Is it Easier to Prove Theorems that are Guaranteed to be True?

Rafael Pass
Cornell Tech
rafael@cs.cornell.edu

Muthuramakrishnan Venkitasubramaniam
University of Rochester
muthuv@cs.rochester.edu

2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-7281-9621-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/FOCS46700.2020.00119

***Abstract*—Consider the following two fundamental open problems in complexity theory:**

- **Does a hard-on-average language in NP imply the existence of one-way functions?**
- **Does a hard-on-average language in NP imply a hard-on-average problem in TFNP (i.e., the class of *total* NP search problem)?**

Our main result is that the answer to (at least) one of these questions is yes.

Both one-way functions and problems in TFNP can be interpreted as *promise-true* distributional NP search problems—namely, distributional search problems where the sampler only samples *true* statements. As a direct corollary of the above result, we thus get that the existence of a hard-on-average distributional NP search problem implies a hard-on-average promise-true distributional NP search problem. In other words,

> ***It is no easier to find witnesses (a.k.a. proofs) for efficiently-sampled statements (theorems) that are guaranteed to be true.***

This result follows from a more general study of *interactive puzzles*—a generalization of average-case hardness in NP—and in particular, a novel round-collapse theorem for computationally-sound protocols, analogous to Babai-Moran's celebrated round-collapse theorem for information-theoretically sound protocols. As another consequence of this treatment, we show that the existence of $O(1)$-round public-coin *non-trivial* arguments (i.e., argument systems that are not proofs) imply the existence of a hard-on-average problem in NP/poly.

I. Introduction

Even if NP $\neq$ P, it could be that *in practice*, NP problems are easy in the sense that the problems we encounter in "real life" come from some distribution that make them easy to solve. The complexity-theoretic study of average-case hardness of NP problems addresses this problem [1], [2], [3], [4]. A particularly appealing abstraction of an average-case analog of NP $\neq$ P was provided by Gurevich in his 1989 essay [5] through his notion of a *Challenger-Solver Game*.[1] Consider a probabilistic polynomial-time *Challenger* C who samples an instance x and provides it to the *Solver* S. The solver S is supposed to find a witness to x and is said to win if either (1) the statement x chosen by the challenger is false, or (2) S succeeds in finding a witness w for x. We refer to the Challenger-Solver game as being *hard* if no probabilistic polynomial-time (PPT) solver succeeds in winning in the game with inverse polynomial probability. (In other words, such a game models a hard-on-average distributional search problem in NP.) The existence of a hard Challenger-Solver game means that there exists a way to efficiently sample mathematical statements x that no computationally bounded mathematician can find proofs for. (Impagliazzo [6] considers a similar type of game between Professor Grauss and young Gauss, where Professor Grauss is trying to embarrass Gauss by picking mathematical problems that Gauss cannot solve.)

But, an unappealing aspect of a Challenger-Solver game (which already goes back to the definition of distributional search problems [3]) is that checking whether the solver wins cannot necessarily be efficiently done, as it requires determining whether the sampled instance x is in the language. Does it make the problem easier if we restrict the challenger to *always* sample true statements x?[2] In other words, *"Is it easier to find proofs for efficiently-sampled mathematical statements that are guaranteed to be true?"* In complexity-theoretic terms:

> *Does the existence of an hard-on-average distributional search problem in* NP *imply the existence of a hard-on-average distributional search problem where the sampler* **only samples true statements***?*

We refer to distributional search problems where the sampler only samples true statements as *promise-true* distributional search problems. The above question, and the notion of a promise-true distributional search problems, actually predates the formal study of average-case complexity: It was noted already by Even, Selman and Yacobi [7] in 1984 that for typical applications of (average-case) hardness for NP problems—in particular, for cryptographic applications—we need hardness for instances that are "promised" to be true. As they noted (following [8][3]), in the context of public-key encryption, security is only required for ciphertexts that are sampled as valid encryptions of some message. (This motivated [7] to introduce the concept of a promise problem; see also [11] for further discussion on this issue and the

[1] Gurevich actually outlines several classes of Challenger-Solver games; we here outline one particular instance of it, focusing on NP search problems.

[2] Or equivalently, to distributions where one can efficiently check when the sampler outputs a false instance.

[3] As remarked in [8], these type of "problems with a promise" can be traced back even further: they are closely related to what was referred to as a "birdy" problem in [9] and a "partial algorithm problem" in [10], in the study of context-free languages

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00119

connection to average-case complexity.)

Intuitively, restricting to challengers that only sample true statements ought to make the job of the challenger a lot harder—it now needs to be sure that the sampled instance is true. There are two natural methods for the challenger to achieve this task:

(a) sampling the statement x together with a witness w (as this clearly enables the challenger to be sure that x is true); and,
(b) restricting to NP languages where *every* statement is true.

As noted by Impagliazzo [5], [6], the existence of a challenger-solver game satisfying restriction (a) is equivalent to the existence of one-way functions.[4] But whether the existence of a hard-on-average language in NP implies the existence of one-way functions is arguably the most important open problem in the foundations of Cryptography: One-way functions are both necessary [12] and sufficient for many of the central cryptographic tasks (e.g., pseudorandom generators [13], pseudorandom functions [14], private-key encryption [15], [16]). As far as we know, there are only two approaches towards demonstrating the existence of one-way functions from average-case NP hardness: (1) Ostrovsky and Wigderson [17] demonstrate such an implication assuming that NP has zero-knowledge proofs [18], (2) Komargodski et al. [19] demonstrate the implication (in fact, an even stronger implication, showing worst-case hardness of NP implies one-way functions) assuming the existence of *indistinguishability obfuscators* [20]. Both of these additional assumptions are not known to imply one-way functions on their own (in fact, they are unconditionally true if $\mathsf{NP} \subseteq \mathsf{BPP}$).

A hard challenger-solver game satisfying restriction (b), on the other hand, is syntactically equivalent to a hard-on-average problem in the class TFNP [21]: the class TFNP (total function NP) is the search analog of NP with the additional guarantee that *any* instance has a solution. In other words, TFNP is the class of search problems in $\mathsf{NP} \cap \mathsf{coNP}$ (i.e., $F(\mathsf{NP} \cap \mathsf{coNP})$). In recent years, TFNP has attracted extensive attention due to its natural syntactic subclasses that capture the computational complexity of important search problems from algorithmic game theory, combinatorial optimization and computational topology—perhaps most notable among those are the classes PPAD [22], [23], which characterizes the hardness of computing Nash equilibrium [24], [25], [26], and PLS [27], which characterizes the hardness of local search. A central open problem is whether (average-case) NP hardness implies (average-case) TFNP hardness. A recent elegant result by Hubacek, Naor, and Yogev [28] shows that under certain strong "derandomization" assumptions [29], [30], [31], [32]—the existence of Nisan-Wigderson (NW) [29] type pseudorandom generators that fool circuits with oracle gates to languages in the second level of the polynomial hierarchy[5]—(almost everywhere) average-case hardness of NP implies average-case hardness of TFNP.[6] Hubacek et al. also present another condition under which TFNP is average-case hard: assuming the existence of one-way functions and *non-interactive witness indistinguishable proofs (NIWI)* [33], [34], [32] for NP.

The above mentioned works thus give complexity-theoretic assumptions (e.g., the existence of zero-knowledge proofs for NP, or strong derandomization assumption) under which the above problem has a positive resolution. But these assumptions are both complex and strong.

Our main result provides a resolution to the above problem *without any complexity-theoretic assumption*:[7]

Theorem I.1 (Informally stated)**.** *The existence of an almost-everywhere hard-on-average language in* NP[8] *implies the existence of a hard-on-average promise-true distributional search problem in* NP.

In fact, we demonstrate an even stronger statement. Perhaps surprisingly, we show that without loss of generality, the sampler/challenger of the distributional search problem needs to satisfy one of the above two "natural" restrictions:

Theorem I.2 (Informally stated)**.** *The existence of an almost-everywhere hard-on-average language in* NP *implies either (a) the existence one-way functions, or (b) a hard-on-average* TFNP *problem.*

In other words, in Impagliazzo's Pessiland [6] (a world where NP is hard-on-average, but one-way functions do not exist), TFNP is unconditionally hard (on average).

