On Worst-Case to Average-Case Reductions for NP Problems*

Andrej Bogdanov† Luca Trevisan‡

January 24, 2006

Abstract

We show that if an NP-complete problem has a non-adaptive self-corrector with respect to any samplable distribution then coNP is contained in NP/poly and the polynomial hierarchy collapses to the third level. Feigenbaum and Fortnow (SICOMP 22:994-1005, 1993) show the same conclusion under the stronger assumption that an NP-complete problem has a non-adaptive random self-reduction.

A self-corrector for a language L with respect to a distribution $\mathcal{D}$ is a *worst-case to average-case reduction* that transforms any given algorithm that correctly decides L on *most* inputs (with respect to $\mathcal{D}$) into an algorithm of comparable efficiency that decides L correctly on *every* input. A random self-reduction is a special case of a self-corrector where the reduction, given an input x, is restricted to only make oracle queries that are distributed according to $\mathcal{D}$. The result of Feigenbaum and Fortnow depends essentially on the property that the distribution of each query in a random self-reduction is independent of the input of the reduction.

Our result implies that the average-case hardness of a problem in NP or the security of a one-way function cannot be based on the worst-case complexity of an NP-complete problem via non-adaptive reductions (unless the polynomial hierarchy collapses).

1 Introduction

The fundamental question in the study of average-case complexity is whether there exist distributional problems in NP that are intractable on average. A distributional problem in NP is a pair $(L, \mathcal{D})$, where L is a decision problem in NP, and $\mathcal{D}$ is a samplable distribution on instances. We will say that such a problem is tractable on average if for every polynomial p there exists a polynomial-time algorithm A such that for all sufficiently large n, when given a random instance of length n from distribution $\mathcal{D}$, algorithm A determines membership in L correctly with probability at least $1 - 1/p(n)$.

*A preliminary version of this paper appeared in the Proceedings of the 2003 IEEE Conference on Foundations of Computer Science (FOCS'03).

†School of Mathematics, Institute for Advanced Study. adib@ias.edu. This work was done while the author was at U.C. Berkeley.

‡Computer Science Division, U.C. Berkeley. luca@cs.berkeley.edu. Research supported by a Sloan Research Fellowship, an Okawa Foundation Grant and NSF grants CCR-9984703 and CCR-0406156.

This notion of average-case tractability is essentially equivalent to Impagliazzo's definition of *heuristic* polynomial-time algorithms [Imp95]. Impagliazzo observes that if every problem in distributional NP is tractable on average, then there are no one-way functions, thus cryptography is impossible. It is therefore generally believed that distributional NP does contain problems that are intractable on average.

The question we consider in this paper concerns the minimal complexity assumption one needs to make in order to guarantee that distributional NP does indeed contain a problem that is intractable on average. Ideally, one would like to base the existence of hard on average problems (and one-way functions) on a worst-case assumption, namely NP $\not\subseteq$ BPP. Equivalently, the question can be formulated as follows: Is the existence of worst-case hard problems in NP sufficient to show the existence of problems in NP that are hard on average?

In the cryptographic setting, the question of whether there are cryptosystems that are NP-hard to break, that is, whose security can be based on the assumption that NP $\not\subseteq$ BPP, is as old as modern cryptography itself, and it was asked by Diffie and Hellman [DH76, Section 6]. As we review below, there is conflicting evidence about the answer to this question.

Previous work

Worst-case versus average-case in NP**.** Impagliazzo [Imp95] observes that we know oracles relative to which NP $\not\subseteq$ P/poly, but there is no intractable problem in distributional NP (and consequently, one-way functions do not exist). Therefore, any proof that NP $\not\subseteq$ BPP implies the existence of an intractable problem in distributional NP must use a non-relativizing argument. However, non-relativizing arguments are commonly used in lattice based cryptography to establish connections between the worst-case and average-case hardness of certain NP problems that are believed to be intractable (but not NP-complete).

Feigenbaum and Fortnow [FF93] consider the notion of a *random self-reduction*, which is a natural, and possibly non-relativizing, way to prove that the average-case complexity of a given problem relates to its worst-case complexity. We begin by discussing the slightly more general notion of *locally random reduction*, introduced in [BFKR97] (see also the earlier works [AFK89, BF90, FKN90]). A locally random reduction from a language L to a distributional problem $(L', \mathcal{D})$ is a polynomial-time oracle procedure R such that $R^{L'}$ solves L and, furthermore, each oracle query of $R^{L'}(x)$ is distributed according to $\mathcal{D}$.[1] Clearly, such a reduction converts a heuristic polynomial time algorithm for $(L', \mathcal{D})$ (with sufficiently small error probability) into a BPP algorithm for L. Observe that the reduction may depend on the choice of distributional problem $(L', \mathcal{D})$, so in general this approach does not relativize. If we had a locally random reduction from, say, 3SAT to some problem $(L', \mathcal{D})$ in distributional NP, then we would have proved that if NP $\not\subseteq$ BPP then distributional NP contains intractable problems. A locally random reduction from L to $(L', \mathcal{D})$ is called a *random self-reduction* if $L = L'$.

Feigenbaum and Fortnow show that if there is a *non-adaptive* random self-reduction from L to $(L, \mathcal{D})$ for an NP-complete language L and a samplable ensemble $\mathcal{D}$, then NP $\subseteq$ coNP/poly and the polynomial hierarchy collapses to the third level. Their proof also establishes the slightly more

[1] More precisely, according to the restriction of $\mathcal{D}$ to inputs of length polynomially related to x (see Section 2).

general result that if there is a non-adaptive locally random reduction from a problem L to a problem $(L', \mathcal{D})$ in distributional NP, then L is in coNP/poly.

Random self-reductions and locally random reductions are natural notions, and they have been used to establish the worst-case to average-case equivalence of certain PSPACE-complete and EXP-complete problems [STV01]. Therefore, the result of Feigenbaum and Fortnow rules out a natural and general approach to prove a statement of the form: "If NP $\not\subseteq$ BPP then distributional NP contains intractable problems."

Cryptography versus NP-hardness. The seminal work of Diffie and Hellman [DH76] introducing public key cryptography asked if there exists a public key encryption scheme whose hardness can be based on an NP-complete problem. This question was seemingly answered in the affirmative by Even and Yacobi [EY80], who devised a public key cryptosystem that is "NP-hard to break". Namely, they showed a reduction that transforms any adversary that "breaks" the cryptosystem into an algorithm that solves SAT. However, the notion of "breaking the cryptosystem" in [EY80] is a worst-case one: Specifically, it is assumed that the adversary can break the encryption for *every* key. Lempel [Lem79] later showed that the same cryptosystem can in fact be broken on *most* keys. Therefore, the NP hardness of breaking a cryptosystem in the worst case does not in general have any implications for cryptographic security.

The gap between worst-case and average-case hardness is even more transparent in the case of symmetric key cryptography, or one-way functions. It is well known that there exist "one-way functions" that are NP-hard to invert in the worst case, but easy to invert on average. (For instance, consider the function that maps a formula ϕ and an assignment a to $(1, \phi)$ if a satisfies ϕ and to $(0, \phi)$ otherwise.)

