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

April 15, 2020

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.

*Cornell Tech. 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 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.

†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.

Contents

1 Introduction

Even if $\mathsf{NP} \neq \mathsf{P}$, 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 [Lev86, Gur91, BCGL92, IL90]. A particularly appealing abstraction of an average-case analog of $\mathsf{NP} \neq \mathsf{P}$ was provided by Gurevich in his 1989 essay [Gur89] 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 [Imp95] 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 [BCGL92]) 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 [ESY84] 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 [EY80][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 [ESY84] to introduce the concept of a promise problem; see also [Gol06] for further discussion on this issue and the 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,

[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 [EY80], 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 [Gin66] and a "partial algorithm problem" in [Ull67], in the study of context-free languages

(b) restricting to NP languages where *every* statement is true.

As noted by Impagliazzo [Gur89, Imp95], 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 [IL89] and sufficient for many of the central cryptographic tasks (e.g., pseudorandom generators [HILL99], pseudorandom functions [GGM84], private-key encryption [GM84, BM88]). 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 [OW93a] demonstrate such an implication assuming that NP has zero-knowledge proofs [GMW91], (2) Komargodski et al. [KMN+14] 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* [BGI+01]. 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 [MP91]: 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 [Pap94, GP16], which characterizes the hardness of computing Nash equilibrium [DGP09, CDT09, DP11], and PLS [JPY85], 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 [HNY17] shows that under certain strong "derandomization" assumptions [NW94, IW97, MV05, BOV07]—the existence of Nisan-Wigderson (NW) [NW94] 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)* [FS90, DN00, BOV07] 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]

[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][HNY17] 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 [OW93a, HNY17] also require starting off with an almost-everywhere hard-on-average language in NP.

Theorem 1.1 (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 1.2 (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 [Imp95] (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 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.

1.1 Interactive Puzzles

We initiate a complexity-theoretic study of *interactive puzzles*: 2-player interactive games between a polynomial-time challenger $\mathcal{C}$ and an 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

[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.

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

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 [IL90] 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 1.1 (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 [Nao03, Pas11, GW11], complexity-theoretic consequences or properties of puzzles have remained largely unexplored.

1.2 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) [BM88] 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 [Wee06] 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 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

[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.

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 [Imp95] (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. [GKM+00] 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 1.3 (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 1.4 (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 1.2 (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 1.2 in Section 1.5. The proof of Theorem 1.3 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 hashfunction [NY89, Rom90] to get a 2-round puzzle.
- **Case 2: (Infinitely-often) one-way functions does not exist.** In such a world, by Lemma 1.2, 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 1.5 (informally stated)**.** *For every $\epsilon > 0$, there exists an n^{ϵ}-round (public-coin) puzzle if and only if $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$.*

[11] The example from [GKM+00] 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"

In particular, if n^ϵ-round public-coin puzzles imply $O(1)$-round public-coin puzzles, then by combining Theorem 1.3 and Theorem 1.5, we have that $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$ implies the existence of a hard-on-average problem in NP, which seems unlikely. Theorem 1.5 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 1.5 follows using mostly standard techniques.[13]

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

1.3 Achieving Perfect Completeness: Proving Theorem 1.2

We outline how the round-reduction theorem can be used to prove Theorem 1.2 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 [FGM+89]) 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 1.2, 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 1.2—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 1.1.

We can get the proof also of the stronger conclusion of Theorem 1.2 (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 1.2 can be strengthened to show that either one-way functions exist, or BM's round-reduction works.[14]

[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 [FF93, BFNW93]. 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 [Sha92, LFKN92] 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 [Sha92]).

[14] More precisely, the variant of Lemma 1.2 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).

1.4 The Complexity of Non-trivial Public-coin Arguments

Soon after the introduction of interactive proof by Goldwasser, Micali and Rackoff [GMR89] and Babai and Moran [BM88], Brassard, Chaum and Crepeau [BCC88] 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 [Kil92, Mic00]. Under widely believed complexity assumptions (i.e., NP not being solvable in subexponential time), interactive *proofs* cannot be succinct [GH98].

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 [Gol18], 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 1.6 (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 $\mathsf{NP}/\mathsf{poly}$. 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 $\mathsf{PSPACE} \not\subseteq \mathsf{P}/\mathsf{poly}$.

Theorem 1.7 (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 $\mathsf{PSPACE} \not\subseteq \mathsf{P}/\mathsf{poly}$.*

The "only-if" direction was already proven by Goldreich [Gol18] and follows just as the only-if direction of Theorem 1.5. The "if" direction follows by combining a standard NP proof with the puzzle from Theorem 1.5 (which becomes sound w.r.t. nu PPT attacker assuming $\mathsf{PSPACE} \not\subseteq \mathsf{P}/\mathsf{poly}$), and requiring the prover to either provide the NP witness, or to provide a solution to the puzzle.

[15] Wee [Wee05] also considers a notion of a non-trivial argument, but his notion refers to what today is called a succinct argument.

1.5 Proof Overview for Lemma 1.2

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 [IL90], 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 [Raz98, HPWP10, CP15]). 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 it 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

[16] Earlier works [HPWP10, CP15] always used Raz' lemma when ϵ was non-negligible. In contrast, we will here use it also when ϵ is actually negligible.

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 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 [HPWP10, CP15].)

- 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.

2 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 $x \in \{0,1\}^*$, $M(x) = L(x)$), $\mathsf{P/poly}$ 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. $\mathsf{P/poly}$). NP (resp $\mathsf{NP/poly}$) 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.

2.1 One-way functions

We recall the definition of one-way functions (see e.g., [Gol01]). 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 [OW93a] 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 2.1. *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.

2.2 Interactive Proofs and Arguments

We recall basic definitions of interactive proofs [GMR89, BM88] and arguments [BCC88]. An interactive protocol (P, V) is a pair of interactive Turing machine; we denote by $\langle P_1, P_2\rangle(x)$ the output of P_2 in an interaction between P_1 and P_2 on common input x.

Definition 2.2. *An interactive protocol (P, V) is an* interactive proof system *for a language $L \subseteq \{0,1\}^*$, if V is PPT and the following conditions hold:*

Completeness: *There exists a negligible function $\mu(\dot{c})$ such that for every $x \in L$,*

$$\Pr[\langle P, V\rangle(x) = 1] \geq 1 - \mu(|x|)$$

Soundness: *For every Turing machine P^*, there exists a negligible function $\mu(\cdot)$ such that for every $x \notin L$,*

$$\Pr[\langle P^*, V\rangle(x) = 1] \leq \mu(|x|)$$

If the soundness condition is relaxed to only hold for all non-uniform PPT P^*, we refer to (P, V) as an interactive argument *for L. We refer to (P, V) as a* public-coin proof/argument system *if V simply sends the outcomes of its coin tosses to the prover (and only performs computation to determine its final verdict).*

Whenever $L \in \mathsf{NP}$, we say that (P, V) has an efficient prover *if there exists some witness relation $\mathcal{R}$ that characterizes L (i.e., $L_{\mathcal{R}} = L$) and a PPT $\widetilde{P}$ such that $P(x) = \widetilde{P}(x, w)$ satisfies the completeness condition for every $(x, w) \in \mathcal{R}$.*

2.3 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. $\mathsf{NP/poly}$) distributional problem if $L \in \mathsf{NP}$ (resp. $L \in \mathsf{NP/poly}$). 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 2.3 (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 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 2.4 (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 an 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.)

Definition 2.5 (hard-on-the-average search (SearchHOA))**.** *We say that a distributional search problem* $(\mathcal{R}, \mathcal{D})$ *is* δ-hard-on-the-average (δ-SearchHOA) *if there does not exist some* PPT A *such that for every sufficiently large* $n \in N$*,*

$$\Pr[x \leftarrow \mathcal{D}(1^n); (w, x) \leftarrow A(1^n, x) : ((L_{\mathcal{R}}(x) = 1) \Rightarrow (x, w) \in \mathcal{R})] > 1 - \delta$$

$(\mathcal{R}, \mathcal{D})$ *is simply* SearchHOA *if there exists* $\delta > 0$ *such that* $(\mathcal{R}, \mathcal{D})$ *is* δ*-SearchHOA.*

We can analogously define an almost-everywhere notion, *aeSearchHOA*, of SearchHAO (by replacing "for every sufficiently large $n \in N$" with "for infinitely many $n \in N$") as well as a non-uniform notion, *nuSearchHOA*, (by replacing PPT with non-uniform PPT).

The following lemmas which essentially directly follow from the result of [IL90, BCGL92, Tre05] will be useful to us. (These results were originally only stated for the standard notion of HOA, whereas we will require it also for the almost-everywhere notion; as we explain in more detail in Appendix A, these results however directly apply also for the almost-everywhere notion of HOA.) The first results from [IL90] (combined with [Tre05]) shows that without loss of generality, we can restrict our attention to the uniform distribution over statements x; we denote by $\mathcal{U}_p$ a PPT such that $\mathcal{U}_p(1^n)$ simply samples a random string in $\{0,1\}^{p(n)}$.

Lemma 2.1 (Private to public distributions)**.** *Suppose there exists a distributional* NP *problem* $(L, \mathcal{D})$ *that is HOA (resp., aeHOA or nuHAO). Then, there exists a polynomial* p *and an* NP*-language* L' *such that* $(L', \mathcal{U}_p)$ *is HAO (resp. aeHOA or nuHOA).*

The next result from [Tre05]) shows that when the distribution over instances is uniform, we can amplify the hardness.

Lemma 2.2 (Hardness amplification)**.** *Let* p *be a polynomial and suppose there exists a distributional* NP*-problem* $(L, \mathcal{U}_p)$ *that is HOA (resp., aeHOA or nuHOA). Then, for every* $\delta < \frac{1}{2}$*, there exists some polynomial* p' *and* NP *language* L' *such that* $(L', \mathcal{U}_{p'})$ *is* δ*-HOA (resp.,* δ*-aeHOA or* δ*-nuHOA).*

Finally, by [BCGL92] (combined with [Tre05], [IL90]) we have a decision-to-search reduction.

