C) < (1-\delta) \cdot 2^{-n}$ *for some* $\beta^{-1} \in \mathsf{Log}$*, then* $C(x) = 0$ *for every* $x \in \{0,1\}^n$*.)*

Proof. We use the bit-by-bit fixing argument, as in the proofs of Lemma 5.7 and [CLOW26, Lemma 3.7]. We argue in APX_1^*. Fix $n, \delta^{-1} \in \mathsf{Log}$, a circuit $C : \{0,1\}^n \to \{0,1\}$, an input $x = x_1 \cdots x_n \in \{0,1\}^n$ such that $C(x) = 1$, and $\beta^{-1} \in \mathsf{Log}$; our goal is to show that $(1-\delta) \cdot 2^{-n} \leq (1+\beta) \cdot \mathsf{P}_\delta(C)$. Let $\eta^{-1} \in \mathsf{Log}$ be a parameter to be determined (satisfying $\eta \leq 1/8$) and $\kappa \triangleq (1+\eta)^2/(1-\eta)$. (Note that every occurrence of $\lesssim^*$ in the axioms below is instantiated at the slack η.)

Let $D_0 \triangleq C$ and $D_{j+1} \triangleq \mathrm{Fix}_{x_{n-j}}(D_j)$ for $j = 0, 1, \ldots, n-1$, so that D_j is obtained from C by fixing its last j input bits to $x_{n-j+1} \cdots x_n$. In particular, D_n does not read its input and outputs $C(x) = 1$, so the BOUNDARY AXIOM FOR APX_1^* gives $\mathsf{P}_\eta(D_n) = 1$. For every $j < n$, the LOCAL CONSISTENCY AXIOM FOR APX_1^* at precision η gives

$$(1-\eta) \cdot \frac{\mathsf{P}_\eta(\mathrm{Fix}_0(D_j)) + \mathsf{P}_\eta(\mathrm{Fix}_1(D_j))}{2} \leq (1+\eta)^2 \cdot \mathsf{P}_\eta(D_j).$$

Since $\mathsf{P}_\eta(\mathrm{Fix}_{1-x_{n-j}}(D_j)) \geq 0$ by the BASIC AXIOM FOR APX_1^* and $\mathrm{Fix}_{x_{n-j}}(D_j) = D_{j+1}$, this implies that

$$\mathsf{P}_\eta(D_j) \geq \frac{1-\eta}{2(1+\eta)^2} \cdot \mathsf{P}_\eta(D_{j+1}) = \frac{1}{2\kappa} \cdot \mathsf{P}_\eta(D_{j+1}).$$

By induction on $i \leq n$ on a quantifier-free formula, $\mathsf{P}_\eta(D_{n-i}) \geq (2\kappa)^{-i}$ for every $i \in \{0, 1, \ldots, n\}$; in particular, $\mathsf{P}_\eta(C) = \mathsf{P}_\eta(D_0) \geq (2\kappa)^{-n}$.

It remains to switch the counting precision back to δ. By the PRECISION CONSISTENCY AXIOM FOR APX_1^* instantiated with $(\delta_1, \delta_2) \triangleq (\delta, \eta)$, we have $(1-\delta) \cdot \mathsf{P}_\eta(C) \le (1+\eta)^2 \cdot \mathsf{P}_\delta(C) \le \kappa \cdot \mathsf{P}_\delta(C)$, and therefore, as $\delta \le 1$,

$$(1-\delta) \cdot 2^{-n} = \kappa^n \cdot (1-\delta) \cdot (2\kappa)^{-n} \le \kappa^n \cdot (1-\delta) \cdot \mathsf{P}_\eta(C) \le \kappa^{n+1} \cdot \mathsf{P}_\delta(C).$$

Finally we set $\eta \triangleq \beta/(8(n+1)) \le 1/8$. Since $\eta \le 1/5$, we have $(1+\eta)^2 \le (1+4\eta)(1-\eta)$, i.e., $\kappa \le 1+4\eta$. We use the inequality $(1+s)^k \le 1+2ks$ for $s \ge 0$ and $ks \le 1/2$, which can be proved in PV_1. Applying it with $s = 4\eta$ and $k = n+1$ (where $ks = \beta/2$), we obtain $\kappa^{n+1} \le (1+4\eta)^{n+1} \le 1+8(n+1)\eta = 1+\beta$. Hence $(1-\delta) \cdot 2^{-n} \le (1+\beta) \cdot \mathsf{P}_\delta(C)$, as desired. □

Lemma 5.9 (Averaging on Linear Combination for APX_1^*)**.** APX_1^* *proves the following statement. Let* $n, m, \delta^{-1} \in \mathsf{Log}$, $C_1, \dots, C_m : \{0,1\}^n \to \{0,1\}$ *be circuits, and* $\lambda_1^+, \dots, \lambda_m^+, \lambda_1^-, \dots, \lambda_m^- \in \mathbb{Q}$ *be non-negative. Let*

$$\mu^+ \triangleq \sum_{i \in [m]} \lambda_i^+ \cdot \mathsf{P}_\delta(C_i) \qquad \text{and} \qquad \mu^- \triangleq \sum_{i \in [m]} \lambda_i^- \cdot \mathsf{P}_\delta(C_i).$$