As these examples show, the notion of hardness in breaking a public key cryptosystem, or inverting a one-way function that one needs in cryptography is fundamentally an average-case notion.

Brassard [Bra79] addresses the question of the existence of public key cryptosystems that are hard to break from a different perspective. He argues that, under some assumptions on the key-generation algorithm and the encryption procedure, the problem of breaking the encryption is in NP $\cap$ coNP. Specifically, he shows that if there is a reduction that transforms an oracle that breaks encryptions to an algorithm for a language L, then the reduction can be used to provide NP certificates for membership in both L and $\overline{L}$, proving that $L \in$ NP $\cap$ coNP. More recent work by Goldreich and Goldwasser [GG98] reaches the same conclusion under weaker assumptions.

In the setting of symmetric key cryptography, a similar conclusion can be reached about the hardness of inverting a one-way function if one makes additional assumptions about the function in question. For instance, if the function f is a permutation, then the existence of a reduction from any language L to an inverter for f establishes that $L \in$ NP $\cap$ coNP. A proof for membership in L or $\overline{L}$ consists of the transcript of all the queries made by the reduction, together with unique preimages of the queries under f. The fact that f is a permutation guarantees that this transcript perfectly simulates the reduction when given access to an inverting oracle for f.

These arguments explain why the hardness of breaking a large class of cryptosystems cannot be based on the worst-case complexity of an NP complete problem (assuming NP $\neq$ coNP). However,

neither of them uses the fact that the reduction that transforms the adversary into an algorithm for L is correct even if the adversary only performs its task well on average. In fact, the arguments merely assume that the reduction behaves correctly when given oracle access to an adversary that violates a *worst-case* assumption. Given the existence of public-key cryptosystems and one-way functions that are hard to break in the worst case, one cannot expect these arguments to explain why breaking a general one-way function or a general public key encryption scheme should be an NP $\cap$ coNP problem, as experience seems to indicate, if this is indeed the case.

If we were to ever hope for such an explanation, we need a stronger notion of "NP hard to break", which allows for the fact that the adversary may err on some fraction of inputs. Again, what we mean by a cryptosystem being "NP-hard to break" is that there exists a reduction that transforms an adversary for the cryptosystem into an algorithm for SAT, but now the reduction is required to solve SAT correctly even if the adversary sometimes outputs an incorrect answer.

This motivates the following definition of a reduction from an NP-complete problem to the problem of inverting well on average a one-way function f: A reduction is an oracle probabilistic polynomial time procedure R such that for some polynomial p and for every oracle A that inverts f on a $1 - 1/p(n)$ inputs of length n, it holds that R^A is a BPP algorithm for SAT. The techniques of Feigenbaum and Fortnow imply that if R is non-adaptive, and if all of its oracle queries are done according to the same distribution (that depends only on the length of the input), then the existence of such a reduction implies that the polynomial hierarchy collapses to the third level.

As we explain below, we reach the same conclusion (see Theorem 17 in Section 4) without any assumption on the distribution of the queries made by R^A, but also assuming as in [FF93] that the queries are made non-adaptively.

Worst-case to average-case reductions within NP**.** The most compelling evidence that the average-case hardness of certain problems in NP *can* be based on worst-case intractability assumptions comes from lattice based cryptography.

Ajtai [Ajt96] shows that an algorithm that solves well on average the shortest vector problem (which is in NP) under a certain samplable distribution of instances implies an algorithm that solves, in the worst case, an approximate version of the shortest vector problem. The latter can be seen as an NP promise problem. If the latter problem were NP-complete, then we would have a reduction relating the average-case hardness of a distributional problem in NP to the worst-case hardness of an NP-complete problem. Unfortunately, the latter problem is known to be in NP $\cap$ coNP, and therefore it is unlikely to be NP-hard. However, it is conceivable that improved versions of Ajtai's argument could show the equivalence between the average-case complexity of a distributional NP problem and the worst-case complexity of an NP problem. Micciancio [Mic04] and Micciancio and Regev [MR04] improve Ajtai's reduction by showing that a good on average algorithm for the generalized subset sum problem implies better worst-case approximation algorithms for a variety of problems on lattices. Such approximations, however, still correspond to promise problems known to be in NP $\cap$ coNP.

Average-case complexity in NP**.** The theory of average-case complexity was pioneered by Levin [Lev86], who defined the notion of "efficient on average" algorithms and gave a distributional

problem that is complete for a large subclass of distributional NP. Levin's notion of efficient on average algorithms is stronger than Impagliazzo's notion of polynomial-time heuristic algorithms that we consider here. Namely, every problem in distributional NP that admits an efficient on average algorithm also admits an efficient heuristic algorithm.

The subclass of distributional NP problems considered by Levin imposes a severe restriction on the distribution according to which instances of the problem are sampled. In particular, it does not include the problem of inverting arbitrary one-way functions. In the case of a one-way function f, the notion of "inverting f well on average" amounts to solving the search problem "Given u, find x s.t. $f(x) = u$", where u is chosen according to the distribution obtained by applying f to the uniform distribution. In general, f may be an arbitrary polynomial-time algorithm, so it makes sense to relax the definition so as to allow instances of L to be generated by arbitrary polynomial-time samplers. This yields the class distributional NP (introduced by Ben-David and others [BCGL89]) of all pairs $(L, \mathcal{D})$ where L is an NP language and $\mathcal{D}$ is an arbitrary samplable distribution according to which inputs for L are generated.

The class distributional NP turns out to be surprisingly robust for (randomized) heuristic algorithms. In particular, there exists an NP language L such that if L is tractable on average with respect to the uniform distribution, then every problem in NP is tractable on average with respect to any samplable distribution. Moreover, the average-case algorithms for distributional NP are "search algorithms" in the sense that they provide witnesses of membership for most of the "yes" instances. In particular, average-case tractability of L implies the ability to efficiently invert one-way functions on most inputs $f(x)$, where x is chosen uniformly at random. These results on distributional NP were established by Ben-David and others [BCGL89] and Impagliazzo and Levin [IL90].

For an overview of these and other notions in average-case complexity, their interrelations, and explanations of the various choices made in definitions, the reader is referred to the expository papers by Impagliazzo [Imp95], Goldreich [Gol97], and the authors [BT05].

Our result

A worst-case to average-case reduction with parameter δ from a language L to a distributional problem $(L', \mathcal{D})$ is a probabilistic polynomial-time oracle procedure R such that, for every oracle A that agrees with L' on inputs of probability mass $1 - \delta$ according to $\mathcal{D}$ on each input length, R^A solves L on every input.

If L and L' are the same language, then the reduction is called a self-corrector, a notion independently introduced by Blum and others [BLR93] and by Lipton [Lip89] in the context of program checking [Blu88, BK95]. As argued below, a locally random reduction is also a worst-case to average-case reduction and a random self-reduction is also a self-corrector, but the reverse need not be true.

In this paper we show that if there is a worst-case to average-case reduction with parameter $1/\mathrm{poly}(n)$ from an NP-complete problem L to a distributional NP problem $(L, \mathcal{D})$, then NP $\subseteq$ coNP/poly and the polynomial hierarchy collapses to the third level. In particular, if an NP-complete problem has a self-corrector with respect to a samplable distribution, then the polynomial

hierarchy collapses to the third level.