Lemma 2.3 (Search to decision)**.** *Suppose there exists a distributional* NP *(resp.* NP/poly*) search problem* $(\mathcal{R}, \mathcal{D})$ *that is SearchHOA (resp., nuSearchHOA). Then, there a polynomial* p *and an* NP *(resp.* NP/poly*) language* L *such that* $(L', \mathcal{U}_p)$ *is HOA (resp., nuHOA).*

3 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 3.1 (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 3.1. *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 [PV12, Hai09, HPWP10, CL10, CP15], 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.*

3.1 Characterizing 2-round Public-coin Puzzles

In this section we make some basic observations regarding 2-round public-coin puzzles; these results mostly follow using standard results in the literature. We begin by observing that the existence of ioOWF imply the existence of 2-round public-coin puzzles.

Proposition 3.2. *Assume the existence of ioOWFs (resp. non-uniform ioOWF). Then, there exists a 2-round public-coin puzzle (reps. non-uniform puzzle).*

Proof: By the result of Rompel [Rom90] (see also [KK05, HHR$^+$10]), we have that ioOWFs imply the existence of infinitely-often "second-preimage" resistant hash-function families that compress n bits to $n/2$ bits.[17] This, in turn, directly yields a simple 2-round puzzle where the challenger $\mathcal{C}(1^n)$ uniformly samples a hashfunction h and input $x \in \{0,1\}^n$ and sends (h, x) to the adversary; $\mathcal{C}$ accepts a response x' if $|x'| = n$, $x' \neq x$ and $h(x') = h(x)$. Since the hash function is compressing, we have that there exists a negligible function μ such that with probability $1 - \mu(n)$, a random $x \in \{0,1\}^n$ will have a "collision" x' and thus an unbounded $\mathcal{A}$ can easily find a collision and thus completeness follows. Computational soundness, on the other hand, directly from the (infinitely-often) second-preimage resistance property. The same result holds also if we start with non-uniform ioOWFs, except that we now get a non-uniform puzzle. ■

We turn to showing that any aeHOA distributional NP problem implies a 2-round puzzle. (In fact, it even implies an almost-everywhere puzzle.)

Lemma 3.3. *Suppose there exists a distributional NP problem $(L, \mathcal{D})$ that is aeHOA. Then there exist an (almost-everywhere) 2-round public-coin puzzle.*

Proof: Assume there exists a distributional problem $(L, \mathcal{D})$ such that $L \in \mathsf{NP}$ and (L, D) is aeHOA. From Lemma 2.2 and Lemma 2.1, we can conclude that there exists a polynomial p and a distributional NP problem $(L', \mathcal{U}_p)$ that is δ-aeHOA for $\delta = \frac{3}{8}$. Let $\mathcal{R}'$ be some NP relation corresponding to L'. Consider a puzzle $\mathcal{C}$ where $\mathcal{C}(1^n)$ samples a random $x \in \{0,1\}^{p(n)}$ and accepts

[17] Roughly speaking, a family of public coin hashfunctions H having the property that for a random $h \in H$ and random input x, it is hard for any PPT to find a different x' of the same length that collides with x under h (that is, $h(x) = h(x')$, $|x| = |x'|$ yet $x \neq x'$. Rompel's theorem was only stated for standard OWFs (as opposed to ioOWFs, but the construction and proof directly also works for the infinitely-often variant as well.

a response y if $(x, y) \in \mathcal{R}'$. We will show that $\mathcal{C}$ is a $(\frac{3}{8}, \frac{1}{4})$-puzzle which by Remark 3.1 implies the existence of a 2-round almost-everywhere puzzle. To show completeness, consider an inefficient algorithm $\mathcal{A}$ that on input $(1^n, x)$ tries to find a witness y (using brute-force) such that $(x, y) \in \mathcal{R}'$ and if it is successful sends it to $\mathcal{C}$ (and otherwise simply aborts). Observe that for all sufficiently large $n \in \mathbb{N}$, for a random $x \leftarrow \{0,1\}^{p(n)}$ we have that $\Pr[x \in L'] > \frac{3}{8}$; otherwise, $(L', \mathcal{U}_p)$ can be decided with probability $1 - \frac{3}{8}$ for infinitely many $n \in \mathbb{N}$ contradicting its $\frac{3}{8}$-aeHOA property.[18] It follows that for all sufficiently large $n \in N$, $\mathcal{A}$ convinces $\mathcal{C}$ with probability $\frac{3}{8}$ and thus completeness of $\mathcal{C}$ follows.

To prove soundness, assume for contradiction that there exists a PPT algorithm $\mathcal{A}^*$ such that $\Pr[\langle \mathcal{A}^*, \mathcal{C}\rangle(1^n) = 1] > \frac{1}{4}$ for infinitely many n. Consider the machine $M(x)$ that runs $\mathcal{A}^*(1^{|x|}, x)$ and outputs 1 if $\mathcal{A}^*$ outputs a valid witness for x and otherwise outputs a random bit. By definition, M solves the distributional problem $(L', \mathcal{U}_p)$ with probability $> \frac{1}{4} + \frac{1}{2}\left(1 - \frac{1}{4}\right) = \frac{5}{8} = 1 - \frac{3}{8}$ for infinitely many n, which contradicts the $\frac{3}{8}$-aeHAO property of $(L', \mathcal{U}_p)$. ■

We now turn to showing that 2-round puzzles imply a HOA distributional NP problem. It will be useful for the sequel to note that the same result also holds in the non-uniform setting.

Lemma 3.4. *Suppose there exists a 2-round public-coin puzzle (resp. a non-uniform puzzle). Then, there exists a distributional* NP *problem (resp. distributional* $\mathsf{NP/poly}$ *problem) that is HOA (resp. nuHOA).*

Proof: Let $\mathcal{C}$ be a 2-round public-coin puzzle (resp. 2-round non-uniform puzzle). Let $\ell(\cdot)$ be an upper bound on the amount of randomness used by $\mathcal{C}$. Consider the NP-relation (resp. $\mathsf{NP/poly}$-relation) $\mathcal{R}$ that includes all tuples $((pad, x), y)$ such that $\mathcal{C}(1^{|pad|})$ given randomness $x \in \ell(|pad|)$ accepts upon receiving y, and the sampler $\mathcal{D}(1^n)$ that picks a random $x \in \{0,1\}^{\ell(n)}$ and outputs $(1^n, x)$. We argue next that $(\mathcal{R}, \mathcal{D})$ is $\frac{1}{8}$-SearchHOA (resp $\frac{3}{8}$-nuSearchHOA), which concludes the proof by applying Lemma 2.3. Assume for contradiction that there exists a PPT (resp. non-uniform PPT) machine M that solves $(\mathcal{R}, \mathcal{D})$ with probability $> 1 - \frac{1}{8} = \frac{7}{8}$ for all $n > n_0$. By the completeness of $\mathcal{C}$, there exists some $\mathcal{A}, n_1$ such that such that for every $n > n_1$, $\Pr[\langle \mathcal{A}, \mathcal{C}\rangle(1^n) = 1] > \frac{7}{8}$. This implies that for all $n > n_1$, for at most an $\frac{1}{8}$ fraction of $\ell(n)$-bit strings x, $(1^n, x) \notin L_{\mathcal{R}}$). In particular, for every $n > \max(n_0, n_1)$, for a random $x \in \{0,1\}^{\ell(n)}$, $M(1^n, x)$ must output a valid witness y for x with probability $> \frac{7}{8} - \frac{1}{8} > \frac{1}{2}$, and can thus be used to break the soundness of the puzzle with probability $> \frac{1}{2}$ for all sufficiently large n which is a contradiction. ■

If the 2-round puzzle has perfect completeness, essentially the same proof gives a SearchHOA problem in TFNP as the relation $\mathcal{R}$ constructed in the proof of Lemma 3.4 is total if the puzzle has perfect completeness.

Lemma 3.5. *Suppose there exists a 2-round public-coin puzzle (resp. almost-everywhere puzzle) with perfect completeness. Then, there exists some search problem* $\mathcal{R} \in \mathsf{TFNP}$ *and some* PPT $\mathcal{D}$ *such that the distributional search problem* $(\mathcal{R}, \mathcal{D})$ *is SearchHAO (aeSearchHAO).*

4 The Round-Collapse Theorem

In this section, we prove our main technical lemma—a round-collapse theorem for $O(1)$-round puzzles.

[18] Note that this is where were are crucially relying on the almost-everywhere hardness of the distributional problem.

4.1 An Efficient Babai-Moran Theorem

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

Lemma 4.1. *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.*

Proof: Consider some $k(\cdot)$-round public-coin puzzle $\mathcal{C}$ and assume for contradiction that ioOWF do not exist. We will show that Babai-Moran's (BM) [BM88] round reduction works in this setting and thus we can obtain a $(k(\cdot)-1)$-round puzzle.

Note that if ioOWF do not exist, every polynomial-time computable function is "invertible" with inverse polynomial probability for all sufficiently long input lengths n. In fact, since by [IL90], the existence of *distributional one-way functions* implies the existence of one-way functions (and this results also works in the infinitely-often setting), we can conclude that if ioOWF do not exist, for any polynomial $p(\cdot)$, and any polynomial-time computable function f, there exists a PPT algorithm Inv such that, for sufficiently large n, the following distributions are $\frac{1}{p(n)}$-statistically close.

- $\{x \leftarrow \{0,1\}^n : (x, f(x))\}$
- $\{x \leftarrow \{0,1\}^n; y = f(x) : (\mathsf{Inv}(y), y))\}$

In this case, we will say that Inv inverts f *with* $\frac{1}{p(n)}$*-statistical closeness.* We now proceed to show how to use such an inverter to prove that BM's round-collapse transformation works on $\mathcal{C}$. To simplify notation, we will make the following assumptions that are without loss of generality:

- $\mathcal{C}$ has at least 4 communication rounds and $\mathcal{C}$ sends the first message; we can always add an initial dummy message to achieve this, while only increasing the number of round by 1. We will then construct a new puzzle that has $k(\cdot) - 2$ rounds (which concludes the theorem). Since, in any puzzle, $\mathcal{A}$ sends the final message, this implies we can assume $k(\cdot)$ is even. To make our notations easier to read, we show how to reduce a $2k(\cdot)$-round protocol to a $2k(\cdot)-2$ rounds.
- There exists polynomials ℓ_c, ℓ_a such that all messages from $\mathcal{C}(1^n)$ are of (the same) length $\ell_c(n)$ and all the messages from $\mathcal{A}(1^n)$ need to be of length $\ell_a(n)$ (or else $\mathcal{C}$ rejects). Furthemore, $\ell_a(\cdot)$ and $\ell_c(\cdot)$ are polynomial-time computable, and strictly increasing.