Towards proving this result, we consider an alternative notion of a Challenger-Solver game, which we refer to as a *Interactive Puzzle*. Roughly speaking, there are 2

[4]That is, a function f that can be computed in polynomial time but cannot be efficiently inverted. Such a function f directly yields the desired sampling method: pick a random string r and let $x = f(r)$ be the statement and r the witness. Conversely, to see why the existence of such a sampling method implies a one-way function, consider the function f that takes the random coins used by the sampling method and outputs the instance generated by it.

[5]Such PRGs are known under the assumption that $E = DTIME[2^{O(n)}]$ has no $2^{\epsilon n}$ sized Π_2-circuits, for all $\epsilon > 0$, where a Π_2-circuit is a standard circuit that can additionally perform oracle queries to any language $L \in \Pi_2$ (i.e., any language in the second level of the polynomial hierarchy).

[6][28] also show that average-case hardness of NP implies an average-case hard problem in $\mathsf{TFNP}/poly$ (i.e,. TFNP with a *non-uniform verifier*). In essence, this follows since non-uniformity enables unconditional derandomization.

[7]Pedantically, it is not a fully complete resolution as we start with an *almost-everywhere* hard problem and only get an *infinitely-often* hard problem. But, except for this minor issue, it is a complete resolution. We also note that earlier results [17], [28] also require starting off with an almost-everywhere hard-on-average language in NP.

[8]That is, a language in NP such that for every $\delta > 0$, no PPT attacker A can decide random instances with probability greater than $\frac{1}{2} + \delta$ for *infinitely many* (as opposed to all) $n \in N$. Such an "almost-everywhere" notion is more commonly used in the cryptographic literature.

differences: (1) whether the solver wins should always be computationally feasible to determine, and (2) we allow for more than just 2 rounds of interaction. As we hope to convey, the study of interactive puzzles is intriguing in its own right and yields other applications.

A. Interactive Puzzles

We initiate a complexity-theoretic study of *interactive puzzles*: 2-player interactive games between a polynomial-time challenger $\mathcal{C}$ and a Solver/Attacker[9] satisfying the following properties:

- **Computational Soundness:** There does not exist a *probabilistic polynomial-time (PPT)* attacker $\mathcal{A}^*$ and polynomial p such that $\mathcal{A}^*(1^n)$ succeeds in making $\mathcal{C}(1^n)$ output 1 with probability $\frac{1}{p(n)}$ for all sufficiently large $n \in N$.
- **Completeness/Non-triviality:** There exists a negligible function μ and an *inefficient* attacker $\mathcal{A}$ that on input 1^n succeeds in making $\mathcal{C}(1^n)$ output 1 with probability $1 - \mu(n)$ for all $n \in N$.
- **Public Verifiability:** Whether $\mathcal{C}$ accepts should just be a deterministic function of the transcript.

In other words, (a) no *polynomial-time* attacker, $\mathcal{A}^*$, can make C output 1 with inverse polynomial probability, yet (b) there exists a *computationally unbounded* attacker $\mathcal{A}$ that makes C output 1 with overwhelming probability. We refer to $\mathcal{C}$ as a $k(\cdot)$-*round computational puzzle* (or simply a $k(\cdot)$-round puzzle) if $\mathcal{C}$ satisfies the above completeness and computational soundness conditions, while restricting $\mathcal{C}(1^n)$ to communicate with $\mathcal{A}$ in $k(n)$ rounds. In this work, we mostly restrict our attention to *public-coin* puzzles, where the Challenger's messages are simply random strings.

As an example of a 2-round public-coin puzzle, let f be a one-way permutation and consider a game where $\mathcal{C}(1^n)$ samples a random $y \in \{0,1\}^n$ and requires the adversary to output a preimage x such that $f(x) = y$. Since f is a permutation, this puzzle has "perfect" completeness—an unbounded attacker $\mathcal{A}$ can always find a pre-image x. By the one-wayness of f (and the permutation property of f), we also have that no PPT adversary $\mathcal{A}^*$ can find such an x (with inverse polynomial probability), and thus soundness holds. If however, f had only been a one-way function and not a permutation, then we can no longer sample a uniform y, but rather must have $\mathcal{C}$ first sample a random x and next output $y = f(x)$. This 2-round puzzle does not satisfy the public-coin property, but it still have perfect completeness.

Its not hard to see that the existence of 2-round (public-coin) puzzles is "essentially" equivalent to the existence of an average-case hard problem in NP: any 2-round public-coin puzzle trivially implies a hard-on-average search problem (w.r.t. the uniform distribution) in NP and thus by [4] also a hard-on-average decision problem in NP. Furthermore, "almost-everywhere" hard-on-average languages in NP also imply the existence of a 2-round puzzle (by simply sampling many random instances x and asking the attacker to provide a witness for at least, say, $1/3$ of the instances).[10]

Proposition I.1 (informally stated)**.** *The existence of an (almost-everywhere) hard-on-average language in* NP *implies the existence of a 2-round puzzle. Furthermore, the existence of a 2-round puzzle implies the existence of a hard-on-average language in* NP.

Thus, 2-round puzzles are "morally" (up to the infinitely-often/almost-everywhere issue) equivalent to the existence of a hard-on-average language in NP. As such, $k(\cdot)$-round puzzles are a natural way to generalize average-case hardness in NP. Additionally, natural restrictions of 2-round puzzles capture natural subclasses of distributional problems in NP:

- the existence of a hard-on-average problem in TFNP is syntactically equivalent to the existence of a 2-round *public-coin* puzzle *with perfect completeness*.
- the existence of a hard-on-average *promise-true* distributional search problem is syntactically equivalent to a 2-round (private-coin) puzzle *with perfect completeness*.

While the game-based modeling in the notion of a puzzle is common in the cryptographic literature—most notably, it is commonly used to model cryptographic assumptions [35], [36], [37], complexity-theoretic consequences or properties of puzzles have remained largely unexplored.

B. The Round-Complexity of Puzzles

Perhaps the most basic question regarding the existence of interactive puzzles is whether the existence of a k-round puzzle is actually a weaker assumption than the existence of a $k-1$ round puzzle. In particular, do interactive puzzles actually generalize beyond just average-case hardness in NP:

> *Does the existence of a k-round puzzle imply the existence of $(k-1)$-round puzzle?*

We here focus our attention only on public-coin puzzles. At first sight, one would hope the classic "round-reduction" theorem due to Babai-Moran (BM) [16] can be applied to collapse any $O(1)$-round puzzle into a 2-round puzzle (i.e., a hard-on-average NP problem). Unfortunately, while BM's round reduction technique indeed works for all *information-theoretically* sound protocols, Wee [38] demonstrated that BM's round reduction fails for computationally sound protocols. In particular, Wee shows that black-box proofs of security cannot be used to prove that BM's transformation

[9]Following the nomenclature in the cryptographic literature, we use the name Attacker instead of Solver.

[10]The reason we need the language to be *almost-everywhere* hard-on-average is to guarantee that YES instances exists for every sufficiently large input length, or else completeness would not hold.

preserves soundness even when applied to just 3-round protocols, and demonstrates (under computational assumptions) a concrete 4-round protocol for which BM's round-reduction results in an unsound protocol.

As BM's round reduction is the only known round-reduction technique (which does not rely on any assumptions), it was generally conjectured that the existence of a k-round puzzle is a strictly stronger assumption than the existence of a $(k+1)$-round puzzle—in particular, this would imply the existence of infinitely many worlds between Impagliazzo's Pessiland and Heuristica [6] (i.e., infinitely many worlds where $\mathsf{NP} \neq P$ yet average-case NP hardness does not exist). Further evidence in this direction comes from a work by Gertner et al. [39] which shows a black-box separation between k-round puzzles and $(k+1)$-round puzzles for a particular cryptographic task (namely that of a key-agreement scheme).[11]

In contrast to the above negative results, our main technical result provides an affirmative answer to the above question—we demonstrates a round-reduction theorem for puzzles.

Theorem I.3 (informally stated)**.** *For every constant c, the existence of a $k(\cdot)$-round public-coin puzzle is equivalent to the existence of a $(k(\cdot)-c)$-round public-coin puzzle.*

In particular, as corollary of this result, we get that the assumption that a $O(1)$-round public-coin puzzle exists is *not* weaker than the assumption that average-case hardness in NP exists:

Corollary I.4 (informally stated)**.** *The existence of an $O(1)$-round puzzle implies the existence of a hard-on-average problem in* NP.