We first prove the result for the special case in which the distribution $\mathcal{D}$ is uniform (Theorem 17). Then, using reductions by Impagliazzo and Levin [IL90] and by Ben-David and others [BCGL89], we show that the same is true even if the reduction assumes a good-on-average algorithm for the *search* version of L', and even if we measure average-case complexity for L' with respect to an arbitrary samplable distribution $\mathcal{D}$ (Theorem 20).

The generalization to arbitrary samplable distributions and to search problems also implies that there cannot be any non-adaptive reduction from an NP-complete problem to the problem of inverting a one way function.

Our result also rules out non-adaptive reductions from an NP-complete problem to the problem of breaking a public-key cryptosystem. The constraint of non-adaptivity of the reduction is incomparable to the constraints in the results of Goldreich and Goldwasser [GG98].

It should be noted that some of the worst-case to average-case reductions of Ajtai, Dwork, Micciancio, and Regev [Ajt96, AD97, Mic04, Reg03, MR04] are *adaptive*. Micciancio and Regev [MR04] observe that their reductions can be made non-adaptive with a slight loss in worst-case approximation factors.

Comparison with Feigenbaum-Fortnow [FF93]. It is easy to see that a locally random reduction R from L to L' that makes q queries, each of which is generated by the reduction according to a distribution $\mathcal{D}$, is also a worst-case to average-case reduction with parameter $\Omega(1/q)$ from L to $(L', \mathcal{D})$. Indeed, if A is an oracle that has agreement, say, $1 - 1/4q$ with L' (as measured by $\mathcal{D}$), and we access the oracle via q queries, each distributed according to $\mathcal{D}$, there is a probability at least $3/4$ that queries made to A are answered in the same way as queries made to L'.

For the result of Feigenbaum and Fortnow, it is not necessary that the distribution of each query made by the reduction be exactly $\mathcal{D}$, but it is essential that the marginal distribution of queries made by the reduction be independent of the reduction's input. This restriction is quite strong, and in this sense, the result of [FF93] is extremely sensitive: If one modifies the distribution of queries even by an exponentially small amount that depends on the input, all statistical properties of the reduction are preserved, but one can no longer draw the conclusion of [FF93]. Our result reaches the same conclusion as [FF93], yet allows the queries made by the reduction to depend arbitrarily on the input.

One natural setting where the queries made by the reduction seem to essentially depend on the input is Levin's theory of average-case complexity. One tool for relating the average-case hardness of two distributional problems is the "average-case to average-case reduction". Such a reduction from $(L, \mathcal{D})$ and $(L', \mathcal{D}')$, uses an oracle that solves L' on most inputs chosen from $\mathcal{D}'$ to solve L on most inputs according to $\mathcal{D}$. Some important reductions, most notably those in [BCGL89, IL90], choose their queries to the oracle from a distribution that depends on the input in an essential way, making the results of [FF93] useless for their study.

The relation between locally random reductions and our notion of worst-case to average-case reduction is similar to the relation between one-round private information retrieval and locally decodable codes [CGKS98, Tre05]. In one-round private information retrieval, a "decoder" is given oracle ac-

cess to the encoding of a certain string, and wants to retrieve one bit of the string by making a bounded number of queries; the restriction is that the i-th query must have a distribution independent of the bit that the decoder is interested in. In a locally decodable code, a decoder is given oracle access to the encoding of a certain string, and the encoding has been corrupted in a δ fraction of places; the decoder wants to retrieve a bit of the original string by making a bounded number of queries (with no restriction on the distribution on queries). An intermediate notion that is useful in the study of the relation between private information retrieval and locally decodable codes is that of a smooth decoder: Such a decoder satisfies the additional requirement that the distribution of each query should be dominated by the uniform distribution. Similarly, in the setting of worst-case to average-case reductions one can restrict attention to smooth reductions, where the distribution of queries made by the reduction is dominated by the uniform distribution.

For computationally unbounded decoders, it has been shown (see [KT00, GKST02]) that uniform, smooth, and general decoders are equivalent, but the same methods do not work in the computationally bounded setting studied in this paper. One step in our proof is, however, inspired by the techniques used to show this equivalence.

Our proof

As in the work of Feigenbaum and Fortnow, we use the fact that problems in coNP/poly cannot be NP-complete unless the polynomial hierarchy collapses to the third level. Our goal is to show that if L has a $1/\mathrm{poly}(n)$ worst-case to average-case reduction to a language $(L', \mathcal{D})$ in distributional NP, then L is in $\mathrm{NP/poly} \cap \mathrm{coNP/poly}$. In particular, if L were NP-complete, then NP would be contained inside coNP/poly, which in particular implies the collapse of the polynomial hierarchy to the third level. (However, the conclusion $\mathrm{NP} \subseteq \mathrm{coNP/poly}$ appears weaker than the more standard statement $\mathrm{NP} = \mathrm{coNP}$.)

Feigenbaum and Fortnow observe that NP/poly is exactly the class of languages that admit AM protocols with polynomial length advice. Then they show $L \in \mathrm{NP/poly} \cap \mathrm{coNP/poly}$ by giving AM protocols with advice for both L and its complement. The protocols for L and its complement are completely analogous, so we focus on describing the protocol for L.

We begin by discussing the case of a reduction from L to $(L', \mathcal{D})$ when $\mathcal{D}$ is the uniform distribution.

The Feigenbaum-Fortnow protocol. Let us first briefly review the proof of Feigenbaum and Fortnow [FF93]. Given x, a prover wants to prove that $R^{L'}(x)$ accepts with high probability (implying $x \in L$), where R makes q non-adaptive queries, each uniformly distributed. The (non-uniform) verifier generates k independent computations of $R^{L'}(x)$ and sends to the prover all the kq queries generated in all the k runs. The prover has to provide all the answers, and certificates for all the "yes" answers. The verifier, non-uniformly, knows the overall fraction p of queries of $R^{L'}(x)$ whose answer is "yes" (recall that we assumed that the queries of R, and thus p, is independent of x). If k is large enough, the verifier expects the number of "yes" answers from the prover to be concentrated around kqp, and in fact is within $kqp \pm O(q\sqrt{k})$ with high probability. If the prover gives fewer than $kqp - O(q\sqrt{k})$ "yes" answers, this provides strong evidence of cheating, and the verifier rejects. Since a cheating prover must provide certificates for all its "yes" claims, such a

prover can only cheat by saying "no" on a "yes" query, and cannot do so on more than $O(q\sqrt{k})$ instances. If k is sufficiently larger than q, then with high probability either the verifier rejects, or a majority of the k computations of $R^{L'}(x)$ yields the correct answer, making the reduction output "yes" and the verifier accept.

Handling Smooth Reductions: The hiding protocol. The Feigenbaum-Fortnow protocol can be used with *every* oracle procedure R, provided that given x we can get a good estimate of the average fraction p_x of oracle queries made by R on input x that are answered "yes" by an oracle for L'. In general, this fraction will depend on x, so it cannot be provided as an advice to the AM circuit certifying membership in L.