We denote by $\mathcal{C}(1^n, p_1, p_2, \ldots, p_i; r_{\mathcal{C}})$ the $(i+1)^{st}$-message (i.e., the message to be sent in round $2i+1$ round) from $\mathcal{C}$ where $r_{\mathcal{C}}$ is $\mathcal{C}$'s randomness randomness and $p_1, p_2, \ldots, p_i$ are bit strings (representing the messages received from $\mathcal{A}$ in the first $2i$ rounds). Let $m(n)$ be $(\ell_a(n)+4)(\log(n))^2$ rounded upwards to the next power of two.[19] We will show that the BM transformation works (if ioOWF do not exist), when using $m(\cdot)$ repetitions. More precisely, consider the following $(2k(\cdot)-2)$-round puzzle challenger $\widetilde{\mathcal{C}}$ that on input $(1^n, p_1, \ldots, p_i; r_{\widetilde{\mathcal{C}}})$ proceeds as follows:

1. If $i < k(n) - 2$, output r_{i+1} (i.e., proceed just like $\mathcal{C}$ before round $2k(n) - 3$);
2. If $i = k(n) - 2$, output $(r_{k(n)-1}, r^1_{k(n)}, \ldots, r^{m(n)}_{k(n)})$ (i.e., in round $2k(n) - 3$, send the original challenge for round $2k(n) - 3$ as well as a "$m(n)$-wise parallel-repetition" challenge for the original round $2k(n) - 1$);

[19] We round to the next power of 2 to make it easy to sample a random number in $[m(n)]$; this is just to simplify presentation/analysis

3. If $i = k(n)-1$ (i.e., after receiving the message in the last round), output 1 if and only if

$$\mathcal{C}(1^n, p_1, p_2, \ldots, p_{k(n)-1}, p^i_{k(n)}; r_1, r_2, \ldots, r_{k(n)-1}, r^i_{k(n)}) = 1$$

for every $i \in [m(n)]$ (i.e., all the parallel instances are accepting),

where $r_{\widetilde{\mathcal{C}}}$ is interpreted as $(r_1, r_2, \ldots, r_{k(n)-1}, r^1_{k(n)}, \ldots, r^{m(n)}_{k(n)})$

We will show that $\widetilde{\mathcal{C}}$ is a $(99/100, 1/2)$-puzzle, and thus by Remark 3.1 this implies a puzzle with the same number of rounds.

We first define some notation:

- Given a transcript $T = (r_1, p_1, \ldots, p_{k(n)-2}, r_{k(n)-1}, r^1_{k(n)}, \ldots, r^{m(n)}_{k(n)}, p_{k(n)-1}, p^1_{k(n)}, \ldots, p^{m(n)}_{k(n)})$ of an interaction between $\widetilde{\mathcal{C}}$ and an adversary, we let $T_{\leq k-1} = (r_1, p_1, \ldots, p_{k(n)-2}, r_{k(n)-1})$ denote the transcript up to and including the round where $\mathcal{C}$ (in the emulation done by $\widetilde{\mathcal{C}}$) sends it $(k(n)-1)$'st message.
- We say that T is *accepting* if

$$\widetilde{\mathcal{C}}(p_1, \ldots, p_{k(n)-2}, p_{k(n)-1}, p^1_{k(n)}, \ldots, p^{m(n)}_{k(n)}; r_1, \ldots r_{k(n)-1}, r^1_{k(n)}, \ldots, r^{m(n)}_{k(n)}) = 1$$

(i.e,. if $\widetilde{\mathcal{C}}$ is accepting in the transcript).

Completeness: Completeness (in fact with all but negligible probability) follows directly from original proof by Babai-Moran [BM88].

Soundness: Assume for contradiction that there exists a PPT algorithm $\mathcal{A}^*$ that convinces $\widetilde{\mathcal{C}}$ on common input 1^n with probability $\epsilon(n)$ such that $\epsilon(n) > \frac{1}{2}$ for all sufficiently large n. Let $h(\cdot)$ be a polynomial such that $\mathcal{A}^*$ runs in time at most $h(n)$ when its first input is 1^n. We assume without loss of generality that A^* only sends a real last message if $\widetilde{\mathcal{C}}$ will be accepting it (note that since $\widetilde{\mathcal{C}}$ is public coin, A^* can verify this, so it is without loss of generality), and otherwise sends $\perp$ as its last message.

On a high-level, using $\mathcal{A}^*$ and the fact that polynomial-time computable functions are "invertible", we will construct a PPT B such that $\Pr[\langle B, \mathcal{C}\rangle(1^n) = 1] \geq \frac{1}{64}$ for sufficiently large n, which contradicts the soundness of the original $2k(n)$-round puzzle $\mathcal{C}$. Towards constructing B, we first define a polynomial-time algorithm M on which we will apply the inverter Inv. As described in the introduction, we will consider an algorithm M that operates on inputs of the form $(1^n, i, r_M)$ where i is an index of one of the $m(n)$ parallel sessions and r_M contains the randomness of $\mathcal{A}$ and $\widetilde{\mathcal{C}}$. To correctly parse such inputs, let $\ell_M(n) = n + \log(m(n)) + h(n) + \ell_c(n)\cdot(k(n)-1+m(n))$ and note that by our assumption on $\ell_c(n)$ and $\ell_a(n)$, this is a strictly increasing and polynomial-time computable function. In the rest of the proof, whenever the security parameter n is clear from context, we omit it and let $k = k(n), \ell_c = \ell_c(n), \ell_a = \ell_a(n), \epsilon = \epsilon(n), m = m(n)$ and $h = h(n)$. Now, consider the machine M that on input u internally incorporates the code of $\mathcal{A}^*$ and proceeds as follows:

1. M finds an n such that $\ell_M(n) = |u|$ (simply by enumerating different n from 1 up to $|u|$). If no such n exists, then M outputs $\perp$ and halts. Otherwise, M interprets u as (pad, i, r_M) such that $|pad| = n$, $|i| = \log(m(n))$, $r_M = (z, r_1, r_2, \ldots, r_{k-1}, r^1_k, \ldots, r^m_k)$, $z \in \{0,1\}^h$, and all the strings $r_1, r_2, \ldots, r_{k-1}, r^1_k, \ldots, r^m_k$ are in $\{0,1\}^{\ell_c}$.

2. It internally emulates an execution between $\mathcal{A}^*$ and $\widetilde{\mathcal{C}}$ on common input 1^n and respectively using randomness z and $(r_1, r_2, \ldots, r_{k-1}, r_k^1, \ldots, r_k^m)$. Let

$$T = (r_1, p_1, \ldots, p_{k-2}, r_{k-1}, r_k^1, \ldots, r_k^m, p_{k-1}, p_k^1, \ldots, p_k^m)$$

denote the transcript of the interaction.

3. If T is accepting, then M outputs

$$(1^{|pad|}, T_{\leq k-1}, p_{k-1}, r_k^i),$$

and otherwise $\perp$.

Let Inv be an inverter for M with $\frac{1}{n}$ statistical-closeness for all sufficiently large n—such an inverter exists due to our assumption on the non-existence of ioOWFs.

We are now ready to describe our adversary B for the $2k$-round puzzle. B on input $(1^n, r_1, r_2, \ldots, r_i; r_B)$ proceeds as follows:

1. B interprets r_B as $(z, pad, s_{k-1}^1, \ldots, s_{k-1}^m)$ such that $z \in \{0,1\}^h, pad \in \{0,1\}^n$ and all the strings $s_{k-1}^1, \ldots, s_{k-1}^m$ are in $\{0,1\}^{\ell_c}$.
2. If $i < k-1$, B outputs $p_i = \mathcal{A}^*(1^n, r_1, r_2, \ldots, r_i; z)$ (i.e., B proceeds just as $\mathcal{A}^*$ in the first $2k-4$ rounds).
3. If $i = k-1$ (i.e., in round $2k-2$), B lets $(p_{k-1}, p_k^1, \ldots, p_k^m) = \mathcal{A}^*(1^n, r_1, r_2, \ldots, r_{k-1}, s_k^1, \ldots, s_k^m; z)$ and outputs p_{k-1}.
4. If $i = k$ (i.e., in round $2k$), then:
   - B lets $T = (r_1, p_1, \ldots, p_{k-2}, r_{k-1}, s_k^1, \ldots, s_k^m, p_{k-1}, p_k^1, \ldots, p_k^m)$, and lets $y = (pad, T_{\leq k-1}, p_{k-1}, r_k)$ if T is accepting and $y = \perp$ otherwise.
   - B lets $u \leftarrow \mathsf{Inv}(y)$ and interprets u as (pad, j, r_M) where $|pad| = n$, $|j| = \log_2(m)$, and $r_M = (z', t_1, t_2, \ldots, t_{k-1}, t_k^1, \ldots, t_k^m)$, such that $z' \in \{0,1\}^h$ and $t_1, t_2, \ldots, t_{k-1}, t_k^1, \ldots, t_k^m$ are in $\{0,1\}^{\ell_c}$.
   - B next lets $(q_{k-1}, q_k^1, \ldots, q_k^m) = \mathcal{A}^*(1^n, r_1, r_2, \ldots, r_{k-1}, t_k^1, \ldots, t_k^m; z')$ and outputs q_k^j.

We now proceed to analyze the success probability of B against $\mathcal{C}$. In particular, we shall show that for all sufficiently large n, $\Pr[\langle B, \mathcal{C}\rangle(1^n) = 1] > \frac{1}{64}$ which will conclude the proof of Lemma 4.1. We denote by $\mathbf{View}_{\mathcal{A}^*}(\langle \mathcal{A}^*, \widetilde{\mathcal{C}}\rangle(1^n))$ the random variable that represents the view of the adversary $\mathcal{A}^*$ in an interaction with $\widetilde{\mathcal{C}}$ on common input 1^n—for convenience, we describe this view $v = (z, T)$ by $\mathcal{A}^*$'s random coin tosses z, as well as the transcript T of the interaction between $\mathcal{A}^*$ and $\widetilde{\mathcal{C}}$.[20] Towards analyzing B, we consider a sequence of hybrid experiments $\mathbf{Expt}_0, \mathbf{Expt}_1, \mathbf{Expt}_2, \mathbf{Expt}_3$—formally, an experiment defines a probability space and a probability density function over it. All experiments will be defined over the same probability space so we can consider the same random variables over all of them. To simplify notation, we abuse of notation and let $\mathbf{Expt}_i(n)$ also denote a random variable describing the output of the experiment $\mathbf{Expt}_i(n)$.