Perhaps paradoxically, we strongly rely on BM's round reduction technique, yet we rely on a *non-black-box* security analysis. Our main technical lemma shows that if *infinitely-often one-way functions*[12] do not exist (i.e., if we can invert any function for all sufficiently large input lengths), then BM's round reduction actually works:

Lemma I.2 (informally stated)**.** *Either infinitely-often one-way functions exist, or BM's round-reduction transformation turns a $k(\cdot)$-round puzzle into a $(k(\cdot)-1)$-round puzzle.*

We provide a proof outline of Lemma I.2 in Section I-E. The proof of Theorem I.3 now easily follows by considering two cases:

Case 1: (Infinitely-often) one-way functions exists. In such a world, we can rely on Rompel's construction of a universal one-way hash function [40], [41] to get a 2-round puzzle.

Case 2: (Infinitely-often) one-way functions does not exist. In such a world, by Lemma I.2, BM's round reduction preserves soundness of the underlying protocol and thus we have gotten a puzzle with one round less. We can next iterate BM's round reduction any constant number of times.

A natural question is whether we can collapse more than a constant number of rounds. Our next result—which characterizes the existence of $\mathsf{poly}(n)$-round puzzles—shows that this is unlikely.

Theorem I.5 (informally stated)**.** *For every $\epsilon > 0$, there exists an n^{ϵ}-round (public-coin) puzzle if and only if* $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$.

In particular, if n^{ϵ}-round public-coin puzzles imply $O(1)$-round public-coin puzzles, then by combining Theorem I.3 and Theorem I.5, we have that $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$ implies the existence of a hard-on-average problem in NP, which seems unlikely. Theorem I.5 also shows that the notion of an interactive puzzle (with a super constant-number of rounds) indeed is a non-trivial generalization of average-case hardness in NP. Theorem I.5 follows using mostly standard techniques.[13]

We next present some complexity-theoretic consequences of our treatment of interactive puzzles.

C. Achieving Perfect Completeness: Proving Theorem I.2

We outline how the round-reduction theorem can be used to prove Theorem I.2 in the following steps:

- As mentioned above, an (almost-everywhere) hard-on-average problem in NP yields a 2-round puzzle;
- We can next use a standard technique from the literature on interactive proofs (namely the result of [46]) to turn this puzzle into a 3-round puzzle with *perfect completeness*.
- We next observe that the BM transformation preserves perfect completeness of the protocol. Thus, by Lemma

[11] The example from [39] isn't quite captured by our notion of a computational puzzle as their challenger is not public coin.

[12] Recall that a *one-way function* f is a function that is efficiently computable, yet there does not exist a PPT attacker A and polynomial $p(\cdot)$ such that A inverts f with probability $\frac{1}{p(n)}$ for *infinitely many* inputs lengths $n \in N$. A function f is *infinitely often one-way* if the same conditions hold except that we only require that no PPT attacker A succeeds in inverting f with probability $\frac{1}{p(n)}$ *for all* sufficiently large $n \in N$—i.e., it is hard for invert f "infinitely often"

[13] Any puzzle $\mathcal{C}$ can be broken using a PSPACE oracle (as the optimal strategy can be found using a PSPACE oracle), so if $\mathsf{PSPACE} \subseteq \mathsf{BPP}$, it can also be broken by a probabilistic polynomial-time algorithm. For the other direction, recall that worst-case to average-case reductions are known for PSPACE [42], [43]. In other words, there exists a language $L \in \mathsf{PSPACE}$ that is hard-on-average assuming $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$. Additionally, recall that PSPACE is closed under complement. We then construct a public-coin puzzle where $\mathcal{C}$ first samples a hard instance for L and then asks $\mathcal{A}$ to determine whether $x \in L$ and next provide an interactive proof—using [44], [45] which is public-coin—for containment or non containment in L. This puzzle clearly satisfies the completeness condition. Computational soundness, on the other hand, follows directly from the hard-on-average property of L (and the unconditional soundness of the interactive proof of [44]).

I.2, either infinitely-often one-way functions exist, or we can get a 2-round puzzle with perfect completeness.
- Finally, as observed above, the existence of a 2-round puzzle with perfect completeness is syntactically equivalent to the existence of a hard-on-average problem in TFNP (with respect to the uniform distribution on instances).

The above proof approach actually only concludes a slightly weaker form of Theorem I.2—we only show that either TFNP is hard or *infinitely-often* one-way functions exist. As infinitely-often one-way functions directly imply 2-round *private-coin* puzzles with perfect completeness, which (as observed above) are syntactically equivalent to hard-on-average *promise-true* distributional search problems, this however already suffices to prove Theorem I.1.

We can get the proof also of the stronger conclusion of Theorem I.2 (i.e., conclude the existence of standard (i.e., "almost-everywhere") one-way functions), by noting that an almost-everywhere hard-on-average language in NP actually implies an 2-round puzzle satisfying a "almost-everywhere" notion of soundness, and for such "almost-everywhere puzzles", Lemma I.2 can be strengthened to show that either one-way functions exist, or BM's round-reduction works.[14]

D. The Complexity of Non-trivial Public-coin Arguments

Soon after the introduction of interactive proof by Goldwasser, Micali and Rackoff [47] and Babai and Moran [16], Brassard, Chaum and Crepeau [48] introduced the notion of an interactive *argument*. Interactive arguments are defined identically to interactive proofs, but we relax the soundness condition to only hold with respect to non-uniform PPT algorithms (i.e., no non-uniform PPT algorithm can produce proofs of false statements, except with negligible probability).

Interactive arguments have proven extremely useful in the cryptographic literature, most notably due to the feasibility (assuming the existence of collision-resistant hashfunctions) of *succinct* public-coin argument systems for NP—namely, argument systems with sublinear, or even polylogarithmic communication complexity [49], [50]. Under widely believed complexity assumptions (i.e., NP not being solvable in subexponential time), interactive *proofs* cannot be succinct [51].

A fundamental problem regarding interactive arguments involves characterizing the complexity of *non-trivial* argument systems—namely interactive arguments that are *not* interactive proofs (in other words, the soundness condition is inherently computational). As far as we know, the first explicit formalization of this question appears in a recent work by Goldreich [52], but the notion of a non-trivial argument has been discussed in the community for at least 15 years.[15]

We focus our attention on *public-coin* arguments (similar to our treatment of puzzles). Using our interactive-average-case hardness treatment, we are able to establish an "almost-tight" characterization of constant-round public-coin non-trivial arguments.

Theorem I.6 (informally stated)**.** *The existence of a $O(1)$-round public-coin non-trivial argument for any language L implies a hard-on-average language in* NP/poly*. Conversely, the existence of a hard-on-average language in* NP *implies an (efficient-prover) 2-round public-coin non-trivial argument for* NP*.*

The first part of the theorem is shown by observing that any public-coin non-trivial argument can be turned into a *non-uniform* public-coin puzzle (where the challenger is a non-uniform PPT algorithm), and next observing that our round-collapse theorem also applies to non-uniform puzzles. The second part follows from the observation that we can take any NP proof for some language L and extending it into a 2-round non-trivial argument for L where the verifier samples a random statement x' from a hard-on-average language L' and next requiring the prover to provide a witness w that either $x \in L$ or $x' \in L'$. Completeness follows trivially (as we can always provide a normal NP witness proving that $x \in L$, and computational soundness follows directly if L' is sufficiently hard-on-average (in the sense that it is hard to find witnesses to true statements with inverse polynomial probability). This argument system is not a proof, though, since by the hard-on-average property of L', there must exist infinitely many input lengths for which random instances are contained in L' with inverse polynomial probability.

We finally observe that the existence of n^ϵ-round non-trivial public-coin arguments is equivalent to PSPACE $\not\subseteq$ P/poly.

Theorem I.7 (informally stated)**.** *For every $\epsilon > 0$, there exists an (efficient-prover) n^ϵ-round non-trivial public-coin argument (for* NP*) if and only if* PSPACE $\not\subseteq$ P/poly*.*

The "only-if" direction was already proven by Goldreich [52] and follows just as the only-if direction of Theorem I.5. The "if" direction follows by combining a standard NP proof with the puzzle from Theorem I.5 (which becomes sound w.r.t. nu PPT attacker assuming PSPACE $\not\subseteq$ P/poly), and requiring the prover to either provide the NP witness, or to provide a solution to the puzzle.