For starters, let us allow the distribution of R's queries to depend on x, but restrict it to be "α-smooth": We assume that every query y of R is generated with probability at most $\alpha 2^{-|y|}$. (It is useful to think of α as constant, or at most polynomial in $|y|$, so that a query made by R is not much more likely to hit any specific string than in the uniform distribution.) We devise an AM protocol with advice in which the verifier either rejects or gets a good estimate of p_x. This estimate is then fed into the Feigenbaum-Fortnow protocol to obtain an AM circuit for L.

Suppose that, given a *random* query y made by $R(x)$, we could force the prover to reveal whether or not $y \in L'$. Then by sampling enough such queries y, we can estimate p_x as the fraction of "yes" queries made by the reduction. But how do we force the prover to reveal if $y \in L'$? The idea is to hide the query y among a sequence of queries $z_1, \ldots, z_k$ for which we *do* know whether $z_i \in L'$, in such a way that the prover cannot tell where in the sequence we hid our query y. In such a case, the prover is forced to give a correct answer for y, for if he were to cheat he wouldn't know where in the sequence to cheat, thus would likely be caught.

The problem is that we do not know a specific set of queries z_i with the desired property. We do, however, know that if we chose z_i independently from the uniform distribution on $\{0,1\}^{|y|}$, then with high probability $pk \pm O(\sqrt{k})$ of these queries will end up in L', where p is the probability that a *uniformly random* query in $\{0,1\}^{|y|}$ is in L'. Since p depends only on the length of x but not on x itself, it can be given to the verifier non-uniformly.

This suggests the following verifier strategy: Set $k = \omega(\alpha^2)$, generate k uniformly random queries $z_1, \ldots, z_k$ of length n, hide y among $z_1, \ldots, z_k$ by inserting it at a random position in the sequence, send all the queries to the prover and ask for membership in L' together with witnesses that at least $pk - O(\sqrt{k})$ queries belong to L'.

We claim that, with high probability, either the verifier rejects or the answer about membership of y in L' must be correct. Intuitively, a cheating prover can give at most $O(\sqrt{k})$ wrong answers. The prover wants to use this power wisely and assign one of these wrong answers to the query y. However, smoothness ensures that no matter how the prover chooses the set of $O(\sqrt{k})$ queries to cheat on, it is very unlikely that the query y falls into that set.

For ease of analysis, the actual proof presented in Section 3.2 combines the step of sampling enough ys to estimate p_x together with the step of hiding y among a sequence of uniform queries into a single step.

This argument already provides an interesting generalization to [FF93]. Notice that we have not yet

used the fact that the reduction is allowed access to any oracle that computes L' well on average.

Handling General Reductions. We now consider the case of general reductions, allowing the distribution of a random query on input x to depend arbitrarily on x. Observe that the hiding protocol will, in general, fail to estimate p_x for this type of reduction. If a particular query y made by the reduction is very likely (that is, it occurs with probability much greater than $\alpha 2^{-|y|}$), then it cannot be hidden in a reasonably long sequence of uniform queries.

However, suppose that the verifier had the ability to identify queries y that occur with probability $\geq \alpha 2^{-|y|}$; let us call such queries "heavy", and the other ones "light". The fraction of heavy queries in the uniform distribution is at most $1/\alpha$. Suppose also that the prover answers all light queries correctly. We can then use R to certify membership in L as follows: If the query made by R is heavy, pretend that the oracle for R answered "no", otherwise use the answer provided by the prover. This process simulates exactly a run of the reduction R^A, where A is an oracle that agrees with L' on all the light queries, and answers "no" on all the heavy queries. In particular, A agrees with L' on a $1-1/\alpha$ fraction of inputs, so the reduction is guaranteed to return the correct answer.

In general, the verifier cannot identify which queries made by the reduction are heavy and which are light. However, suppose the verifier knew the probability q_x that a random query, on input x, is heavy. Then, among any set of k independent queries, the verifier expects to see, with high probability, $q_x k \pm O(\sqrt{k})$ heavy queries. Using a protocol of Goldwasser and Sipser [GS86], the verifier can now obtain approximate AM certificates of heaviness for at least $q_x k - O(\sqrt{k})$ queries from the prover. This leaves at most $O(\sqrt{k})$ queries about whose heaviness the verifier may be misinformed.

A verifier with access to q_x can run a variant of the hiding protocol to calculate the fraction p_x of "yes" instances of L' among the light queries (treating the $O(\sqrt{k})$ heavy queries that "slip in the sample" as a statistical error to this estimate), followed by a variant of the Feigenbaum-Fortnow protocol, simulating "no" answers on all the heavy queries.

Finally, we need a protocol that helps the verifier estimate the probability q_x of heavy queries. The verifier can obtain an approximate lower bound on q_x by sampling random queries and asking for proofs that each query is heavy. To obtain an approximate upper bound on q_x, the verifier uses an "upper bound" protocol for the size of certain NP sets, due to Fortnow [For87]. The explanation of the exact roles of these protocols in estimating q_x is deferred to Section 3.1.

We observe that the generalization of the Feigenbaum-Fortnow result about locally random reductions to smooth, and then arbitrary non-adaptive reductions parallels an analogous sequence of steps establishing the equivalence of uniform, smooth, and arbitrary decoders for locally decodable codes.

General Distributions, Search Problems, One-Way Functions. So far we have described our results for the case in which the distribution on inputs $\mathcal{D}$ is the uniform distribution. We now consider the case where $\mathcal{D}$ is an arbitrary samplable distribution. Impagliazzo and Levin [IL90] show that for every distributional NP problem $(L, \mathcal{D})$ and bound $\delta = n^{-O(1)}$ there is a non-adaptive probabilistic polynomial time oracle algorithm R, an NP language L', and a bound $\delta' = \delta^{O(1)}$ such

that for every oracle A that agrees with L' on a $1-\delta'$ fraction of inputs, R^A solves L on a subset of inputs of density $1-\delta$ under the distribution $\mathcal{D}$.

This means that if there were a non-adaptive worst-case to average-case reduction with parameter $1/\text{poly}(n)$ from a problem L to a distributional problem $(L', \mathcal{D})$, there would also be such a reduction from L to $(L'', \mathcal{U})$, where $\mathcal{U}$ is the uniform distribution and L'' is in NP. By the previously described results, this would imply the collapse of the polynomial hierarchy.

A reduction by Ben-David and others [BCGL89] implies that for every distributional NP problem $(L, \mathcal{U})$ there is a problem L' in NP such that an algorithm that solves the decision version of $(L', \mathcal{U})$ on a $1-\delta$ fraction of inputs can be modified (via a non-adaptive reduction) into an algorithm that solves the search version of $(L, \mathcal{U})$ on a $1-\delta \cdot \text{poly}(n)$ fraction of input. This implies that even if we modify the definition of worst-case to average-case reduction so that the oracle A is supposed to solve the *search* version of the problem, our results still apply. In particular, for every polynomial time computable function f, the problem of inverting f well on average is precisely the problem of solving well on average a distributional NP search problem. Therefore our results also rule out the possibility of basing one-way functions on NP-hardness using non-adaptive reductions.