$\mathbf{Expt}_0(n)$ will simply consider an execution between B^{Inv} and $\mathcal{C}$ on common input 1^n and will output 1 if $\mathcal{C}$ is accepting and 0 otherwise; see Figure 1 for a formalization. To simplify the transition

[20] This is a bit redundant—as the messages sent by $\mathcal{A}^*$ can of course be recomputed given just the randomness of $\mathcal{A}^*$ and the messages from $\widetilde{\mathcal{C}}$, but will simplify notation.

to later experiments, we formalize $\mathbf{Expt}_0(n)$ as first sampling a full transcript T of an execution between $\mathcal{A}^*$ and $\widetilde{\mathcal{C}}$, keeping only the prefix $T_{\leq k-1}$ (this gives exactly the same distribution as an interaction between B and $\mathcal{C}$ up to round $2k-2$), next sampling an "external" random message r (just as $\mathcal{C}$ would in round $2k-1$), and finally producing the last message just as B does. (We additionally sample a random index $i \in [m]$ is not used in the current experiment, but will be useful in later experiments.) We thus directly have:

Claim 1. $\Pr[\langle B, \mathcal{C}\rangle(1^n) = 1] = \Pr[\mathbf{Expt}_0(n) = 1]$

We now slowly transform the experiment into one that becomes easy to analyze. See Figure 1 for a formal description of the experiments.

1. We first define an a "good" event $G = W \cap G'$, where W is the event that the originally sampled transcript is accepting and G' is the event that the "prefix" $(T_{\leq k-1, p_{k-1}})$ is "good" in a well defined sense (roughly speaking, that continuations conditioned $T_{\leq k-1}$ are successful with high probability, and that that in such successful continuations p_{k-1} is used with not "too low" probability). $\mathbf{Expt}_1$ will next proceeds just like $\mathbf{Expt}_0$ except that we additionally fail if the event G does not happen. We thus have that the probability of $\mathsf{Expt}_0(n)$ outputting 1 is at least as high as the probability of $\mathsf{Expt}_1(n)$ outputting 1.

   Claim 2. $\Pr[\mathbf{Expt}_0(n) = 1] \geq \Pr[\mathbf{Expt}_1(n) = 1]$.

   Additionally, as we shall show (using an averaging argument), the event G happens with non-negligible probability, not just in $\mathbf{Expt}_1$ but also in all the other experiments $\mathbf{Expt}_j$ for $j \in \{1, 2, 3\}$ (as Step 1 of the experiment remains unchanged in all of them).

   Claim 3. *For $j \in \{1, 2, 3\}$, $\Pr[G] \geq \frac{\epsilon^2}{8}$, where the probability is over the randomness in experiment $\mathbf{Expt}_j(n)$*

2. We next transition to an experiment $\mathbf{Expt}_2$ where instead of choosing the message r_k at random (as it was in $\mathbf{Expt}_1$), we select it as the message in the i'th repetition of $\widetilde{\mathcal{C}}$'s $k-1$'st message in the initially sampled transcript T, where i is a randomly sampled index $i \in [m]$. The reason for defining this experiment is that, in it, we are applying the one-way function inverter on the "right" distribution (just as in the definition of M). The central claim to show is that this change does not change the success probability by too much. As discussed in the introduction, we shall prove it using Raz's sampling lemma.

   Claim 4. $\Pr[\mathbf{Expt}_1(n) = 1] \geq \Pr[\mathbf{Expt}_2(n) = 1] - \frac{1}{\log(n)}$.

3. Finally, we transition to an experiment $\mathbf{Expt}_2$ where we employ a *perfect inverter* PInv—that always samples uniform preimages to M, instead of the (imperfect) inverter Inv. It directly follows from the fact that Inv is an inverter with statistical closeness $\frac{1}{n}$ and that the inverter is applied to an element that is sampled as a uniform image of M[21] that the statistical distance between $\mathbf{Expt}_2(n)$ and $\mathbf{Expt}_3(n)$ is bounded by $\frac{1}{n}$ for sufficiently large n. In particular,

   Claim 5. *For all sufficiently large n,*

$$\Pr[\mathbf{Expt}_2(n) = 1] \geq \Pr[\mathbf{Expt}_3(n) = 1] - \frac{1}{n}$$

[21] Note that we here rely on the fact that $y = \perp$ when T is not accepting.

Experiment $\mathbf{Expt}_0(n)$.

1. Sample $(z, T) \leftarrow \mathbf{View}_{\mathcal{A}^*}(\langle \mathcal{A}^*, \widetilde{\mathcal{C}} \rangle (1^n)); pad \leftarrow \{0,1\}^n; i \leftarrow [m]; r \leftarrow \{0,1\}^{\ell_c}$. Interpret T as $(r_1, p_1, \ldots, p_{k-2}, r_{k-1}, s_k^1, \ldots, s_k^m, p_{k-1}, p_k^1, \ldots, p_k^m)$.
2. Let $r_k = r$ and let $y = (pad, T_{\leq k-1}, p_{k-1}, r_k)$ if T is accepting and $y = \perp$ otherwise.
3. Let $u \leftarrow \mathsf{Inv}(y)$; interpret u as (pad, j, r_M) where $|pad| = n$, $|j| = \log_2(m)$, and $r_M = (z', t_1, t_2, \ldots, t_{k-1}, t_k^1, \ldots, t_k^m)$ just as B does and let $(q_{k-1}, q_k^1, \ldots, q_k^m) = \mathcal{A}^*(1^n, r_1, r_2, \ldots, r_{k-1}, t_k^1, \ldots, t_k^m; z')$.
4. Output 1 iff $T' = (T_{\leq k-1}, p_{k-1}, r_k, q_k^j)$ is accepting, and 0 otherwise.

Experiment $\mathsf{Expt}_1(n)$.

1. Sample $(z, T) \leftarrow \mathbf{View}_{\mathcal{A}^*}(\langle \mathcal{A}^*, \widetilde{\mathcal{C}} \rangle (1^n)); pad \leftarrow \{0,1\}^n; i \leftarrow [m]; r \leftarrow \{0,1\}^{\ell_c}$. Interpret T as $(r_1, p_1, \ldots, p_{k-2}, r_{k-1}, s_k^1, \ldots, s_k^m, p_{k-1}, p_k^1, \ldots, p_k^m)$.
2. Let $r_k = r$ and let $y = (pad, T_{\leq k-1}, p_{k-1}, r_k)$ if T is accepting and $y = \perp$ otherwise.
3. Let $u \leftarrow \mathsf{Inv}(y)$; interpret u as (pad, j, r_M) where $|pad| = n$, $|j| = \log_2(m)$, and $r_M = (z', t_1, t_2, \ldots, t_{k-1}, t_k^1, \ldots, t_k^m)$ just as B does and let $(q_{k-1}, q_k^1, \ldots, q_k^m) = \mathcal{A}^*(1^n, r_1, r_2, \ldots, r_{k-1}, t_k^1, \ldots, t_k^m; z')$.
4. Output 1 iff $T' = (T_{\leq k-1}, p_{k-1}, r_k, q_k^j)$ is accepting and G holds, and 0 otherwise.

Distribution Expt_2^n

1. Sample $(z, T) \leftarrow \mathbf{View}_{\mathcal{A}^*}(\langle \mathcal{A}^*, \widetilde{\mathcal{C}} \rangle (1^n)); pad \leftarrow \{0,1\}^n; i \leftarrow [m]; r \leftarrow \{0,1\}^{\ell_c}$. Interpret T as $(r_1, p_1, \ldots, p_{k-2}, r_{k-1}, s_k^1, \ldots, s_k^m, p_{k-1}, p_k^1, \ldots, p_k^m)$.
2. Let $r_k = s_k^i$ and let $y = (pad, T_{\leq k-1}, p_{k-1}, r_k)$ if T is accepting and $y = \perp$ otherwise.
3. Let $u \leftarrow \mathsf{Inv}(y)$; interpret u as (pad, j, r_M) where $|pad| = n$, $|j| = \log_2(m)$, and $r_M = (z', t_1, t_2, \ldots, t_{k-1}, t_k^1, \ldots, t_k^m)$ just as B does and let $(q_{k-1}, q_k^1, \ldots, q_k^m) = \mathcal{A}^*(1^n, r_1, r_2, \ldots, r_{k-1}, t_k^1, \ldots, t_k^m; z')$.
4. Output 1 iff $T' = (T_{\leq k-1}, p_{k-1}, r_k, q_k^j)$ is accepting and G holds, and 0 otherwise.

Distribution Expt_3^n

1. Sample $(z, T) \leftarrow \mathbf{View}_{\mathcal{A}^*}(\langle \mathcal{A}^*, \widetilde{\mathcal{C}} \rangle (1^n)); pad \leftarrow \{0,1\}^n; i \leftarrow [m]; r \leftarrow \{0,1\}^{\ell_c}$. Interpret T as $(r_1, p_1, \ldots, p_{k-2}, r_{k-1}, s_k^1, \ldots, s_k^m, p_{k-1}, p_k^1, \ldots, p_k^m)$.
2. Let $r_k = s_k^i$ and let $y = (pad, T_{\leq k-1}, p_{k-1}, r_k)$ if T is accepting and $y = \perp$ otherwise.
3. Let $u \leftarrow \mathsf{PInv}(y)$; interpret u as (pad, j, r_M) where $|pad| = n$, $|j| = \log_2(m)$, and $r_M = (z', t_1, t_2, \ldots, t_{k-1}, t_k^1, \ldots, t_k^m)$ just as B does and let $(q_{k-1}, q_k^1, \ldots, q_k^m) = \mathcal{A}^*(1^n, r_1, r_2, \ldots, r_{k-1}, t_k^1, \ldots, t_k^m; z')$.
4. Output 1 iff $T' = (T_{\leq k-1}, p_{k-1}, r_k, q_k^j)$ is accepting and G holds, and 0 otherwise.

Figure 1: Description of intermediate experiments.