[14] More precisely, the variant of Lemma I.2 says that either one-way functions exist, or the existence of a k-round almost-everywhere puzzle yields the existence of a $k-1$-round puzzle (with the standard, infinitely-often, notion of soundness).

[15] Wee [53] also considers a notion of a non-trivial argument, but his notion refers to what today is called a succinct argument.

E. Proof Overview for Lemma I.2

We here provide a proof overview of our main technical lemma. As mentioned, we shall show that if one-way functions do not exist, then Babai-Moran's round reduction method actually works. Towards this we will rely on two tools:

- *Pre-image sampling*. By the result of Impagliazzo and Levin [4], the existence of so-called "distributional one-way functions" (function for which it is hard to sample a uniform pre-image) imply the existence of one-way function. So if one-way functions do not exist, we have that for every efficient function f, given a sample $f(x)$ for a random input x, we can efficiently sample a (close to random) pre-image x'.
- *Raz's sampling lemma* (from the literature on parallel repetition for 2-prover games and interactive arguments [54], [55], [56]). This lemma states that if we sample ℓ uniform n-bit random variables $R_1, R_2, \ldots R_\ell$ conditioned on some event W that happens with sufficiently large probability ϵ, then the conditional distribution R_i of a randomly selected index i will be close to uniform. More precisely, the statistical distance will be $\sqrt{\frac{\log(\frac{1}{\epsilon})}{\ell}}$, so even if ϵ is tiny, as long as we have sufficiently many repetitions ℓ, the distance will be small.[16]

To see how we will use these tools, let us first recall the BM transformation (and its proof for the case of information-theoretically sound protocols). To simplify our discussion, we here focus on showing how to collapse a 3-round public-coin protocol between a prover P and a public-coin verifier V into a 2-round protocol. We denote a transcript of the 3-round protocol (p_1, r_1, p_2) where p_1 and p_2 are the prover messages and r_1 is the randomness of the verifier. Let $n = |p_1|$ be the length of the prover message. The BM transformation collapses this protocol into a 2-round protocol in the following two steps:

- **Step 1: Reducing soundness error:** First, use a form of parallel repetition to make the soundness error 2^{-n^2} (i.e., *extremely small*). More precisely, consider a 3-round protocol where P first still send just p_1, next the verifier picks $\ell = n^2$ random strings $\vec{r} = (r_1^1, \ldots, r_1^\ell)$, and finally P needs to provide accepting answers $\vec{p_2} = (p_2^1, \ldots, p_2^\ell)$ to all of the queries $\vec{r}$ (so that for every $i \in [\ell]$, (p_1, r_1^i, p_2^i) is accepting transcript).
- **Step 2: Swap order of messages:** Once the soundness error is small, yet the length of the first message is short, we can simply allow the prover to pick its first message p_1 after having $\vec{r}$. In other words, we now have a 2-round protocol where V first picks $\vec{r}$, then the prover responds by sending $p_1, \vec{p_2}$. This swapping preserves soundness by a simple union bound: since (by soundness) for every string p_1, the probability over $\vec{r}$ that there exists some accepting response $\vec{r}$ is 2^{-n^2}, it follows that with probability at most $2^n \times 2^{-n^2} = 2^{-n}$ over $\vec{r}$, *there exists some* p_1 that has an accepting $\vec{p_2}$ (as the number of possible first messages p_1 is 2^n). Thus soundness still holds (with a 2^n degradation) if we allow P to choose p_1 after seeing $\vec{r}$.

For the case of computationally sound protocols, the "logic" behind both steps fail: (1) it is not known how to use parallel repetition to reduce soundness error beyond being negligible, (2) the union bound cannot be applied since, for computationally sound protocols, it is not the case that responses $\vec{p_2}$ do not exist, rather, they are just hard to find. Yet, as we shall see, using the above tools, we present a different proof strategy. More precisely, to capture computational hardness, we show a reduction from any polynomial-time attacker A that breaks soundness of the collapsed protocol with some inverse polynomial probability ϵ, to a polynomial-time attacker B that breaks soundness of the original 3-round protocol.

B starts by sampling a random string $\vec{r'}$ and computes A's response given this challenge $(p_1', \vec{p_2'}) \leftarrow A(\vec{r'})$. If the response is not an accepting transcript, simply abort; otherwise, take p_1' and forward externally as B's first message. (Since A is successful in breaking soundness, we have that B won't abort with probability ϵ.) Next, B gets a verifier challenge r from the external verifier and needs to figure out how to provide an answer to it. If B is lucky and r is one of the challenges r_i' in $\vec{r'}$, then B could provide the appropriate p_2 message, but this unfortunately will only happen with negligible probability. Rather, B will try to get A to produce another accepting transcript $(p_1'', \vec{r''}, \vec{p_2''})$ that (1) still contains p_1' as the prover's first message (i.e., $p_1'' = p_1'$), and (2) contains r in some coordinate i of $\vec{r''}$. To do this, B will consider the function $f(\vec{r}, z, i)$—which runs $(p_1, \vec{p_2}) \leftarrow A(\vec{r}; z)$ (i.e., A has its randomness fixed to z) and outputs (p_1, r_i) if $(p_1, \vec{r}, \vec{p_2})$ is accepting and $\perp$ otherwise—and runs the pre-image sampler for this function f on (p_1', r) to recover some new verifier challenge, randomness, index tuple $(\vec{r''}, z, i)$ which leads $A(\vec{r''}; z)$ to produce a transcript $(p_1', \vec{r''}, \vec{p_2''})$ of the desired form, and B can subsequently forward externally the i'th coordinate of $\vec{p_2''}$ as its response and convince the external verifier.

So, as long as the pre-image sampler indeed succeeds with high enough probability, we have managed to break soundness of the original 3-round protocol. The problem is that the pre-image sampler is only required to work given outputs that are correctly distributed over the range of the function f, and the input (p_1, r) that we now feed it may not be so—for instance, perhaps $A(\vec{r})$ chooses the string p_1 as a function of $\vec{r}$. So, whereas the marginal distribution of both p_1 and r are correct, the *joint* distribution is not. In particular, the distribution of r conditioned on p_1 may be

[16] Earlier works [55], [56] always used Raz' lemma when ϵ was non-negligible. In contrast, we will here use it also when ϵ is actually negligible.

off. We, however, show how to use Raz's lemma to argue that if the number of repetitions ℓ is sufficiently bigger than the length of p_1, the conditional distribution of r cannot be too far off from being uniform (and thus the pre-image sampler will work). On a high-level, we proceed as follows:

- Note that in the one-way function experiment, we can think of the output distribution (p_1, r) of f on a random input, as having been produced by first sampling p_1 and next, if $p_1 \neq \perp$, sampling $\vec{r}$ conditioned on the event W_{p_1} that A generates a successful transcript with first-round prover message p_1, and finally sampling a random index i and outputting p_1 and r_i (and otherwise output $\perp$).
- Note that by an averaging argument, we have that with probability at least $\frac{\epsilon}{2}$ over the choice of p_1, $\Pr[W_{p_1}] \geq \frac{\epsilon}{2^{n+1}}$ (otherwise, the probability that A succeeds would need to be smaller than $\frac{\epsilon}{2} + 2^n \times \frac{\epsilon}{2^{n+1}} = \epsilon$, which is a contradiction).
- Thus, whenever we pick such a "good" p_1 (i.e., a p_1 such that $\Pr[W_{p_1}] \geq \frac{\epsilon}{2^{n+1}}$), by Raz' lemma the distribution of r_i for a random i can be made $\frac{1}{p(n)}$ close to uniform for any polynomial p by choosing ℓ to be sufficiently large (yet polynomial). Note that even though the lower bound on $\Pr[W_{p_1}]$ is negligible, the key point is that it is independent of ℓ and as such we can still rely on Raz lemma by choosing a sufficiently large ℓ. (As we pointed out above, this usage of Raz' lemma even on very "rare" events—with negligible probability mass—is different from how it was previously applied to argue soundness for computationally sound protocols [55], [56].)
- It follows that conditioned on picking such a "good" p_1, the pre-image sampler will also successfully generate correctly distributed preimages if we feed him p_1, r where r is randomly sampled. But this is exactly the distribution that B feeds to the pre-image sampler, so we conclude that with probability $\frac{\epsilon}{2}$ over the choice of p_1, B will manage to convince the outside verifier with probability close to 1.