Then for every $\beta^{-1} \in \mathsf{Log}$, *there is* $z \in \{0,1\}^n$ *such that*

$$\frac{(1+\beta)(1+\delta) \cdot \sum_{i \in [m]} \lambda_i^+ \cdot C_i(z)}{(1-\delta) \cdot \sum_{i \in [m]} \lambda_i^- \cdot C_i(z)} \ge \frac{\mu^+}{\mu^-},$$

where $a/b \ge c/d$ *is an abbreviation of* $ad \ge bc$. *Moreover, if* $\mu^+ > 0$ *and* $\mu^- > 0$, *then* z *can be chosen such that the inequality is strict, i.e.,* $ad > bc$.

Proof. We follow the bit-by-bit fixing trick in the proof of Lemma 5.7. We argue in APX_1^*. Fix $\beta^{-1} \in \mathsf{Log}$, and for $z \in \{0,1\}^n$ let $\Lambda^+(z) \triangleq \sum_{i \in [m]} \lambda_i^+ \cdot C_i(z)$ and $\Lambda^-(z) \triangleq \sum_{i \in [m]} \lambda_i^- \cdot C_i(z)$. Our goal is to find $z \in \{0,1\}^n$ such that

$$\alpha^+ \cdot \Lambda^+(z) \ge \alpha^- \cdot \Lambda^-(z), \qquad \text{where} \quad \alpha^+ \triangleq (1+\beta)(1+\delta) \cdot \mu^- \quad \text{and} \quad \alpha^- \triangleq (1-\delta) \cdot \mu^+. \tag{A.5}$$

Note that $\alpha^+, \alpha^- \ge 0$, as $\lambda_i^+, \lambda_i^- \ge 0$, $\mathsf{P}_\delta(C_i) \ge 0$ by the BASIC AXIOM FOR APX_1^*, and $\delta \le 1$. Let $\eta^{-1} \in \mathsf{Log}$ be a parameter to be determined and $\kappa \triangleq (1+\eta)^2/(1-\eta)$. (Note that every occurrence of $\lesssim^*$ in the axioms below is instantiated at the slack η.)

For a tuple $\vec{D} = (D_1, \dots, D_m)$ of circuits on a common input length and $b \in \{0,1\}$, let $\mathrm{Fix}_b(\vec{D})$ denote the tuple $(\mathrm{Fix}_b(D_1), \dots, \mathrm{Fix}_b(D_m))$, and let

$$A(\vec{D}) \triangleq \sum_{i \in [m]} \lambda_i^+ \cdot \mathsf{P}_\eta(D_i) \qquad \text{and} \qquad B(\vec{D}) \triangleq \sum_{i \in [m]} \lambda_i^- \cdot \mathsf{P}_\eta(D_i).$$

For every circuit D with at least one input bit, the LOCAL CONSISTENCY AXIOM FOR APX_1^* at precision η gives $(1-\eta)\mathsf{P}_\eta(D) \le (1+\eta)^2 \cdot p_M$ and $(1-\eta) \cdot p_M \le (1+\eta)^2 \cdot \mathsf{P}_\eta(D)$, where $p_M = \frac{1}{2}(\mathsf{P}_\eta(\mathrm{Fix}_0(D)) + \mathsf{P}_\eta(\mathrm{Fix}_1(D)))$; that is, $\kappa^{-1} \cdot p_M \le \mathsf{P}_\eta(D) \le \kappa \cdot p_M$. Multiplying by the non-negative coefficients λ_i^+ (resp. λ_i^-) and summing over $i \in [m]$, we obtain

$$\begin{aligned} A(\vec{D}) &\le \frac{\kappa}{2} \cdot \big(A(\mathrm{Fix}_0(\vec{D})) + A(\mathrm{Fix}_1(\vec{D}))\big) \qquad \text{and} \\ B(\vec{D}) &\ge \frac{1}{2\kappa} \cdot \big(B(\mathrm{Fix}_0(\vec{D})) + B(\mathrm{Fix}_1(\vec{D}))\big). \end{aligned} \tag{A.6}$$

For $j \in \{0, 1, \dots, n\}$ and a tuple $\vec{D}$ of circuits on input length $n-j$, we define the potential function

$$\Phi_j(\vec{D}) \triangleq \kappa^{j-n} \cdot \alpha^+ \cdot A(\vec{D}) - \kappa^{n-j} \cdot \alpha^- \cdot B(\vec{D}).$$