4. We finally note that in $\mathbf{Expt}_3$, there are only two reason the experiment can output 0: (1) The originally sampled transcript T is not accepting (i.e., the event W does not hold); if is is accepting, the perfect inverter will make sure that T' is also accepting, or (2) the event G does not hold. Additionally note, since G is defined as $W \cap G'$, we have that whenever G holds, W holds as well and thus the experiment must output 1. Thus, by Claim 3, we have:

   Claim 6.
$$\Pr[\mathbf{Expt}_3(n) = 1] \geq \frac{\epsilon^2}{8}$$

5. By combining claims 1, 2, 4, 6, we have that for all sufficiently large n,
$$\Pr[\langle B, \mathcal{C}\rangle(1^n) = 1] = \Pr[\mathbf{Expt}_0(n) = 1] \geq \Pr[\mathbf{Expt}_1(n) = 1] \geq \Pr[\mathbf{Expt}_2(n) = 1] - \frac{1}{\log(n)}$$
$$\geq \Pr[\mathbf{Expt}_3(n) = 1] - \frac{2}{\log(n)} \geq \frac{\epsilon^2}{8} - \frac{2}{\log(n)} > \frac{1}{64}$$

   which is a contradiction.

To to conclude the proof of Lemma 4.1, it just remains to formalize the event $G = W \cap G'$ and proving Claim 3 and Claim 5.

4.2 The Good Event G and the Proof of Claim 3

We begin by defining some random variables over the probability space over which $\mathbf{Expt}_1$ is defined. Note that the probability space is the same for $\mathbf{Expt}_0, \mathbf{Expt}_1, \mathbf{Expt}_2$ and as such random variables and events over $\mathbf{Expt}_1$ are also defined over all the other experiments. We use boldface to denote random variables describing the outcome of variables in the experiments—for instance, we let $\mathbf{T}$ denote a random variable describing the value of T as sampled in the experiments.

Let W denote the event that $\mathbf{T}$ is accepting (i.e., the transcript sampled in Step 1 is accepting) and let Θ be the set of partial transcripts θ such that
$$\Pr[W \mid \mathbf{T}_{\leq(k-1)} = \theta] \geq \frac{\epsilon}{2}.$$
where the probability is over $\mathbf{Expt}_1(n)$. That is, Θ is the set of "good" partial transcripts conditioned on which $\mathcal{A}^*$ has a reasonable probability of succeeding. Note that by a standard averaging argument, we have that such transcripts occur often:
$$\Pr[\mathbf{T}_{\leq(k-1)} \in \Theta] \geq \frac{\epsilon}{2}. \tag{1}$$
Now, consider the event W_p that W holds and $\mathbf{p}_{k-1} = p$; let $P(\theta)$ be the set of messages $p \in \{0,1\}^{\ell_a}$ for which
$$\Pr[W_p \mid \mathbf{T}_{\leq(k-1)} = \theta] \geq \frac{\epsilon}{2^{\ell_a+2}}.$$
In other words, $P(\theta)$ is the set of "good" (adversary) messages p such that conditioned on the partial transcript θ, the probability that $\mathcal{A}^*$ succeeds while using p as its $k-1$'st message is greater than $\frac{\epsilon}{2^{\ell_a+2}}$. As we shall now show using another (standard) averaging argument, for every $\theta \in \Theta$, we have
$$\Pr[\mathbf{p}_{k-1} \in P(\theta) \mid \mathbf{T}_{\leq(k-1)} = \theta] \geq \frac{\epsilon}{4}. \tag{2}$$

Suppose for contradiction that for some $\theta \in \Theta$, Equation 2 does not hold. Then, we have

$$\begin{aligned}
\Pr[W \mid \mathbf{T}_{\leq (k-1)} = \theta] &= \sum_{p \in \{0,1\}^{\ell_a}} \Pr[W_p \mid \mathbf{T}_{\leq (k-1)} = \theta] \\
&= \sum_{p \in P(\theta)} \Pr[W_p \mid \mathbf{T}_{\leq (k-1)} = \theta] + \sum_{p \in \{0,1\}^{\ell_a} - P(\theta)} \Pr[W_p \mid \mathbf{T}_{\leq (k-1)} = \theta] \\
&\leq \Pr[\mathbf{p}_{k-1} \in P(\theta) \mid \mathbf{T}_{\leq (k-1)} = \theta] + \sum_{p \in \{0,1\}^{\ell_a} - P(\theta)} \Pr[W_p \mid \mathbf{T}_{\leq (k-1)} = \theta] \\
&< \frac{\epsilon}{4} + \sum_{p \in \{0,1\}^{\ell_a} - P(\theta)} \Pr[W_p \mid \mathbf{T}_{\leq (k-1)} = \theta] \\
&\leq \frac{\epsilon}{4} + \sum_{p \in \{0,1\}^{\ell_a} - P(\theta)} \frac{\epsilon}{2^{\ell_a + 2}} \\
&\leq \frac{\epsilon}{4} + 2^{\ell_a} \cdot \frac{\epsilon}{2^{\ell_a + 2}} = \frac{\epsilon}{2}
\end{aligned}$$

which contradicts that $\theta \in \Theta$.

Next, define G' to be the event that $\mathbf{T}_{\leq (k-1)} \in \Theta$ and $\mathbf{p}_{k-1} \in P(\mathbf{T}_{\leq (k-1)})$, and define G as holding when W and G' both hold (i.e., the originally sampled transcript is accepting and G' holds). Note that G' in fact implies that W holds (since $\mathbf{p}_{k-1} \in P(\mathbf{T}_{\leq (k-1)})$ implies that $\mathbf{p}_{k-1} \neq \perp$ which by our assumption on A^* means that $\mathbf{T}$ must be accepting), thus in fact $G' = G$. By combing Equations 1 and 2, we have:

$$\begin{aligned}
\Pr[G] = \Pr[G'] &= \Pr[\mathbf{T}_{\leq (k-1)} \in \Theta \wedge \mathbf{p}_{k-1} \in P(\mathbf{T}_{\leq (k-1)})] \\
&= \Pr[\mathbf{T}_{\leq (k-1)} \in \Theta] \times \Pr[\mathbf{p}_{k-1} \in P(\mathbf{T}_{\leq (k-1)}) \mid \mathbf{T}_{\leq (k-1)} \in \Theta] \\
&\geq \frac{\epsilon}{2} \times \frac{\epsilon}{4} = \frac{\epsilon^2}{8}
\end{aligned}$$

where the probability is taken over $\mathbf{Expt}_1(n)$. Finally, note that since Step 1 (whose outcome determines whether G happens) remains unchanged in all the experiments, we can conclude that $\Pr[G] \geq \frac{\epsilon^2}{8}$ where the probability is taken over $\mathbf{Expt}_j(n)$ for every $j \in \{1, 2, 3\}$, which concludes the proof of Claim 3.

4.3 Proof of Claim 4

Recall that we need to show that $\Pr[\mathbf{Expt}_1(n) = 1] \geq \Pr[\mathbf{Expt}_2(n) = 1] - \frac{1}{\log(n)}$. Observe that the only difference between experiments $\mathbf{Expt}_1$ and $\mathbf{Expt}_2$ is that, in $\mathbf{Expt}_1$, we set $r_k = r$ and in $\mathbf{Expt}_2$, we set $r_k = s_k^i$. Furthermore, both the experiments sample $((z, T), pad, i, r)$ from the same distributions and output 0 whenever G does not hold (which is a function only of T). It follows that the statistical distance between $\mathbf{Expt}_1(n)$ and $\mathbf{Expt}_2(n)$ is bounded by the statistical distance of $\mathbf{Expt}_1(n)$ and $\mathbf{Expt}_2(n)$ conditioned on the event G. Note that we can rephrase the event G as

$$G = \bigcup_{\theta \in \Theta, p \in P(\theta)} W_p \cap (\mathbf{T}_{\leq k-1} = \theta)$$

Below, we shall show that for every $\theta \in \Theta, p \in P(\theta)$, it holds that the statistical distance between $\{\mathbf{Expt}_1(n) \mid \mathbf{T}_{\leq k-1} = \theta, W_p\}$ and $\{\mathbf{Expt}_2(n) | \mathbf{T}_{\leq k-1} = \theta, W_p\}$ is bounded by $\frac{1}{\log(n)}$, which concludes the proof of Claim 5.

Consider some $\theta \in \Theta, p \in P(\theta)$ and consider the experiments $\{\mathbf{Expt}_1(n) \mid \mathbf{T}_{\leq k-1} = \theta, W_p\}$ and $\{\mathbf{Expt}_2(n) \mid \mathbf{T}_{\leq k-1} = \theta, W_p\}$. Note both experiments proceed exactly the same after r_k is defined in Step 2, so we can ignore everything that happens after this. Additionally, note that the only variables that are relevant after this point are $\mathbf{pad}, \mathbf{T}_{\leq k-1}, \mathbf{p}_{k-1}$, $\mathbf{i}$ and $\mathbf{r}$. Note that $\mathbf{pad}, \mathbf{i}$ are both independent of the events $\mathbf{T}_{\leq k-1} = \theta, W_p$ and thus still independently and uniformly sampled in both experiments. $\mathbf{T}_{\leq k-1}$ and $\mathbf{p}_{k-1}$, on the other hand are fixed (constant) conditioned on $\mathbf{T}_{\leq k-1} = \theta, W_p$. Thus, to bound the statistical difference between $\{\mathbf{Expt}_1(n) \mid \mathbf{T}_{\leq k-1} = \theta, W_p\}$ and $\{\mathbf{Expt}_1(n) \mid \mathbf{T}_{\leq k-1} = \theta, W_p\}$, it suffices to bound the statistical distance between $\mathbf{r_k}$ in $\{\mathbf{Expt}_1(n) \mid \mathbf{T}_{\leq k-1} = \theta, W_p\}$ and $\mathbf{r_k}$ in $\{\mathbf{Expt}_2(n) \mid \mathbf{T}_{\leq k-1} = \theta, W_p\}$. In other words, we need to upper bound,

$$\Delta = \mathsf{SD}(\mathbf{r}, \mathbf{s}_k^{\mathbf{i}} \mid W_p) = \mathsf{SD}(\mathbf{s}_k^{\mathbf{i}}, \mathbf{s}_k^{\mathbf{i}} \mid W_p) \leq \sum_{j \in [m]} \frac{1}{m} \mathsf{SD}(\mathbf{s}_k^j, \mathbf{s}_k^j \mid W_p)$$

over $\{\mathbf{Expt}_2(n) \mid \mathbf{T}_{\leq k-1} = \theta\}$ since for each j, $\mathbf{s}_k^j$ is sampled uniformly at random, independent of $T_{\leq k-1}$ (and independent of $\mathbf{s}_k^{j'}$ for $j' \neq j$). Towards bounding this quantity, we will rely on Raz's sampling lemma.