Organization. Section 2 provides the relevant definitions of notions in average case complexity and interactive proof systems. In Section 3 we present the protocols for estimating the fraction of heavy queries of a reduction, the fraction of light "yes" queries of a reduction, and for simulating the reduction, respectively. Section 4 contains the proof of our main result (Theorem 17) concerning the average-case complexity of languages with respect to the uniform distribution. In Section 5 we prove our result for the average-case complexity of distributional search problems (Theorem 20).

2 Preliminaries

In this section we formalize the notions from average case complexity and interactive proof systems needed to state and prove our result on the impossibility of worst-case to average-case reductions in NP.

For a distribution $\mathcal{D}$, we use $x \sim \mathcal{D}$ to denote a sample x chosen according to $\mathcal{D}$. For a finite set S, we use $x \sim S$ to denote a sample x chosen uniformly at random from S. For a sample x, we use $\mathcal{D}(x)$ to denote the probability of x in the distribution $\mathcal{D}$. For a set S, we use $\mathcal{D}(S)$ to denote the probability that a random sample chosen according to $\mathcal{D}$ falls inside the set S.

2.1 Distributional problems and heuristic algorithms

Intuitively, we think of an algorithm A as a "good heuristic algorithm" for distributional problem $(L, \mathcal{D})$ if the set of "yes"-instances of A (which we also denote by A) and the set L are close according to $\mathcal{D}$. Formalizing this definition requires one to make choices regarding how $\mathcal{D}$ measures closeness and what the threshold for closeness is.

Roughly, Levin [Lev86] considers two sets A and L to be close according to $\mathcal{D}$ if on a *random*

input length n, the measure of the symmetric difference $A \triangle L$ according to the restriction of $\mathcal{D}$ on $\{0,1\}^n$ is small. We will make the stronger requirement that this quantity be small for *all* n. Notice that every heuristic algorithm satisfying the stronger requirement also satisfies the weaker requirement (and therefore reductions that work for algorithms satisfying the weaker requirement also work for algorithms satisfying the stronger requirement), so it makes our impossibility result more general. This requirement simplifies some of the definitions, as we can now restrict our attention to ensembles of distributions over various input lengths rather than a single distribution over $\{0,1\}^*$.

We now turn to the actual definitions.

Definition 1 (Samplable ensemble)**.** *An* efficiently samplable ensemble of distributions *is a collection* $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots\}$*, where* $\mathcal{D}_n$ *is a distribution on* $\{0,1\}^n$*, for which there exists a probabilistic polynomial-time sampling algorithm* S *that, on input* 1^n*, outputs a sample from* $\mathcal{D}_n$*.*[2]

The *uniform ensemble* is the ensemble $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \dots\}$, where $\mathcal{U}_n$ is the uniform distribution on $\{0,1\}^n$.

A *distributional problem* is a pair $(L, \mathcal{D})$ where L is a language and $\mathcal{D}$ is an ensemble of distributions. A distributional problem $(L, \mathcal{D})$ is in the class *distributional* NP, denoted DistNP, if L is in NP and $\mathcal{D}$ is efficiently samplable.

In this paper we study hypothetical reductions that might be used to establish average-case intractability of distributional NP problems based on a worst-case assumption, such as NP $\not\subseteq$ BPP. The notion of average-case intractability that we have in mind is the absence of good-on-average algorithms of the following type. (The definition is in the spirit of the treatment by Impagliazzo [Imp95].)

Definition 2 (Heuristic Polynomial Time)**.** *We say that a probabilistic polynomial time algorithm* A *is a* heuristic algorithm with success probability $s(n)$ *for a distributional problem* $(L, \mathcal{D})$ *if, for every* n*,* $\Pr_{x \sim \mathcal{D}_n}[A(x) = L(x)] \geq s(n)$*, where the probability is taken over the sampling of* x *from* $\mathcal{D}_n$ *and over the internal coin tosses of* A*. The class of distributional problems for which such algorithms exist is denoted by* $\mathrm{Heur}_{s(n)}\mathrm{BPP}$*.*

We consider a distributional problem $(L, \mathcal{D})$ to be "hard on average" if there is a polynomial p such that $(L, \mathcal{D}) \notin \mathrm{Heur}_{1-1/p(n)}\mathrm{BPP}$. This is a fairly robust notion with respect to the choice of p: Trevisan [Tre05] proves that there is a constant $c > 0$ such that for every polynomial p,

$$\text{DistNP} \not\subseteq \mathrm{Heur}_{1-1/p(n)}\mathrm{BPP} \text{ if and only if } \text{DistNP} \not\subseteq \mathrm{Heur}_{1/2+(\log n)^{-c}}\mathrm{BPP}$$

Stronger collapses are known for non-uniform heuristic classes [O'D02, HVV04].

Levin's alternative notion of an "efficient on average" algorithm [Lev86] imposes the additional requirement that the average-case algorithm A be errorless: For every x for which $A(x) \neq L(x)$,

[2] This definition restricts the strings in the support of $\mathcal{D}_n$ to have length exactly n. It is possible to use a more relaxed definition in which the length of strings in the support of $\mathcal{D}_n$ is variable, as long as S is non-shrinking: Namely, the length of every string in the support of $\mathcal{D}_n$ must be at least n^ϵ for some constant $\epsilon > 0$.

$A(x)$ must output "fail" (with high probability over its internal coin tosses).[3] Hence every efficient on average algorithm for $(L, \mathcal{D})$ is also a heuristic algorithm for $(L, \mathcal{D})$, so the class of problems (in distributional NP) that are "hard for efficient on average algorithms" is possibly larger than the class of problems that are "hard for heuristic algorithms". Therefore opting for "hard for heuristic algorithms" as our notion of average-case hardness makes our impossibility result weaker, but in fact our result holds even with respect to the notion of "hard for efficient on average algorithms" (as explained in section 4).

For a function $\delta(n)$, two languages L and L', and an ensemble of distributions $\mathcal{D}$ on inputs, we say that L and L' are δ-close with respect to $\mathcal{D}$ if for sufficiently large n, the measure of the set $L_n \triangle L'_n$ according to $\mathcal{D}_n$ is at most $\delta(n)$. The definition of "heuristic polynomial time with success probability $s(n)$" requires that A and L be $(1 - s(n))$-close.

2.2 Worst-case to average-case reductions

A worst-case to average-case reduction is a procedure that transforms any average-case algorithm for one problem into an algorithm that works on all inputs for another problem. The reduction is called *non-adaptive* if the reduction decides on all its queries before it makes any of them. The following definition formalizes this notion.

Definition 3. *A* nonadaptive worst-case to average-case randomized reduction *from L to $(L', \mathcal{D})$ with* average hardness *δ (in short, a δ* worst-to-average reduction*) is a family of polynomial size circuits $R = \{R_n\}$ such that on input $x \in \{0,1\}^n$ and randomness r, $R_n(x; r)$ outputs strings $y_1, \ldots, y_k$ (called* queries*) and a circuit C (called a* decider*) such that for any L^* that is δ-close to L' with respect to $\mathcal{D}$, it holds that*

$$\Pr_r[C(L^*(y_1), \ldots, L^*(y_k)) = L(x)] \geq 2/3.$$

We can also think of R as a non-adaptive oracle procedure that, when provided any L^* that is δ-close to L' as an oracle, agrees with L on every input (with probability at least $2/3$ over its internal coin tosses).