This concludes the proof overview for 3-round protocols. When the protocol has more than 3 rounds, we can apply a similar method to collapse the last rounds of the protocol. The analysis now needs to be appropriately modified to condition also on the prefix of the partial execution up until the last rounds.

II. Preliminaries

We assume familiarity with basic concepts such as Turing machines, interactive Turing machine, polynomial-time algorithms, probabilistic polynomial-time algorithms (PPT), non-uniform polynomial-time and non-uniform PPT algorithms. A function μ is said to be *negligible* if for every polynomial $p(\cdot)$ there exists some n_0 such that for all $n > n_0$, $\mu(n) \leq \frac{1}{p(n)}$. For any two random variables X and Y, we let $\mathsf{SD}(X, Y) = \max_{T \subseteq U} |\Pr[X \in T] - \Pr[Y \in T]|$ denote the *statistical distance* between X and Y.

Basic Complexity Classes: Recall that P is the class of languages L decidable in polynomial time (i.e., there exists a polynomial-time algorithm M such that for every $x \in \{0,1\}^*$, $M(x) = L(x)$), P/poly is the class of languages decidable in non-uniform polynomial time, and BPP is the class of languages decidable in probabilistic polynomial time with probability $2/3$ (i.e., there exists a PPT M such that for every $x \in \{0,1\}^*$, $\Pr[M(x) = L(x)] > 2/3$ where we abuse of notation and define $L(x) = 1$ if $x \in L$ and $L(x) = 0$ otherwise.)

We refer to a relation $\mathcal{R}$ over pairs (x, y) as being *polynomially bounded* if there exists a polynomial $p(\cdot)$ such that for every $(x, y) \in \mathcal{R}$, $|y| \leq p(|x|)$. We denote by $L_{\mathcal{R}}$ the language characterized by the "witness relation" $\mathcal{R}$—i.e., $x \in L$ iff there exists some y such that $(x, y) \in \mathcal{R}$. We say that a relation $\mathcal{R}$ is *polynomial-time* (resp. non-uniform polynomial-time) if $\mathcal{R}$ is polynomially-bounded and the languages consisting of pairs $(x, y) \in \mathcal{R}$ is in P (resp. P/poly). NP (resp NP/poly) is the class of languages L for which there exists a polynomial-time (resp. non-uniform polynomial-time) relation $\mathcal{R}$ such that $x \in L$ iff there exists some y such that $(x, y) \in \mathcal{R}$.

Search Problems: A search problem $\mathcal{R}$ is simply a polynomially-bounded relation; an NP search problem $\mathcal{R}$ is a polynomial-time relation. We say that the search problem is *solvable in polynomial-time (resp. non-uniform polynomial time)* if there exists a polynomial-time (resp. non-uniform polynomial-time) algorithm M that for every $x \in L_{\mathcal{R}}$ outputs a "witness" y such that $(x, y) \in \mathcal{R}$. Analogously, $\mathcal{R}$ is *solvable in* PPT if there exists some PPT M that for every $x \in L_{\mathcal{R}}$ outputs a "witness" y such that $(x, y) \in \mathcal{R}$ with probability $2/3$.

An NP search problem $\mathcal{R}$ is *total* if for every $x \in \{0,1\}^*$ there exists some y such that $(x, y) \in \mathcal{R}$) (i.e., every instance has a witness). We refer to FNP (function NP) as the class of NP search problems and TFNP (total-function NP) as the class of total NP search problems.

A. One-way functions

We recall the definition of one-way functions (see e.g., [57]). Roughly speaking, a function f is one-way if it is polynomial-time computable, but hard to invert for PPT attackers. The standard (cryptographic) definition of a one-way function requires every PPT attacker to fail (with high probability) on all sufficiently large input lengths. We will also consider a weaker notion of an *infinitely-often* one-way function [17] which only requires the PPT attacker to fail for infinitely many inputs length (in other words, there is no PPT attacker that succeeds on all sufficiently large input lengths, analogously to complexity-theoretic notions of hardness).

Definition II.1. *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. f is said to be a* one-way function (OWF) *if for every* PPT *algorithm A, there exists a negligible function μ such that for all $n \in \mathbb{N}$,*

$$\Pr[x \leftarrow \{0,1\}^n; y = f(x) : A(1^n, y) \in f^{-1}(f(x))] \leq \mu(n)$$

f is said to be an infinitely-often one-way function (ioOWF) *if the above condition holds for infinitely many $n \in \mathbb{N}$ (as opposed to all).*

We may also consider a notion of a *non-uniform* (a.k.a. "auxiliary-input") one way function, which is identically defined except that (a) we allow f to be computable by a non-uniform PPT, and (b) the attacker A is also allowed to be a non-uniform PPT.

B. Average-Case Complexity

We recall some basic notions from average-case complexity. A *distributional problem* is a pair $(L, \mathcal{D})$ where $L \subseteq \{0,1\}^*$ and $\mathcal{D}$ is a PPT; we say that $(L, \mathcal{D})$ is an NP (resp. NP/poly) distributional problem if $L \in$ NP (resp. $L \in$ NP/poly). Roughly speaking, a distributional problem $(L, \mathcal{D})$ is hard-on-average if there does not exist some PPT algorithm that can decide instances drawn from $\mathcal{D}$ with probability significantly better than $1/2$.

Definition II.2 (δ-hard-on-the-average)**.** *We say that a distributional problem $(L, \mathcal{D})$ is* δ-hard-on-the-average (δ-HOA) *if there does not exist some* PPT *A such that for every sufficiently large $n \in \mathbb{N}$,*

$$\Pr[x \leftarrow \mathcal{D}(1^n) : A(1^n, x) = L(x)] > 1 - \delta$$

We say that a distributional problem $(L, \mathcal{D})$ is simply hard-on-the-average (HOA) *if it is δ-HOA for some $\delta > 0$.*

We also define an notion of HOA w.r.t. non-uniform PPT algorithm (*nuHAO*) in exactly the same way but where we allow A to be a non-uniform PPT (as opposed to just a PPT).

The above notion of average-case hardness (traditionally used in the complexity-theory literature) is defined analogously to the notion of an *infinitely-often* one-way function: we simply require every PPT "decider" to fail for infinitely many $n \in \mathbb{N}$. For our purposes, we will also rely on an "almost-everywhere" notion of average-case hardness (similar to standard definitions in the cryptography, and analogously to the definition of a one-way function), where we require that every decider fails on *all* (sufficiently large) input lengths.

Definition II.3 (almost-everywhere hard-on-the-average (aeHOA))**.** *We say that a distributional problem $(L, \mathcal{D})$ is* almost-everywhere δ hard-on-the-average (δ-aeHOA) *if there does not exist some* PPT *A such that for infinitely many $n \in \mathbb{N}$,*

$$\Pr[x \leftarrow \mathcal{D}(1^n) : A(1^n, x) = L(x)] > 1 - \delta$$

We say $(L, \mathcal{D})$ is almost-everywhere hard-on-the-average (aeHOA) *if $(L, \mathcal{D})$ is δ-aeHOA for some $\delta > 0$.*

We move on to defining hard-on-the-average *search problems*. A *distributional search problem* is a pair $(\mathcal{R}, \mathcal{D})$ where $\mathcal{R}$ is a search problem and $\mathcal{D}$ is a PPT. If $\mathcal{R}$ is an NP search problem (resp. NP/poly search problem), we refer to $(\mathcal{R}, \mathcal{D})$ as a distributional NP (resp. NP/poly) search problem.

Finally, we say that a distributional search problem $(\mathcal{R}, \mathcal{D})$ is *promise-true* if for every n and every x in the support of $\mathcal{D}(1^n)$, it holds that $x \in L_{\mathcal{R}}$. (That is, $\mathcal{D}$ only samples true instances.)