Lemma 4.2 ([Raz98])**.** *Let $\mathbf{X}_1, \ldots, \mathbf{X}_,$ be independent random variables on a finite domain U. Let E be an event over $\vec{\mathbf{X}} = (\mathbf{X}_1, \ldots, \mathbf{X}_m)$. Then,*

$$\frac{1}{m} \cdot \sum_{i=1}^{m} \mathsf{SD}(\mathbf{X}_i, \mathbf{X}_i | E) \leq \sqrt{\frac{1}{m} \cdot \log \frac{1}{\Pr[E]}}$$

By applying Raz's lemma, we directly get that

$$\Delta \leq \sqrt{\frac{1}{m} \cdot \log \frac{1}{\Pr[W_p]}}$$

where the probability is over $\{\mathbf{Expt}_2(n) \mid \mathbf{T}_{\leq k-1} = \theta\}$. Since by our assumption $p \in P(\theta)$, we have that the probability of W_p conditioned on $\mathbf{T}_{\leq k-1} = \theta$ is at least $\frac{\epsilon}{2^{\ell_a+2}}$, thus

$$\Delta \leq \sqrt{\frac{1}{m} \cdot (\ell_a + 2 - \log(\epsilon))} \leq \frac{1}{\log(n)}$$

since $\epsilon > \frac{1}{2}$ and $m = (\ell_a + 4)(\log(n))^2 > (\ell_a + 2 - \log(\epsilon))(\log(n))^2$.

■

4.4 Variations

Using essentially the same proofs, we can directly get the following vacations of 4.1. The first variant simply states that the same result holds for almost-everywhere puzzles.

Lemma 4.3 (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 4.4 (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 4.5 (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.*[22]

4.5 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 4.1. *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 $\mathsf{NP/poly}$ problem) that is HOA (resp. nuHOA).*

Proof: If (non-uniform) ioOWF exists, then by applying Proposition 3.2 we have that 2-round (non-uniform) public-coin puzzles exist. If (non-uniform) ioOWF do not exist, we can apply Lemma 4.1 (Lemma 4.5) iteratively to collapse any constant-round protocol to a 2-round protocol. (Note that we can only apply Lemma 4.1 a constant number of times, as the communication complexity of the resulting protocol grows polynomially with each application.). Thus in either case, we conclude that the existence of a $O(1)$-round (non-uniform) public-coin puzzle implies a 2-round (non-uniform) public-coin puzzle. The corollary is concluded by applying Lemma 3.4. ■

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.

5 Characterizing Polynomial-round Puzzles

We observe that the existence of a poly-round public-coin puzzle is equivalent to the statement that $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$. A consequence of this result (combined with Lemma 3.4) 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 $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$ (which would be highly unexpected).

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

Theorem 5.1. *For every $\epsilon > 0$, there exists an n^ϵ-round public-coin puzzle (resp. a non-uniform puzzle) if and only if* $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$ *(resp.* $\mathsf{PSPACE} \not\subseteq \mathsf{P/poly}$*).*

Proof: For the "only-if" direction, note that using the same proof as (the easy direction) in $\mathsf{IP} = \mathsf{PSPACE}$ [Sha92, LFKN92], we can use a PSPACE oracle to implement the optimal adversary strategy in every puzzle and thus (due to the completeness condition of the puzzle) break the soundness of every puzzle using a PSPACE oracle. So, if $\mathsf{PSPACE} \subseteq \mathsf{BPP}$, soundness of every puzzle can be broken in PPT and thus puzzles cannot exist. (We remark that a very similar statement—in the language of non-trivial interactive arguments—was already observed by Goldreich [Gol18]; see Section 7 for more details.)

For the "if" direction, recall that by the classic result of [BFNW93] (see also [TV07]), if $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$, then there is PSPACE language L', constant $c \in \mathbb{N}$, and a polynomial $p(\cdot)$ such that $(L', \mathcal{U}_p)$ is $\frac{1}{n^c}$-HOA. We will now use this HAO language L' together with the fact that by [Sha92, LFKN92] all of PSPACE has a public-coin interactive proof (and the fact that PSPACE is closed under complement, to get a puzzle. The puzzle challenger $\mathcal{C}(1^n)$ simply samples a random statement $x \in \{0,1\}^{p(n)}$ and sends it to the adversary. The adversary next announces a bit b (determining whether $x \in L'$ or not) and next if $b = 1$, $\mathcal{C}$ runs the IP verifier for $x \in L'$ and if $b = 0$ instead runs the IP verifier for $x \notin L'$. Due to [Sha92, LFKN92], we may assume without loss of generality that the IP has completeness 1 and soundness error 2^{-n}. As we shall now argue $\mathcal{C}$ is a $(1, 1 - \frac{2}{n^c})$-puzzle which by remark Remark 3.1 implies a puzzle. Completeness follows directly from the completeness of the IP. For soundness, consider a PPT machine $\mathcal{A}^*$ that convinces $\mathcal{C}$ with probability better than $1 - \frac{2}{n^c}$. We construct a machine B that breaks the $\frac{1}{n^c}$-HAO property of L'. $B(1^n, x)$ simply emulates an interaction between $\mathcal{C}(1^n)$ and $\mathcal{A}^*$ while fixing $\mathcal{C}$'s first message to x and accepts x if $\mathcal{C}$ is accepting, and rejects otherwise. Since B is feeding $\mathcal{A}^*$ messages according to the same distribution as in the real execution (with $\mathcal{C}$), we have that $\mathcal{A}^*$ convinces $\mathcal{C}$ in the emulation by B with probability at least $1 - \frac{2}{n^c}$. By the soundness of the IP, we have that except with probability 2^{-n}, whenever the proof is accepting, the bit b must correctly decide x. We conclude (by a union bound) that B correctly decides x with probability $1 - \frac{2}{n^c} - 2^{-n} > 1 - \frac{1}{n^c}$ for all sufficiently large $n \in \mathbb{N}$.

The non-uniform version of the theorem follows using exactly the same proof. ■

6 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.

6.1 From Imperfect to Perfect Completeness (by Adding a Round)

Furer et al. [FGM+89] 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 6.1. *Suppose there exists 2-round public-coin puzzle. Then there exists a 3-round public-coin puzzle with perfect completeness.*

Proof: Let $\mathcal{C}$ be a 2-round public-coin puzzle. Let ℓ_c, ℓ_a be polynomials such that the message from $\mathcal{C}(1^n)$ is of length $\ell_c(n)$ and the message from $\mathcal{A}(1^n)$ is of length $\ell_a(n)$; we assume without loss of generality that $\ell_c(n) > 2$. When the security parameter n is clear from the context we will omit it and let $\ell_c(n) = \ell_c$ and $\ell_a(n) = \ell_a$.

We now apply the Furer et al. [FGM+89] transformation to this puzzle to create a 3-round puzzle $\widetilde{\mathcal{C}}$. The puzzle will proceed by first having the adversary sending ℓ_c "pads" $z_1, \ldots, z_{\ell_c} \in \ell_c$ to $\widetilde{\mathcal{C}}$; $\widetilde{\mathcal{C}}$ next sends back a random message $r_{\widetilde{\mathcal{C}}} \in \{0,1\}^{\ell_c}$, and the adversary is next supposed to find a response i, p such that $(r \oplus z_i, p)$ is a valid transcript for the original puzzle (i.e., the adversary needs to win in one of the parallel "padded" instances of the original puzzle). More formally, $\widetilde{\mathcal{C}}(1^n, (z_1, \ldots, z_{\ell_c}), (i,p); r_{\widetilde{\mathcal{C}}}) = 1$ if and only if $\mathcal{C}(1^n, p; r_{\widetilde{\mathcal{C}}} \oplus z_i)$ outputs 1. Perfect completeness of $\widetilde{\mathcal{C}}$ follows directly from the original proof by [FGM+89]. For completeness, we recall their proof. From the (imperfect) completeness of $\mathcal{C}$, we have that there exists some adversary $\mathcal{A}$ such that $\Pr[\langle \mathcal{A}, \mathcal{C}\rangle(1^n) = 1] \geq 1 - \frac{1}{n}$ for all sufficiently large n; without loss of A is deterministic. Fix some $n > 2$ for which this holds. Let $S \subseteq \{0,1\}^{\ell_c}$ be the set of challenges for which $\mathcal{A}$ provides an accepting response; the probability that a random challenge $z \in \{0,1\}^{\ell_c}$ is inside S is thus at least $1 - \frac{1}{n}$. We will show that there exists "pads" $z_1, \ldots, z_{\ell_c}$ such that for every $r \in \{0,1\}^{\ell_c}$, there exists some i such that $r \oplus z_i \in S$, which concludes that an unbounded attacker $\widetilde{\mathcal{A}}$ can succeed with probability 1 (by selecting those pads and next providing an accepting response). Note that for every fixed r, for a *randomly* chosen pad z_i, the probability that $r \oplus z_i \notin S$ is at most $\frac{1}{n}$; and thus the probability over randomly chosen pads $z_1, \ldots, z_{\ell_c}$ that $r \oplus z_i \notin S$ *for all* i is at most $\frac{1}{n^{\ell_c}}$. We conclude, by a union bound, that the probability over randomly chosen pads $z_1, \ldots, z_{\ell_c}$ that *there exists* some $r \in \{0,1\}^{\ell_c}$ such that $r \oplus z_i \notin S$ for all i is at most $\frac{2^{\ell_c}}{n^{\ell_c}} < 1$. Thus, there exists pads $z_1, \ldots, z_{\ell_c}$ such that *for every* $r \in \{0,1\}^{\ell_c}$ there exists some i such that $r \oplus z_i \notin S$, which concludes perfect completeness.