Notice that if there is a δ worst-to-average reduction from L to $(L', \mathcal{D})$, and $L \notin \text{BPP}$, then $(L', \mathcal{D}) \notin \text{Heur}_{1-\delta}\text{BPP}$. When the distribution $\mathcal{D}$ is uniform, we may denote the distributional problem $(L', \mathcal{D})$ just by L'.

Remarks on the definition.

- The choice of constant $2/3$ for the success probability of the reduction in the definition is rather arbitrary. If there exists a worst-to-average reduction R_n from L to $(L', \mathcal{D})$ that succeeds with probability $1/2 + n^{-c}$, there also exists one that succeeds with probability $1 - 2^{-n^{c'}}$ for arbitrary constants c, c'.

[3] Levin's original definition [Lev86] is formulated differently from the one we give here, but Impagliazzo [Imp95] shows that the two are essentially equivalent.

- Without loss of generality, we may also assume that the strings $y_1, \ldots, y_k$ are identically distributed. Suppose R is an arbitrary reduction R that makes k queries. We define a new reduction R' that randomly permutes the queries of R. Then each query of R' is distributed as a random query of R.
- Without loss of generality, we can fix two polynomials $k(n)$ and $m(n)$ such that for every n, when given an input x of length n, the reduction makes exactly $k(n)/m(n)$ queries of length i for every i between 1 and $m(n)$ (so that the total number of queries made is $k(n)$). This condition guarantees that for every i between 1 and $m(n)$, and every string y of length i, the probability that a random query made by the reduction equals y is exactly $1/m(n)$ times the probability that a random query made by the reduction equals y, conditioned on the length of this query being i. When working with distributions over queries, it is convenient to fix the query length; this restriction will allow us to relate statistics over queries of fixed length to statistics over queries of arbitrary length produced by the reduction.
- We used a non-uniform definition of reductions as it gives us a more general impossibility result. In particular, our result holds for uniform reductions.

2.3 Constant-round interactive protocols

We now discuss the types of protocols used in our proof, as well as certain extensions that will be used as building blocks.

Constant-round interactive protocols with advice. All the protocols in this paper are constant-round interactive protocols with polynomially long advice. An interactive protocol with advice consists of a pair of interactive machines (P, V), where P is a computationally unbounded prover and V is a randomized polynomial-time verifier which receive a common input x and advice string a. Feigenbaum and Fortnow [FF93] define the class $\mathrm{AM}^{\mathrm{poly}}$ as the class of languages L for which there exists a constant c, a polynomial p and an interactive protocol (P, V) with advice such that for every n, there exists an advice string a of length $p(n)$ such that for every x of length n, on input x and advice a, (P, V) produces an output after c rounds of interaction and

- If $x \in L$, then $\Pr[(P, V) \text{ accepts } x \text{ with advice } a] \geq 2/3$
- If $x \notin L$, then for every prover P^*, $\Pr[(P^*, V) \text{ accepts } x \text{ with advice } a] \leq 1/3$.

We observe that this definition is weaker than the definition of the class AM/poly following the Karp-Lipton notion of classes with advice, which requires that the (P, V) be a valid constant-round interactive protocol (possibly for some language other than L) for all possible settings of the advice. (We note that in both cases, the advice is accessible to both the prover and the verifier.) Even though $\mathrm{AM}^{\mathrm{poly}}$ appears to be larger than AM/poly, they are in fact both equal to NP/poly (cf. [FF93]). Owing to this, in our description of protocols we will not be concerned with the behavior of the protocol when the advice is bad.

The protocol of Feigenbaum and Fortnow uses public coins. In contrast, our protocol will use private coins. In the case of constant-round interactive protocols without advice, it is known that

private coin protocols can be simulated by public-coin protocols [GS86]. The argument extends to protocols with advice, and therefore we may drop the public coin requirement in the definition of $\mathrm{AM}^{\mathrm{poly}}$.

The existence of $\mathrm{AM}^{\mathrm{poly}}$ protocols for all of coNP implies a partial collapse of the polynomial hierarchy, as was also observed in [FF93]: By the above observations, the assumption $\mathrm{coNP} \subseteq \mathrm{AM}^{\mathrm{poly}}$ implies $\mathrm{coNP} \subseteq \mathrm{NP/poly}$, and by a result of Yap [Yap83], this gives $\Sigma_3 = \Pi_3$.

Protocols with shared auxiliary input. When applying two protocols in sequence, the second protocol has access to the transcript of the interaction from the first protocol. To allow access to this transcript, we extend our definition of interactive protocol to include a shared auxiliary input. This shared auxiliary input comes with a promise Υ, which may depend on the actual input. The completeness and soundness conditions are required to hold for all auxiliary inputs satisfying the promise. In the case of sequential composition of two protocols, the promise of the second protocol includes the set of all transcripts that are not rejecting for the first protocol. The running time of the verifier in a protocol with shared auxiliary input is measured with respect to the length of the concatenation of the actual input and the shared auxiliary input.

Protocols with private verifier input. In a protocol with private verifier input, the verifier is given, in addition to the input x, a private input r not known to the prover. The input r will be a "secret" of the verifier–a random string, uniformly distributed among a range of secrets that may depend on the input x. We represent the range of secrets by an NP-relation H: A secret r for input x is chosen uniformly among all r such that $(x; r)$ satisfies H.

In our application, the protocol will be applied to instances of promise problems (see [ESY84, Gol05]) instead of languages. For this reason, we state a definition of constant-round protocols with private verifier input for promise problems.

Definition 4. *An* interactive protocol with private verifier input *consists of a polynomial-time verifier V, an unbounded prover P, and an* NP *relation H. A promise problem $\Pi = (\Pi_Y, \Pi_N)$ admits such a protocol with completeness c and soundness s if:*

1. *For all $x \in \Pi_Y$, $\Pr[(P, V(r))(x)$ accepts$] \geq c$.*
2. *For all $x \in \Pi_N$ and every prover P^*, $\Pr[(P^*, V(r))(x)] \leq s$.*

In both cases, the randomness is taken over V and over r chosen uniformly from all strings that satisfy $(x; r) \in H$.

Notice that this definition extends the standard notion of proof system without a private input, as we can specialize the definition to an NP relation H that mandates a unique choice of r for every x. The definition can be naturally extended in the case of protocols with shared auxiliary input. For such protocols to be sequentially composable, we must require that the private input of the verifier is independent of the shared auxiliary input, conditioned on the actual input.