III. Interactive Puzzles

Roughly speaking, an *interactive puzzle* is described by an interactive polynomial-time challenger $\mathcal{C}$ having the property that (a) there exists an inefficient $\mathcal{A}$ that succeeds in convincing $\mathcal{C}(1^n)$ with probability negligibly close to 1, yet (b) no PPT attacker $\mathcal{A}^*$ can make $\mathcal{C}(1^n)$ output 1 with inverse polynomial probability for sufficiently large n.

Definition III.1 (interactive puzzle)**.** *An interactive algorithm $\mathcal{C}$ is referred to as a $k(\cdot)$-*round puzzle *if the following conditions hold:*

- **$k(\cdot)$-round, publicly-verifiability:** *$\mathcal{C}$ is an (interactive)* PPT *that on input 1^n (a) only communicates in $k(n)$ communication rounds, and (b) only performs some deterministic computation as a function of the transcript to determine its final verdict.*
- **Completeness/Non-triviality:** *There exists a (possibly unbounded) Turing machine $\mathcal{A}$ and a negligible function $\mu_C(\cdot)$ such that for all $n \in \mathbb{N}$,*

$$\Pr[\langle \mathcal{A}, \mathcal{C} \rangle(1^n) = 1] \geq 1 - \mu(n)$$

- **Computational Soundness:** *There does not exist a* PPT *machine $\mathcal{A}^*$ and polynomial $p(\cdot)$ such that for all sufficiently large $n \in \mathbb{N}$,*

$$\Pr[\langle \mathcal{A}^*, \mathcal{C} \rangle(1^n) = 1] \geq \frac{1}{p(n)}$$

In other words, a $k(\cdot)$-round puzzle, $\mathcal{C}$, gives rise to an $k(\cdot)$-round interactive proof (P, V) (where $P = \mathcal{A}, V = \mathcal{C}$) for the "trivial" language $L = \{0,1\}^*$ with the property that there does not exist a PPT prover that succeeds in convincing the verifier with inverse polynomial probability for all sufficiently large n.

We will consider several restricted, or alternative, types of puzzle:

- We refer to the puzzle $\mathcal{C}$ as being *public-coin* if $\mathcal{C}$ simply sends the outcomes of its coin tosses in each communication round.
- We may also define an *almost-everywhere* notion of a puzzle by replacing "for all sufficiently large $n \in \mathbb{N}$" in the soundness condition with "for infinitely many $n \in \mathbb{N}$", and a *non-uniform* notion of a puzzle $\mathcal{C}$ which

allows both $\mathcal{C}$ and $\mathcal{A}^*$ to be *non-uniform* PPT (as opposed to just PPT).
- Finally, a puzzle $\mathcal{C}$ is said to have *perfect completeness* if the "completeness error", $\mu_C(n)$, is 0—in other words, the completeness condition holds with probability 1.

Remark III.1. *One can consider a more relaxed notion of a $(c(\cdot), s(\cdot))$-puzzle for $c(n) > s(n) + \frac{1}{\mathsf{poly}(n)}$, where the completeness condition is required to hold with probability $c(\cdot)$ for every sufficiently large $n \in \mathbb{N}$, and the soundness condition holds with probability $s(\cdot)$ for every sufficiently large $n \in \mathbb{N}$. But, by "Chernoff-type" parallel-repetition theorems for computationally-sound protocols [58], [59], [55], [60], [56], the existence of such a $k(\cdot)$-round $(c(\cdot), s(\cdot))$-puzzle implies the existence of a $k(\cdot)$-round puzzle. The same holds for almost-everywhere (resp. non-uniform) puzzles.*

In the remainder of this extended abstract, we state our results and the proofs can be found in the full version [61].

A. The Round-Collapse Theorem

In this section, we state our main theorems about puzzles and some variants.

Our main lemma shows that if ioOWF do not exist, the the Babai-Moran transformation preserves computational soundness.

Lemma III.2. *Assume there exists a $k(\cdot)$-round public-coin puzzle such that $k(n) \geq 3$. Then, either there exists an ioOWF, or there exists a $(k(\cdot)-1)$-round public-coin puzzle. Moreover, if the $k(\cdot)$-round puzzle has perfect completeness, then either there exists an ioOWF, or a $(k(\cdot)-1)$-round public-coin puzzle with perfect-completeness.*

Variations: Using essentially the same proofs, we can directly get the following vacations of III.2. The first variant simply states that the same result holds for almost-everywhere puzzles.

Lemma III.3 (Almost-everywhere variant 1)**.** *Assume there exists a $k(\cdot)$-round almost-everywhere public-coin puzzle such that $k(n) \geq 3$. Then, either there exists an ioOWF, or there exists a $(k(\cdot)-1)$-round almost-everywhere public-coin puzzle. Moreover, if the $k(\cdot)$-round puzzle has perfect completeness, then either there exists an ioOWF, or a $(k(\cdot)-1)$-round almost-everywhere public-coin puzzle with perfect-completeness.*

The next variant shows that if we start off with an almost-everywhere puzzle, we can either get a (standard) one-way function or a puzzle with one less round (but this new puzzle no longer satisfies almost-everywhere security) iThis follows from the fact that if the attacker A^* succeeds on all sufficiently large input lengths, then it suffices for Inv to work on infinitely many input lengths, to conclude that B^{Inv} works on infinitely many inputs length (thus violating almost-everywhere security of the original puzzle).

Lemma III.4 (Almost-everywhere variant 2)**.** *Assume there exists a $k(\cdot)$-round almost-everywhere public-coin puzzle such that $k(n) \geq 3$. Then, either there exists a OWF, or there exists a $(k(\cdot)-1)$-round public-coin puzzle. Moreover, if the $k(\cdot)$-round puzzle has perfect completeness, then either there exists a OWF, or a $(k(\cdot)-1)$-round public-coin puzzle with perfect-completeness.*

We additionally consider a variant for non-uniform puzzles. As the challenger now may be a non-uniform PPT, the function M that we are required to invert is also a non-uniform PPT and thus we can only conclude the existence of non-uniform OWFs.

Lemma III.5 (Non-uniform variant)**.** *Assume there exists a $k(\cdot)$-round non-uniform public-coin puzzle such that $k(n) \geq 3$. Then, either there exists a non-uniform ioOWF, or there exists a $(k(\cdot)-1)$-round non-uniform public-coin puzzle.*[17]

B. Characterizing O(1)-Round Public-coin Puzzles

We next apply our round-collapse theorem (and its variants) to get a characterization of $O(1)$-round puzzles. This characterization applies to both standard puzzles and non-uniform puzzles.

Corollary III.2. *Assume the existence of a $O(1)$-round (resp. a $O(1)$-round non-uniform) public-coin puzzle. Then there exists a 2-round public-coin puzzle (resp. 2-round non-uniform public-coin puzzle) and thus a distributional* NP *problem (resp. distributional* NP/poly *problem) that is HOA (resp. nuHOA).*

We remark that the reason we cannot get an (unconditional) characterization of almost-everywhere puzzles is that ioOWFs. are not known to imply 2-round almost-everywhere puzzles.

IV. Characterizing Polynomial-round Puzzles

We observe that the existence of a poly-round public-coin puzzle is equivalent to the statement that PSPACE $\not\subseteq$ BPP. A consequence of this result is that any round-collapse theorem that (unconditionally) can transform a polynomial-round puzzle into a $O(1)$-round puzzle, must show the existence of a HAO distributional NP problem based on the assumption that PSPACE $\not\subseteq$ BPP (which would be highly unexpected).

Theorem IV.1. *For every $\epsilon > 0$, there exists an n^ϵ-round public-coin puzzle (resp. a non-uniform puzzle) if and only if* PSPACE $\not\subseteq$ BPP *(resp.* PSPACE $\not\subseteq$ P/poly*).*

[17] The transformation still preserves perfect completeness, but this will not be of relevance for us.

V. Achieving Perfect Completeness

We show that any 2-round public-coin puzzle can be transformed into a 3-round public-coin puzzle with perfect completeness; next, we shall use this result together with our round-reduction theorem to conclude our main result.

A. From Imperfect to Perfect Completeness (by Adding a Round)

Furer et al. [46] showed how to transform any 2-round public-coin proof system into a 3-round public-coin proof system with perfect completeness. We will rely on the same protocol transformation to transform any 2-round puzzle into a 3-round puzzle with perfect completeness. The perfect completeness condition will follow directly from their proof; we simply must argue that the transformation also preserves *computational* soundness (as they only showed that it preserves information-theoretic soundness).