We now turn to proving computational soundness. Consider some adversary $\widetilde{\mathcal{A}}^*$ that succeeds in convincing $\widetilde{\mathcal{C}}$ with probability $\epsilon(n)$ for all $n \in \mathbb{N}$. We construct an adversary $\mathcal{A}^*$ that convinces $\mathcal{C}$ with probability $\frac{\epsilon(n)}{\ell_c}$, which is a contradiction. $\mathcal{A}^*(1^n)$ picks a random tape $r_{\widetilde{\mathcal{A}}^*}$ for $\widetilde{\mathcal{A}}^*$, lets $(z_1, \ldots, z_{\ell_c}) = \widetilde{\mathcal{A}}^*(1^n; r_{\widetilde{\mathcal{A}}^*})$, picks a random index $i \in [\ell_c]$ and outputs z_i. Upon receiving a "challenge" r, it lets $(j,p) = \widetilde{\mathcal{A}}^*(1^n, r \oplus z_i; r_{\widetilde{\mathcal{A}}^*})$ outputs p if $i = j$ and $\perp$ otherwise. First, note that in the emulation by $\mathcal{A}^*$, $\mathcal{A}^*$ feeds $\widetilde{\mathcal{A}}^*$ the same distribution of messages as $\widetilde{\mathcal{A}}^*$ would see in a "real" interaction with $\widetilde{\mathcal{C}}$; thus, we have that the (j,p) is an accepting message (w.r.t., the challenge $r \oplus z_i$) with probability ϵ. Additionally, since $r \oplus z_i$ information-theoretically hides i (as r is completely random), we have that the probability that $i = j$ is $\frac{1}{\ell_c}$ and furthermore, the event that this happens is independent of whether the message (j,p) is accepting. We conclude that $\mathcal{A}^*$ convinces $\mathcal{C}$ with probability $\frac{\epsilon(n)}{\ell_c}$, which concludes the soundness proof. ■

6.2 Promise-trueDistributional 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 6.2. *Suppose there exists* 2*-round public-coin puzzle. Then there exists a* 2*-round private-coin puzzle with perfect completeness.*

Proof: The theorem follows directly by applying our earlier proved results:

- By Theorem 6.1 (perfect completeness through adding a round), a 2-round public-coin puzzle implies a 3-round public-coin puzzle with perfect completeness.
- By Lemma 4.4 (round-collapse lemma), we conclude that either ioOWF exists, or there exists a 2-round public-coin puzzle with perfect completeness.
- As ioOWFs trivially imply a 2-round (private-coin) puzzle, the theorem follows.

■

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 3.3, aeHOA distributional NP problem implies a 2-round puzzle, we directly get the following corollary:

Corollary 6.3. *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".

6.3 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 6.4. *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* $\mathcal{R} \in \mathsf{TFNP}$ *and some* PPT $\mathcal{D}$ *such that* $(\mathcal{R}, \mathcal{D})$ *is SearchHAO.*

Proof: Again, the theorem follows by simply applying our earlier proved results:

- From Lemma 3.3, we have that an aeHOA distributional NP problem implies a 2-round almost-everywhere puzzle.
- By Theorem 6.1 (perfect completeness through adding a round), this implies a 3-round almost-everywhere puzzle with perfect completeness.
- Applying Lemma 4.4 (round-collapse, variant 2), we conclude that either one-way functions exists, or there exists a 2-round public-coin puzzle with perfect completeness.
- Finally, by applying Lemma 3.5, a 2-round *public-coin* puzzle with perfect completeness implies the existence of some $\mathcal{R} \in \mathsf{TFNP}$ and some PPT $\mathcal{D}$ such that $(\mathcal{R}, \mathcal{D})$ is SearchHAO.

■

By replacing the use of Lemma 4.4 with Lemma 4.3 (round-collapse, variant 1), we instead get the following variants.

Theorem 6.5. *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* $\mathcal{R} \in \mathsf{TFNP}$ *and some* PPT $\mathcal{D}$ *such that* $(\mathcal{R}, \mathcal{D})$ *is aeSearchHAO.*

7 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 [Gol18]. 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 7.1 (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 $\mathsf{NP}/\mathsf{poly}$ problem that is nuHAO.

Theorem 7.2. *Assume there exists a $O(1)$-round public-coin non-trivial argument for some language L. Then, there exists a distributional $\mathsf{NP}/\mathsf{poly}$ problem that is nuHOA.*

Proof: Consider some k-round non-trivial public-coin argument system (P, V). We show that this implies the existence of a k-round non-uniform puzzle. The theorem next follows by applying Corollary 4.1.

Since (P, V) is not a proof system, there must exist a some polynomial $p(\cdot)$, an unbounded prover B, and sequences $I = \{n_1, n_2, \ldots\}$ and $\{x_{n_i}\}_{i\in\mathbb{N}}$ such that for all $i \in \mathbb{N}$, $|x_{n_i}| = n_i, x_{n_i} \notin L$ yet B convinces V on common input x_{n_i} with probability $\frac{1}{p(n_i)}$.

Now consider the k-round non-uniform puzzle $\mathcal{C}$ that for each n receives $(1, x_n)$ as non-uniform advice if $n \in I$ and otherwise $(0, \perp)$. Given non-uniform advice (b, x), $\mathcal{C}(1^n)$ simply accepts if $b = 0$ and otherwise runs the verifier $V(x_n)$. We shall argue that $\mathcal{C}$ is a $(\frac{1}{p(n)}, \frac{1}{2p(n)})$ puzzle which by Remark 3.1 implies a puzzle. Completeness follows directly from the existence of B (when $b = 0$, we have completeness 1 and otherwise, we have completeness $\frac{1}{p(n)}$ by construction). To show soundness, notice that any non-uniform PPT adversary $\mathcal{A}^*$ that breaks soundness of the puzzle with probability $\frac{1}{2p(n)}$ for all sufficiently large n, must in particular break it for infinitely many $n \in I$, and as such breaks the soundness of (P, V) for infinitely many $x \in \{0, 1\}^* - L$ with probability $\frac{1}{2p(|x|)}$, which contradicts the soundness of (P, V). ■

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

Lemma 7.1. *Suppose there exists a distributional NP problem $(L', \mathcal{D})$ that is nuHOA . Then, for every language $L \in \mathsf{NP}$, there exists a non-trivial 2-round public-coin argument for L with an efficient prover.*

Proof: We first observe that by the same proof as for Lemma 3.3, a nuHOA NP problem implies a 2-round puzzle satisfying a "weak" completeness property, where completeness only holds for infinitely many $n \in \mathbb{N}$, but where soundness holds also against non-uniform PPT algorithms. (Recall that in the proof of Lemma 3.3, we only relied on the almost-everywhere HOA property of the NP problem to ensure that completeness held for *all* sufficiently large input lengths.) We next simply combine this "weakly-complete" puzzle with a standard NP proof for L to get a non-trivial 2-round argument for L. More precisely, the verifier $V(x)$ samples the first message of the puzzle and sends it to the prover; next the verifier accepts the prover's response if it is either a witness for $x \in L$ (for some witness relation for L), or if the response is a valid response to the puzzle. The honest prover P simply sends a valid witness for x. Completeness of (P, V) trivially holds. Soundness holds due to the soundness of the puzzle (w.r.t. nu PPT). By the weak completeness property of the puzzle, we additionally have that (P, V) is not an interactive proof (since there are infinitely many input lengths on which an unbounded prover can find a puzzle solution and thus break soundness). ■

We finally observe that the existence of n^ϵ-round non-trivial public-coin arguments is equivalent to $\mathsf{PSPACE} \not\subseteq \mathsf{P/poly}$. We remark that one direction (that non-trivial arguments imply $\mathsf{PSPACE} \not\subseteq \mathsf{P/poly}$) was already previously proven by Goldreich [Gol18].

Theorem 7.3 (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* $\mathsf{PSPACE} \not\subseteq \mathsf{P/poly}$*.*

Proof: The "only-if" direction (which was already proven by Goldreich [Gol18]) follows just as the only-if direction of Theorem 5.1. The "if" direction follows by combining a standard NP proof with the puzzle from Theorem 5.1 and requiring the prover to either provide the NP witness, or to provide a solution to puzzle. ■

Round Collapse for Succinct Arguments We proceed to remark that the proof of our round-collapse theorem also has consequences for succinct [Kil92] and universal [Mic00, BG02] argument systems.

Theorem 7.4. *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 $O(\ell(n)\mathsf{polylog}(n))^{k(n)-1}$.*

Proof: We apply the BM round-collapse transformation to the k-round argument system $k-1$ times, and between each application repeat the protocol in parallel $\log^2 n$ times (where n is the length of the statement to be proven). Completeness (also w.r.t. efficient provers) follows directly from the classic proof of the BM round collapse [BM88]. To show soundness, as before, we consider a single application of the round-collapse transformation. Consider an adversary that breaks the soundness of the $k-1$-round argument system with probability $\epsilon(n)$ for infinitely many $\{x_n\}_n$; by [PV12, HPWP10] such an adversary can be turned into an adversary that break the soundness of a single of the $\log^2 n$ repetitions of the protocol obtained after the BM transformation with probability $\epsilon'(n) > \frac{1}{2}$ for infinitely many $\{x_n\}_n$. If non-uniform ioOWFs does not exist, then we can rely on the same construction as in the proof of Lemma 4.1 to construct an adversary B^* that breaks the k-round argument system for the same statements $\{x_n\}_n$ with probability $\frac{1}{64}$, which contradicts the soundness of the k-round argument.

Note that each step of the round-collapse transformation has a multiplicative overhead of $O(\ell_a(n)\mathsf{polylog}(\mathsf{n}))$ where $\ell_a(n)$ bounds the length of the prover messages. Therefore, iterating the round collapse transformation i times will result in a multiplicative overhead of $O((\ell(n)\mathsf{polylog}(\mathsf{n}))^i)$. ■

Theorem 7.4 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* [CGGM00] privacy for the prover (e.g., resettable witness indistinguishability (WI) or witness hiding (WH) [CGGM00, FS90]), then so will the resulting 2-round protocol. (The reason we do not consider resettable zero-knowledge is that due to [OW93b] 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.)

8 Acknowledgements

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.

References

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[BCGL92] Shai Ben-David, Benny Chor, Oded Goldreich, and Michael Luby. On the theory of average case complexity. *J. Comput. Syst. Sci.*, 44(2):193–219, 1992.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BG02] Boaz Barak and Oded Goldreich. Universal arguments and their applications. In *IEEE Conference on Computational Complexity*, pages 194–203, 2002.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke 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*, pages 1–18, 2001.