Parallel repetition of two-round protocols. Suppose we are given instances $x_1, \ldots, x_n$ of promise problem Π in $\mathrm{AM}^{\mathrm{poly}}$. We want a $\mathrm{AM}^{\mathrm{poly}}$ protocol that distinguishes between the case when all $x_i \in \Pi_Y$ and the case when at least one $x_i \in \Pi_N$. A natural approach is to run n independent instantiations of the protocol for Π in parallel, and accept if all of them accept. Intuitively, if the protocol for Π has completeness $1-\epsilon$ and soundness δ, we expect the parallel protocol to have completeness $1-n\epsilon$ and soundness δ. This worsens the completeness of the original protocol, while leaving the soundness essentially unchanged.

One way to improve the soundness is the following. Suppose that we could settle for distinguishing between the case when all $x_i \in \Pi_Y$ and the case when $x_i \in \Pi_N$ *for at least t of the x_is*. Intuitively, this relaxation makes the work required of a cheating prover much more demanding: Such a prover is now trying to convince the verifier to accept an instance in which at least t of the x_is are "no" instances of Π. We expect such a prover to have success probability at most δ^t.

We prove that this is indeed the case for public-coin two-round protocols (which is sufficient for our application), even for proof systems with private verifier input. We begin by describing the promise problem intended to be solved by parallel composition. We then define parallel composition for two-round protocols with private verifier input, and prove that parallel composition solves the intended problem.[4]

Given a promise problem $\Pi = (\Pi_Y, \Pi_N)$, we define the *n-wise repetition of Π with threshold t* to be the promise problem $\Pi^{n,t} = (\Pi_Y^{n,t}, \Pi_N^{n,t})$ as follows:

$$\Pi_Y^{n,t} = \{(x_1, \ldots, x_n) : x_i \in \Pi_Y \text{ for all } i\}$$
$$\Pi_N^{n,t} = \{(x_1, \ldots, x_n) : x_i \in \Pi_N \text{ for at least } t \text{ values of } i.\}$$

Suppose (P, V, H) is a k round protocol with private verifier input and with advice. We define its n-fold parallel composition as the k round protocol (P^n, V^n, H^n) with private verifier input, where

- V^n is the machine that, on input $(x_1, \ldots, x_n)$ and private verifier input $(r_1, \ldots, r_n)$, simulates n independent runs of V, where the ith run takes input x_i, private verifier input r_i, uses randomness independent of all other runs, and responds according to the next message function of V given the transcript of messages generated by the ith run of (P, V) so far. At the end of the interaction, V^n accepts if the transcripts produced by *all* the runs are accepting.
- P^n is the machine, that, on input $(x_1, \ldots, x_n)$, simulates n runs of P, where the ith run takes input x_i and responds according to the next message function of P given the transcript of messages generated by the ith run of (P, V) so far.
- H^n is defined as follows: $((x_1, \ldots, x_n); (r_1, \ldots, r_n)) \in H^n$ if and only if $(x_i; r_i) \in H$ for all $1 \leq i \leq n$.

Lemma 5. *Suppose (P, V, H) is a two-round protocol (where the first message is sent by the verifier) with private verifier input for promise problem Π with completeness $1-\epsilon$ and soundness δ. Moreover,*

[4] Goldreich [Gol99, Appendix C.1] proves that parallel composition has the desired completeness and soundness errors for private-coin protocols with arbitrary round complexity, but without private verifier input. His proof easily extends to our setting, but for simplicity we present a self-contained proof here.

suppose that the message sent by V contains all of V's coin tosses. Then (P^n, V^n, H^n) is a protocol for $\Pi^{n,t}$ with completeness $1 - n\epsilon$ and soundness δ^t.

Proof. The completeness of (P^n, V^n, H^n) follows by taking a union bound over the n runs of (P, V, H). To argue soundness, suppose that $\mathbf{x} = (x_1, \ldots, x_n) \in \Pi_N^{n,t}$. Without loss of generality, assume that specifically $x_1, \ldots, x_t \in \Pi_N$. For an input x and private verifier input r, define $B_{x,r}$ as the set of all messages μ sent by V on input (x, r) for which there exists a response ν that makes V accept.

Consider an arbitrary prover P^* that interacts with V^n. Suppose that V^n receives private verifier input $\mathbf{r} = (r_1, \ldots, r_n)$, and sends the message $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$. In the second round, P^* responds with the message $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_n)$. Note that

$$\Pr_{\mathbf{r},\boldsymbol{\mu}}[V^n(\mathbf{x}, \mathbf{r}, \boldsymbol{\mu}, \boldsymbol{\nu}) \text{ accepts}] \leq \Pr_{\mathbf{r},\boldsymbol{\mu}}[\mu_i \in B_{x_i,r_i} \text{ for all } 1 \leq i \leq t].$$

The independence of the verifier strategies on different runs implies that for every i, the event $\mu_i \in B_{x_i,r_i}$ is independent of any function of μ_j, r_j, for $j \neq i$. Therefore

$$\Pr_{\mathbf{r},\boldsymbol{\mu}}[\mu_i \in B_{x_i,r_i} \text{ for all } 1 \leq i \leq t] = \prod_{i=1}^{t} \Pr_{r_i,\mu_i}[\mu_i \in B_{x_i,r_i}] \leq \delta^t,$$

by the soundness of (P, V, H) for promise problem Π. □

The lemma and the proof also extend to protocols with shared auxiliary input (in addition to the private verifier input).

2.4 Lower and upper bound protocols

We now outline two protocols, used for proving approximate bounds on the number of accepting inputs of a circuit that will be used as components in our constructions. The lower bound protocol of Goldwasser and Sipser [GS86] is used to prove an approximate lower bound on the number of accepting inputs of a circuit C. The upper bound protocol of Fortnow [For87] (also used by Aiello and Håstad [AH91]) is used to prove an approximate upper bound on the same quantity, when the verifier is given private access to a random accepting input of the circuit. We stress that our version of the upper bound protocol is somewhat different from the protocols in [For87, AH91], as for our application we need a better approximation than in the original protocols, but we can allow for a much larger soundness error. The lower and upper bound protocols will allow the verifier to check whether queries made by the reduction are light or heavy.

We state the completeness and soundness conditions of the protocols, and provide proof sketches in Appendix A.2.

The Lower Bound Protocol. The lower bound protocol solves the following promise problem, which we denote $\Pi_{LB,\epsilon}$ (where $\epsilon > 0$):

Inputs. (C, s), where $C : \{0,1\}^m \to \{0,1\}$ is a circuit, $0 \leq s \leq 2^m$.

Shared auxiliary input. $\delta, \epsilon > 0$ (represented in unary), where δ is a parameter that controls the completeness and soundness errors, and ϵ controls the precision of the lower bound.

Yes instances. (C, s) such that $|C^{-1}(1)| \geq s$.

No instances. (C, s) such that $|C^{-1}(1)| \leq (1-\epsilon)s$.

The protocol. On input (C, s) and shared auxiliary input (δ, ϵ):

1 Verifier: Set $k = \lceil 9/\delta\epsilon^2 \rceil$. Choose a pairwise independent hash function $h : \{0,1\}^m \to \Gamma$ at random, where $|\Gamma| = \lfloor s/k \rfloor$, and send h to the prover.

2 Prover: Send a list $r_1, \ldots, r_l \in \{0,1\}^m$, where $l \leq (1+\epsilon/3)k$. An honest prover sends all r_i such that $C(r_i) = 1$ and $h(r_i) = 0$.