Theorem V.1. *Suppose there exists* 2*-round public-coin puzzle. Then there exists a* 3*-round public-coin puzzle with perfect completeness.*

B. Promise-true Distributional Problems

We now conclude our main theorem that a hard-on-average language in NP implies hard-on-average promise-true distributional search problem.

We first show that 2-round public-coin puzzles imply 2-round (private-coin) puzzles with perfect completeness:

Theorem V.2. *Suppose there exists* 2*-round public-coin puzzle. Then there exists a* 2*-round private-coin puzzle with perfect completeness.*

By observing that 2-round *private-coin* puzzles with perfect completeness are syntactically equivalent to a hard-on-average *promise-true* distributional search problem, and recalling that by Lemma **??**, aeHOA distributional NP problem implies a 2-round puzzle, we directly get the following corollary:

Corollary V.3. *Suppose there exists a distributional* NP *problem* $(L, \mathcal{D})$ *that is aeHOA. Then, there exists a hard-on-average promise-true distributional* NP *search problem.*

In other words, "it isn't easier to prove efficiently-sampled statements that are guaranteed to the true".

C. TFNP is Hard in Pessiland

We next use the same approach to conclude that a hard-on-average language in NP implies either (1) the existence of one-way functions, or (2) the existence of a hard-on-average problem in TFNP.

Theorem V.4. *Suppose there exists a distributional* NP *problem* $(L, \mathcal{D})$ *that is aeHOA. Then, either of the following holds:*

- *There exists a OWF;*
- *There exists some* $\mathcal{R} \in$ TFNP *and some* PPT $\mathcal{D}$ *such that* $(\mathcal{R}, \mathcal{D})$ *is SearchHAO.*

By replacing the use of Lemma III.4 with Lemma III.3 (round-collapse, variant 1), we instead get the following variants.

Theorem V.5. *Suppose there exists a distributional* NP *problem* $(L, \mathcal{D})$ *that is aeHOA. Then, either of the following holds:*

- *There exists an ioOWF;*
- *There exists some* $\mathcal{R} \in$ TFNP *and some* PPT $\mathcal{D}$ *such that* $(\mathcal{R}, \mathcal{D})$ *is aeSearchHAO.*

VI. Characterizing Non-trivial Public-coin Arguments

We finally apply our round-collapse theorem to arguments systems.

Non-trivial arguments: We first define the notion of a non-trivial argument. Whereas such a notion of a non-trivial argument has been discussed in the community for at least 15 years, as far as we know, the first explicit formalization in the literature appears in a recent work by Goldreich [52]. We simply say that an argument system is *non-trivial* if it is not a proof systems—i.e., the computation aspect of the soundness condition is "real".

Definition VI.1 (non-trivial arguments)**.** *An argument system* (P, V) *for a language* L *is called* non-trivial *if* (P, V) *is not an interactive proof system for* L*.*

We focus our attention on *public-coin arguments*. We show that the existence of any $O(1)$-round public-coin non-trivial argument implies the existence of distributional NP/poly problem that is nuHAO.

Theorem VI.2. *Assume there exists a* $O(1)$*-round public-coin non-trivial argument for some language* L*. Then, there exists a distributional* NP/poly *problem that is nuHOA.*

We next remark that the implication is almost tight. The existence of a nuHOA problem in NP (as opposed to NP/poly) implies a 2-round non-trivial public-coin argument for NP.

Lemma VI.1. *Suppose there exists a distributional* NP *problem* $(L', \mathcal{D})$ *that is nuHOA . Then, for every language* $L \in$ NP*, there exists a non-trivial 2-round public-coin argument for* L *with an efficient prover.*

We finally observe that the existence of n^{ϵ}-round non-trivial public-coin arguments is equivalent to PSPACE $\not\subseteq$ P/poly. We remark that one direction (that non-trivial arguments imply PSPACE $\not\subseteq$ P/poly) was already previously proven by Goldreich [52].

Theorem VI.3 (informally stated)**.** *For every* $\epsilon > 0$*, there exists an (efficient-prover)* n^{ϵ}*-round non-trivial public-coin argument (for* NP*) if and only if* PSPACE $\not\subseteq$ P/poly.

Round Collapse for Succinct Arguments: We proceed to remark that the proof of our round-collapse theorem also has consequences for succinct [49] and universal [50], [63] argument systems.

Theorem VI.4. *Assume there exists a k-round public-coin (efficient-prover) argument system for L with communication complexity $\ell(\cdot)$, where k is a constant. Then, either non-uniform ioOWFs exists, or there exists a 2-round public-coin (efficient-prover) argument for L with communication complexity $O(\ell(n)\mathsf{polylog}(n))^{k(n)-1}$.*

Theorem VI.4 thus shows that the existence of a $O(1)$-round succinct (i.e., with sublinear or polylogarithmic communication complexity) public-coin argument systems can either be collapsed into a 2-round public-coin succinct argument for the same language (and while preserving communication complexity up to polylogarithmic factors, as well as prover efficiency), or non-uniform ioOWF exist.

It is worthwhile to also note that if the underlying $O(1)$-round protocol satisfies some notion of *resettable* [64] privacy for the prover (e.g., resettable witness indistinguishability (WI) or witness hiding (WH) [64], [33]), then so will the resulting 2-round protocol. (The reason we do not consider resettable zero-knowledge is that due to [65] even just plain zero-knowledge protocols for non-trivial languages imply the existence of a non-uniform ioOWF; thus for resettable zero-knowledge, the result would hold vacuously assuming $NP \not\subseteq \mathsf{BPP}$. However, it is not known whether (resettable) WI or WH arguments for non-trivial languages imply non-uniform ioOWFs.)

VII. Acknowledgements

The authors thank the anonymous reviewers for their helpful suggestions. We are grateful to Johan Håstad and Salil Vadhan for discussions about non-trivial arguments back in 2005. We are also very grateful to Eylon Yogev for helpful discussions.

The first author was supported in part by NSF Award SATC-1704788, NSF Award RI-1703846, and AFOSR Award FA9550-18-1-0267. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2019-19-020700006. The second author was supported by Google Faculty Research Grant, NSF Award CNS-1618884 and Intelligence Advanced Research Projects Activity (IARPA) via 2019-19-020700009. Work done partially at Cornell Tech sponsored by Cornell Tech and DIMACS Research Visit Program via DIMACS/Simons Collaboration in Cryptography.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] L. A. Levin, "Average case complete problems," *SIAM J. Comput.*, vol. 15, no. 1, pp. 285–286, 1986.

[2] Y. Gurevich, "Average case completeness," *J. Comput. Syst. Sci.*, vol. 42, no. 3, pp. 346–398, 1991. [Online]. Available: https://doi.org/10.1016/0022-0000(91)90007-R

[3] S. Ben-David, B. Chor, O. Goldreich, and M. Luby, "On the theory of average case complexity," *J. Comput. Syst. Sci.*, vol. 44, no. 2, pp. 193–219, 1992.

[4] R. Impagliazzo and L. A. Levin, "No better ways to generate hard NP instances than picking uniformly at random," in *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume II*, 1990, pp. 812–821.

[5] Y. Gurevich, "The challenger-solver game: variations on the theme of p=np," in *Logic in Computer Science Column, The Bulletin of EATCS*, 1989.

[6] R. Impagliazzo, "A personal view of average-case complexity," in *Structure in Complexity Theory '95*, 1995, pp. 134–147.

[7] S. Even, A. L. Selman, and Y. Yacobi, "The complexity of promise problems with applications to public-key cryptography," *Information and Control*, vol. 61, no. 2, pp. 159–173, 1984.

[8] S. Even and Y. Yacobi, "Cryptocomplexity and np-completeness," in *Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherlands, July 14-18, 1980, Proceedings*, 1980, pp. 195–207.

[9] S. Ginsburg, *The Mathematical Theory of Context-Free Languages*. USA: McGraw-Hill, Inc., 1966.

[10] J. S. Ullian, "Partial algorithm problems for context free languages," *Information and Control*, vol. 11, no. 1/2, pp. 80–101, 1967. [Online]. Available: https://doi.org/10.1016/S0019-9958(67)90401-9

[11] O. Goldreich, "On promise problems: A survey," in *Theoretical Computer Science, Essays in Memory of Shimon Even*, 2006, pp. 254–290.