As $\alpha^+, \alpha^- \geq 0$, (A.6) implies that for every $j < n$,

$$\Phi_j(\vec{D}) \leq \frac{1}{2} \cdot \big(\Phi_{j+1}(\mathrm{Fix}_0(\vec{D})) + \Phi_{j+1}(\mathrm{Fix}_1(\vec{D}))\big),$$

and hence there is $b \in \{0,1\}$ such that $\Phi_{j+1}(\mathrm{Fix}_b(\vec{D})) \geq \Phi_j(\vec{D})$. Consider the algorithm that starts from $\vec{C}^{(0)} \triangleq (C_1, \ldots, C_m)$ and, for $j = 0, 1, \ldots, n-1$, computes such a bit b_j (by querying P_η on the $2m$ circuits in $\mathrm{Fix}_0(\vec{C}^{(j)})$ and $\mathrm{Fix}_1(\vec{C}^{(j)})$) and sets $\vec{C}^{(j+1)} \triangleq \mathrm{Fix}_{b_j}(\vec{C}^{(j)})$. By induction on $j \leq n$ on a quantifier-free formula, $\Phi_n(\vec{C}^{(n)}) \geq \Phi_0(\vec{C}^{(0)})$.

Let $z \triangleq b_{n-1} \cdots b_1 b_0 \in \{0,1\}^n$, so that $\vec{C}^{(n)} = (\mathrm{Fix}_z(C_1), \ldots, \mathrm{Fix}_z(C_m))$, where $\mathrm{Fix}_z(C)$ fixes the last $\|z\|$ input bits of C to z. Each circuit $\mathrm{Fix}_z(C_i)$ does not read its input and outputs $C_i(z)$, so the BOUNDARY AXIOM FOR APX_1^* gives $\mathsf{P}_\eta(\mathrm{Fix}_z(C_i)) = C_i(z)$, and therefore

$$\alpha^+ \cdot \Lambda^+(z) - \alpha^- \cdot \Lambda^-(z) = \Phi_n(\vec{C}^{(n)}) \geq \Phi_0(\vec{C}^{(0)}). \tag{A.7}$$

It remains to show that $\Phi_0(\vec{C}^{(0)}) \geq 0$, for which we switch the precision from η back to δ. By the PRECISION CONSISTENCY AXIOM FOR APX_1^* instantiated with $(\delta_1, \delta_2) \triangleq (\eta, \delta)$ and with $(\delta_1, \delta_2) \triangleq (\delta, \eta)$, for every $i \in [m]$,

$$\begin{aligned}
&(1+\delta) \cdot \mathsf{P}_\eta(C_i) \geq \frac{1-\eta}{1+\eta} \cdot \mathsf{P}_\delta(C_i) \geq \frac{1}{\kappa} \cdot \mathsf{P}_\delta(C_i) \qquad \text{and} \\
&(1-\delta) \cdot \mathsf{P}_\eta(C_i) \leq (1+\eta)^2 \cdot \mathsf{P}_\delta(C_i) \leq \kappa \cdot \mathsf{P}_\delta(C_i).
\end{aligned}$$

Multiplying by λ_i^+ (resp. λ_i^-) and summing over $i \in [m]$, we obtain $(1+\delta) \cdot A(\vec{C}^{(0)}) \geq \kappa^{-1} \cdot \mu^+$ and $(1-\delta) \cdot B(\vec{C}^{(0)}) \leq \kappa \cdot \mu^-$. Plugging in α^+ and α^-, and using $\mu^+, \mu^- \geq 0$, we have

$$\begin{aligned}
\Phi_0(\vec{C}^{(0)}) &= \kappa^{-n}(1+\beta)\mu^- \cdot (1+\delta)A(\vec{C}^{(0)}) - \kappa^n \mu^+ \cdot (1-\delta)B(\vec{C}^{(0)}) \\
&\geq \kappa^{-(n+1)} \cdot \mu^+\mu^- \cdot \big((1+\beta) - \kappa^{2(n+1)}\big).
\end{aligned}$$

Finally we set the parameters. Let $\eta \triangleq \beta/(32(n+1)) \leq 1/32$. We have $\kappa = (1+\eta)^2/(1-\eta) \leq 1 + 4\eta$. Recall that $(1+s)^k \leq 1 + 2ks$ if $s \geq 0$ and $ks \leq 1/2$. Applying it with $s = 4\eta$ and $k = 2(n+1)$, we have

$$\kappa^{2(n+1)} \leq (1+4\eta)^{2(n+1)} \leq 1 + 16(n+1)\eta = 1 + \beta/2.$$

Hence $(1+\beta) - \kappa^{2(n+1)} \geq \beta/2$, so $\Phi_0(\vec{C}^{(0)}) \geq \frac{\beta}{2}\kappa^{-(n+1)}\mu^+\mu^- \geq 0$, which together with (A.7) gives (A.5). If moreover $\mu^+ > 0$ and $\mu^- > 0$, then $\Phi_0(\vec{C}^{(0)}) > 0$, so the inequality in (A.5) is strict. □