[BM88] László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

[BOV07] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *J. ACM*, 56(3):14:1–14:57, 2009.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC '00*, pages 235–244, 2000.

[CL10] Kai-Min Chung and Feng-Hao 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*, pages 19–36, 2010.

[CP15] Kai-Min Chung and Rafael 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*, pages 229–246, 2015.

[DGP09] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *Commun. ACM*, 52(2):89–97, 2009.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 283–293, 2000.

[DP11] Constantinos Daskalakis and Christos 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*, pages 790–804, 2011.

[ESY84] Shimon Even, Alan L. Selman, and Yacov Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, 1984.

[EY80] Shimon Even and Yacov Yacobi. Cryptocomplexity and np-completeness. In *Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherlands, July 14-18, 1980, Proceedings*, pages 195–207, 1980.

[FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993.

[FGM+89] Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. *Advances in Computing Research*, 5:429–442, 1989.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC '90*, pages 416–426, 1990.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. On the cryptographic applications of random functions. In *CRYPTO*, pages 276–288, 1984.

[GH98] Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Inf. Process. Lett.*, 67(4):205–214, 1998.

[Gin66] Seymour Ginsburg. *The Mathematical Theory of Context-Free Languages.* McGraw-Hill, Inc., USA, 1966.

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh 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*, pages 325–335, 2000.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[Gol01] Oded Goldreich. *Foundations of Cryptography — Basic Tools*. Cambridge University Press, 2001.

[Gol06] Oded Goldreich. On promise problems: A survey. In *Theoretical Computer Science, Essays in Memory of Shimon Even*, pages 254–290, 2006.

[Gol18] Oded Goldreich. On doubly-efficient interactive proof systems. *Foundations and Trends in Theoretical Computer Science*, 13(3):158–246, 2018.

[GP16] Paul W. Goldberg and Christos H. Papadimitriou. Towards a unified complexity theory of total functions. Unpublished manuscript, 2016.

[Gur89] Yuri Gurevich. The challenger-solver game: variations on the theme of p=np. In *Logic in Computer Science Column, The Bulletin of EATCS*. 1989.

[Gur91] Yuri Gurevich. Average case completeness. *J. Comput. Syst. Sci.*, 42(3):346–398, 1991.

[GW11] Craig Gentry and Daniel 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*, pages 99–108, 2011.

[Hai09] Iftach Haitner. A parallel repetition theorem for any interactive argument. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:27, 2009.

[HHR+10] Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. In *EUROCRYPT*, pages 616–637, 2010.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HNY17] Pavel Hub'avcek, Moni Naor, and Eylon 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*, pages 60:1–60:21, 2017.

[HPWP10] Johan Håstad, Rafael Pass, Douglas Wikström, and Krzysztof Pietrzak. An efficient parallel repetition theorem. In *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, pages 1–18, 2010.

[IL89] Russell Impagliazzo and Michael 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*, pages 230–235, 1989.

[IL90] Russell Impagliazzo and Leonid 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*, pages 812–821, 1990.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory '95*, pages 134–147, 1995.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if e requires exponential circuits: Derandomizing the xor lemma. In *STOC '97*, pages 220–229, 1997.

[JPY85] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? (extended abstract). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 39–42, 1985.

[Kil92] Joe 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*, pages 723–732, 1992.

[KK05] Jonathan Katz and Chiu-Yuen Koo. On constructing universal one-way hash functions from arbitrary one-way functions. Cryptology ePrint Archive, Report 2005/328, 2005.

[KMN+14] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. *IACR Cryptology ePrint Archive*, 2014:347, 2014.

[Lev86] Leonid A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.

[MP91] Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2):317–324, 1991.

[MV05] Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005.

[Nao03] Moni 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*, pages 96–109, 2003.

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

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC '89*, pages 33–43, 1989.

[OW93a] Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *ISTCS*, pages 3–17, 1993.

[OW93b] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Theory and Computing Systems, 1993*, pages 3–17, 1993.

[Pap94] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994.

[Pas11] Rafael 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*, pages 109–118, 2011.

[PV12] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. A parallel repetition theorem for constant-round arthur-merlin proofs. *TOCT*, 4(4):10:1–10:22, 2012.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, 1990.

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[Tre05] Luca 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*, pages 31–38, 2005.

[TV07] Luca Trevisan and Salil P. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007.

[Ull67] Joseph S. Ullian. Partial algorithm problems for context free languages. *Information and Control*, 11(1/2):80–101, 1967.

[Wee05] Hoeteck Wee. On round-efficient argument systems. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, pages 140–152, 2005.

[Wee06] Hoeteck Wee. Finding pessiland. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 429–442, 2006.

A Some Theorems from Average-Case Complexity

In this section, we provide formal justifications for Lemmas 2.1, 2.2 and 2.3 We recall some previous results on average-case complexity relevant to our work.

Theorem A.1 ([Tre05])**.** *Suppose that there exists a* NP *language* L *and polynomials* $\ell(\cdot)$ *and* $p(\cdot)$ *such that* $(L,\mathcal{U}_\ell)$ *is* $\frac{1}{p(n)}$*-HOA (resp.,* $\frac{1}{p(n)}$*-aeHOA and* $\frac{1}{p(n)}$*-nuHOA). Then there exists a* NP *language* L' *and polynomial* $\ell'(\cdot)$ *such that* $(L',\mathcal{U}_{\ell'})$ *is* $\frac{1}{2}-\frac{1}{(\log n)^\alpha}$*-HOA (resp.,* $\frac{1}{2}-\frac{1}{(\log n)^\alpha}$*-aeHOA and* $\frac{1}{2}-\frac{1}{(\log n)^\alpha}$*-nuHOA). The value* $\alpha > 0$ *is an absolute constant.*

Theorem A.2 ([IL90])**.** *Suppose there exists a distributional* NP *search problem* $(\mathcal{R},\mathcal{D})$ *that is* $\frac{1}{p(n)}$*-SearchHOA (resp.,* $\frac{1}{p(n)}$*-aeSearchHOA and* $\frac{1}{p(n)}$*-nuSearchHOA) for some polynomial* $p(\cdot)$*. Then there exists a search problem* $\mathcal{R}'$ *and polynomials* $\ell(\cdot)$ *and* $q(\cdot)$ *such that* $(\mathcal{R}',\mathcal{U}_\ell)$ *is* $\frac{1}{q(n)}$*-SearchHOA (resp.,* $\frac{1}{q(n)}$*-aeSearchHOA and* $\frac{1}{q(n)}$*-nuSearchHOA).*

Theorem A.3 ([BCGL92])**.** *Suppose that there exists a distributional* NP *search problem* $(\mathcal{R},\mathcal{U}_\ell)$ *that is* $\frac{1}{p(n)}$*-SearchHOA (resp.,* $\frac{1}{p(n)}$*-aeSearchHOA and* $\frac{1}{p(n)}$*-nuSearchHOA) for some polynomials* $p(\cdot)$ *and* $\ell(\cdot)$*. Then there is a* NP*-language* L' *and polynomials* $\ell'(\cdot)$ *and* $q(\cdot)$ *such that* $(L',\mathcal{U}_{\ell'})$ *is* $\frac{1}{q(n)}$*-HOA (resp.,* $\frac{1}{q(n)}$*-aeHOA and* $\frac{1}{q(n)}$*-nuHOA). If we start with a distributional* $\mathsf{NP/poly}$ *search problem* $(\mathcal{R},\mathcal{U}_\ell)$ *that is* $\frac{1}{p(n)}$*-nuSearchHOA, then we obtain* $L' \in \mathsf{NP/poly}$ *such that* $(L',\mathcal{U}_{\ell'})$ *is* $\frac{1}{q(n)}$*-nuHOA.*

Theorems A.1, A.2 and A.3 are stated in a slightly different form in [Tre05, IL90, BCGL92]. Namely, we highlight that the reduction is in fact "length-regular"[23] in that solving instances of size $\ell(n)$ in the target language helps solving instances of size n in the source language. We will require this stronger property for the reductions to hold in the case of almost-everywhere hardness.

Proof of Lemma 2.2. This follows immediately from Theorem A.1.

Proof of Lemma 2.3. Suppose there exists a distributional NP-search problem $(\mathcal{R}, \mathcal{D})$ that is SearchHOA (resp., aeSearchHOA and nuSearchHOA). By Theorem A.2, there exists a search problem $\mathcal{R}'$ and polynomials $\ell(\cdot), q(\cdot)$ such that $(\mathcal{R}', \mathcal{U}_\ell)$ is $\frac{1}{q(n)}$-SearchHOA (resp., aeSearchHOA and nuSearchHOA). Next, by Theorem A.3, there is a NP-language L' and polynomials $\ell'(\cdot)$ and $q'(\cdot)$ such that $(L, \mathcal{U}_\ell)$ is $\frac{1}{q'(n)}$-HOA (resp. aeHOA and nuHAO), which when combined with Theorem A.1 yields a NP language L'' and a polynomial ℓ'' such that $(L', \mathcal{U}_{\ell''})$ is $\frac{1}{2} - \frac{1}{(\log n)^\alpha}$-HOA (resp., aeHOA and nuHAO). This implies that $(L'', \mathcal{U}_{\ell''})$ is HOA (resp., aeHOA and nuHOA).

Proof of Lemma 2.1. Suppose $(L, \mathcal{D})$ is a distributional NP problem that is HOA (resp. a distributional $\mathsf{NP/poly}$ problem that is nuHOA), then $(\mathcal{R}, \mathcal{D})$ is SearchHOA (resp., nuSearchHOA) where $\mathcal{R}$ is the witness relation corresponding to L. By Lemma 2.3, we can obtain a NP (resp. $\mathsf{NP/poly}$) language L' and polynomial ℓ' such that $(L', \mathcal{U}_{\ell'})$ is HOA (resp., nuHOA).

[23]A length-regular function $f : \{0,1\}^* \to \{0,1\}^*$ satisfies the properties that: (1) $|x| = |y| \Leftrightarrow |f(x)| = |f(y)|$,and (2) $|x| < |y| \Leftrightarrow |f(x)| < |f(y)|$ for any two strings x,y. We require the "length-regular" property for Turing (Cook) reductions where solving an instance x on the target language requires queries the oracle only on instances of size $\ell(|x|)$ on the source language where ℓ is a non-decreasing function.