[12] R. Impagliazzo and M. Luby, "One-way functions are essential for complexity based cryptography (extended abstract)," in *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, 1989, pp. 230–235.

[13] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.

[14] O. Goldreich, S. Goldwasser, and S. Micali, "On the cryptographic applications of random functions," in *CRYPTO*, 1984, pp. 276–288.

[15] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[16] L. Babai and S. Moran, "Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes," *J. Comput. Syst. Sci.*, vol. 36, no. 2, pp. 254–276, 1988.

[17] R. Ostrovsky and A. Wigderson, "One-way fuctions are essential for non-trivial zero-knowledge," in *ISTCS*, 1993, pp. 3–17.

[18] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems," *J. ACM*, vol. 38, no. 3, pp. 691–729, 1991.

[19] I. Komargodski, T. Moran, M. Naor, R. Pass, A. Rosen, and E. Yogev, "One-way functions and (im)perfect obfuscation," *IACR Cryptology ePrint Archive*, vol. 2014, p. 347, 2014.

[20] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," in *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, 2001, pp. 1–18.

[21] N. Megiddo and C. H. Papadimitriou, "On total functions, existence theorems and computational complexity," *Theor. Comput. Sci.*, vol. 81, no. 2, pp. 317–324, 1991.

[22] C. H. Papadimitriou, "On the complexity of the parity argument and other inefficient proofs of existence," *J. Comput. Syst. Sci.*, vol. 48, no. 3, pp. 498–532, 1994.

[23] P. W. Goldberg and C. H. Papadimitriou, "Towards a unified complexity theory of total functions," Unpublished manuscript, 2016. [Online]. Available: http://www.cs.ox.ac.uk/people/paul.goldberg/papers/paper-2.pdf

[24] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," *Commun. ACM*, vol. 52, no. 2, pp. 89–97, 2009.

[25] X. Chen, X. Deng, and S. Teng, "Settling the complexity of computing two-player nash equilibria," *J. ACM*, vol. 56, no. 3, pp. 14:1–14:57, 2009.

[26] C. Daskalakis and C. H. Papadimitriou, "Continuous local search," in *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, 2011, pp. 790–804.

[27] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, "How easy is local search? (extended abstract)," in *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, 1985, pp. 39–42.

[28] P. Hub'avcek, M. Naor, and E. Yogev, "The journey from NP to TFNP hardness," in *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, 2017, pp. 60:1–60:21.

[29] N. Nisan and A. Wigderson, "Hardness vs randomness," *J. Comput. Syst. Sci.*, vol. 49, no. 2, pp. 149–167, 1994.

[30] R. Impagliazzo and A. Wigderson, "$P = BPP$ if e requires exponential circuits: Derandomizing the xor lemma," in *STOC '97*, 1997, pp. 220–229.

[31] P. B. Miltersen and N. V. Vinodchandran, "Derandomizing arthur-merlin games using hitting sets," *Computational Complexity*, vol. 14, no. 3, pp. 256–279, 2005.

[32] B. Barak, S. J. Ong, and S. P. Vadhan, "Derandomization in cryptography," *SIAM J. Comput.*, vol. 37, no. 2, pp. 380–400, 2007.

[33] U. Feige and A. Shamir, "Witness indistinguishable and witness hiding protocols," in *STOC '90*, 1990, pp. 416–426.

[34] C. Dwork and M. Naor, "Zaps and their applications," in *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, 2000, pp. 283–293.

[35] M. Naor, "On cryptographic assumptions and challenges," in *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, 2003, pp. 96–109.

[36] R. Pass, "Limits of provable security from standard assumptions," in *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, 2011, pp. 109–118.

[37] C. Gentry and D. Wichs, "Separating succinct non-interactive arguments from all falsifiable assumptions," in *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, 2011, pp. 99–108.

[38] H. Wee, "Finding pessiland," in *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, 2006, pp. 429–442.

[39] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan, "The relationship between public key encryption and oblivious transfer," in *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, 2000, pp. 325–335.

[40] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," in *STOC '89*, 1989, pp. 33–43.

[41] J. Rompel, "One-way functions are necessary and sufficient for secure signatures," in *STOC*, 1990, pp. 387–394.

[42] J. Feigenbaum and L. Fortnow, "Random-self-reducibility of complete sets," *SIAM Journal on Computing*, vol. 22, no. 5, pp. 994–1005, 1993.

[43] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson, "BPP has subexponential time simulations unless EXPTIME has publishable proofs," *Computational Complexity*, vol. 3, pp. 307–318, 1993.

[44] A. Shamir, "IP = PSPACE," *J. ACM*, vol. 39, no. 4, pp. 869–877, 1992.

[45] C. Lund, L. Fortnow, H. J. Karloff, and N. Nisan, "Algebraic methods for interactive proof systems," *J. ACM*, vol. 39, no. 4, pp. 859–868, 1992.

[46] M. Fürer, O. Goldreich, Y. Mansour, M. Sipser, and S. Zachos, "On completeness and soundness in interactive proof systems," *Advances in Computing Research*, vol. 5, pp. 429–442, 1989.

[47] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186–208, 1989.

[48] G. Brassard, D. Chaum, and C. Crépeau, "Minimum disclosure proofs of knowledge," *J. Comput. Syst. Sci.*, vol. 37, no. 2, pp. 156–189, 1988.

[49] J. Kilian, "A note on efficient zero-knowledge proofs and arguments (extended abstract)," in *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, 1992, pp. 723–732.

[50] S. Micali, "Computationally sound proofs," *SIAM J. Comput.*, vol. 30, no. 4, pp. 1253–1298, 2000.

[51] O. Goldreich and J. Håstad, "On the complexity of interactive proofs with bounded communication," *Inf. Process. Lett.*, vol. 67, no. 4, pp. 205–214, 1998.

[52] O. Goldreich, "On doubly-efficient interactive proof systems," *Foundations and Trends in Theoretical Computer Science*, vol. 13, no. 3, pp. 158–246, 2018. [Online]. Available: https://doi.org/10.1561/0400000084

[53] H. Wee, "On round-efficient argument systems," in *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, 2005, pp. 140–152.

[54] R. Raz, "A parallel repetition theorem," *SIAM Journal on Computing*, vol. 27, no. 3, pp. 763–803, 1998.

[55] J. Håstad, R. Pass, D. Wikström, and K. Pietrzak, "An efficient parallel repetition theorem," in *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, 2010, pp. 1–18.

[56] K. Chung and R. Pass, "Tight parallel repetition theorems for public-coin arguments using kl-divergence," in *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, 2015, pp. 229–246.

[57] O. Goldreich, *Foundations of Cryptography — Basic Tools*. Cambridge University Press, 2001.

[58] R. Pass and M. Venkitasubramaniam, "A parallel repetition theorem for constant-round arthur-merlin proofs," *TOCT*, vol. 4, no. 4, pp. 10:1–10:22, 2012.

[59] I. Haitner, "A parallel repetition theorem for any interactive argument," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 16, p. 27, 2009.

[60] K. Chung and F. Liu, "Parallel repetition theorems for interactive arguments," in *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, 2010, pp. 19–36.

[61] R. Pass and M. Venkitasubramaniam, "A round-collapse theorem for computationally-sound protocols; or, TFNP is hard (on average) in pessiland," *CoRR*, vol. abs/1906.10837, 2019. [Online]. Available: http://arxiv.org/abs/1906.10837

[62] L. Trevisan and S. P. Vadhan, "Pseudorandomness and average-case complexity via uniform reductions," *Computational Complexity*, vol. 16, no. 4, pp. 331–364, 2007.

[63] B. Barak and O. Goldreich, "Universal arguments and their applications," in *IEEE Conference on Computational Complexity*, 2002, pp. 194–203.

[64] R. Canetti, O. Goldreich, S. Goldwasser, and S. Micali, "Resettable zero-knowledge (extended abstract)," in *STOC '00*, 2000, pp. 235–244.

[65] R. Ostrovsky and A. Wigderson, "One-way functions are essential for non-trivial zero-knowledge," in *Theory and Computing Systems, 1993*, 1993, pp. 3–17.

[66] L. Trevisan, "On uniform amplification of hardness in NP," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, 2005, pp. 31–38.