3 Verifier: If $C(r_i) \neq 1$ for any i, reject. If $|l - k| > \epsilon k/3$, reject. If $h(r_i) \neq 0$ for any i, reject. Otherwise, accept.

An alternative version of this protocol that achieves somewhat better parameters appears in work by Goldreich and others [GVW01]. The protocol presented here parallels the upper bound protocol, described below, more closely.

Lemma 6. *The lower bound protocol is a protocol for $\Pi_{LB,\epsilon}$ with completeness $1-\delta$ and soundness δ.*

In our applications we will need to apply the lower bound protocol on many instances in parallel, and be guaranteed that all runs of the protocol are correct with high probability. For this reason, we resort to Lemma 5, observing that the lower bound protocol satisfies the hypothesis of this lemma. Applying Lemma 5 for $t = 1$ and setting $\delta = \epsilon/n$ yields a parallel lower bound protocol for $\Pi^{n,1}_{LB,\epsilon}$ with completeness $1 - \epsilon$ and soundness ϵ.[5]

Corollary 7 (Parallel lower bound protocol)**.** *For every n and $\epsilon > 0$ there exists a constant round protocol for $\Pi^{n,1}_{LB,\epsilon}$ (with shared auxiliary input ϵ, represented in unary) with completeness $1-\epsilon$ and soundness ϵ.*

The Upper Bound Protocol. The upper bound protocol solves the following promise problem, which we denote by $\Pi_{UB,\epsilon}$ (where $\epsilon > 0$):

Inputs. (C, s), where $C : \{0,1\}^m \to \{0,1\}$ is a circuit, $0 \leq s \leq 2^m$.

Shared auxiliary input. $\delta, \epsilon > 0$ (represented in unary), where δ is a parameter that controls the completeness error and the soundness error. In contrast to the lower bound protocol, ϵ controls the precision of the protocol and also affects the soundness.

[5] For this setting of parameters, ϵ controls both the precision of the approximate lower bound and the completeness and soundness errors. Had we wished to do so, we could have used independent parameters for the precision and for the completeness / soundness errors, though this is not necessary for our application. In contrast, in the parallel upper bound protocol presented below, the precision of the protocol and the soundness error are intricately related.

Yes instances. (C, s) such that $|C^{-1}(1)| \leq s$.

No instances. (C, s) such that $|C^{-1}(1)| \geq (1+\epsilon)s$.

The upper bound protocol is a protocol with private verifier input: The verifier is provided a random sample r of $C^{-1}(1)$, not known to the prover.

Private verifier input. A string $r \in S$. (Notice that the relation $\{((C, s); r) : C(r) = 1\}$ is an NP relation.)

The protocol. On input (C, s), shared auxiliary input (δ, ϵ), and private verifier input r:

1 Verifier: Set $k = \lceil 9/\delta\epsilon^2 \rceil$. Choose a 3-wise independent hash function $h : \{0,1\}^m \to \Gamma$ at random, where $|\Gamma| = \lfloor (s-1)/k \rfloor$ and send the pair $(h, h(r))$ to the prover.

2 Prover: Send a list $r_1, \ldots, r_l \in \{0,1\}^m$, where $l \leq (1+\epsilon/3)k$. An honest prover sends all r_i such that $C(r_i) = 1$ and $h(r_i) = h(r)$.

3 Verifier: If $C(r_i) \neq 1$ for any i, reject. If $l > (1+\epsilon/3)k$ or $r \notin \{r_1, \ldots, r_l\}$, reject. Otherwise, accept.

Lemma 8. *The upper bound protocol is a protocol with private verifier input for $\Pi_{UB,\epsilon}$ completeness $1-\delta$ and soundness $1-\epsilon/6+\delta$.*

Setting $\delta = o(\epsilon)$, this yields a protocol with completeness $1-o(\epsilon)$ and soundness $1-\Omega(\epsilon)$, giving a narrow gap. To improve the soundness of the protocol, which is necessary for our application, we apply parallel repetition. In particular, fixing $\delta = o(\epsilon/n)$ and setting $t = \omega(1/\epsilon)$ in Lemma 5 yields a protocol for $\Pi^{n,t}_{UB,\epsilon}$ with completeness $1-o(1)$ and soundness $(1-\epsilon/6+o(\epsilon/n))^t = o(1)$. More generally, we have a parallel upper bound protocol for $\Pi^{n,t}_{UB,\epsilon}$ with the following parameters:

Corollary 9 (Parallel upper bound protocol)**.** *For every n and $\epsilon > 0$ there exists a constant round protocol with private verifier input (and shared auxiliary input ϵ, represented in unary) such that for every $t > 0$ the protocol decides $\Pi^{n,t}_{UB,\epsilon}$ with completeness $1-\epsilon$ and soundness $(1-\epsilon/9)^t$.*

3 The protocols

In this section we describe the constant-round interactive protocols that will constitute the building blocks of our main protocol. The order in which the protocols are presented follows the order in which they will be composed sequentially, which is opposite from the description in the Introduction. Recall that we need protocols that accomplish each of the following tasks:

- For a fixed input x, estimate the probability that a random query of a given length produced by the worst-to-average reduction on input x is light (that is, the probability of it being produced by the reduction is smaller than a specified threshold). We consider the following more abstract version of the problem: Given a circuit C, estimate the fraction of heavy outputs of C. The heavy samples protocol, described in Section 3.1, solves this problem.

- For a fixed input x, estimate the probability that a random query of a given length produced by the worst-to-average reduction on input x is both light and a "yes" instance. Abstractly, we can think of this problem as follows. We model the worst-to-average reduction as a sampler circuit C and the set of "yes" instances of a given length as a nondeterministic circuit V. As auxiliary input, we are given the fraction of accepting inputs for V as well as the probability that a random output of C is heavy. The task is to construct a protocol that estimates the probability that a random output of C is both light and accepting for V. The hiding protocol, described in Section 3.2, accomplishes this task.

- For a fixed input x, simulate an "approximate membership oracle" for queries made by the reduction on input x. This calls for a protocol for the following task: We are given a "querier" Q, describing an instantiation of the reduction on x, and an NP verifier V for "yes" instances. The following promise holds: When provided an oracle for the set

  $$S_{Q,V} = \{y : V \text{ accepts } y \text{ and } y \text{ is a light query of } Q\}$$

  Q either outputs "yes" with very high probability, or outputs "no" with very high probability (these cases corresponding to the reduction saying "$x \in L$" and "$x \notin L$", respectively). The simulation protocol, described in Section 3.3, distinguishes between these two cases when given as auxiliary inputs the fraction of heavy queries of Q and the fraction of "yes" instances of $S_{Q,V}$.

A note on approximations. The protocols described in this Section cannot be expected to certify the exact values of the probabilities in question, but can only obtain approximations thereof. Intuitively, we will think of a protocol as computing an approximation of p if, for arbitrary ϵ, the protocol runs in time polynomial in $1/\epsilon$ and distinguishes between instances whose probability is p and instances whose probability is outside the interval $(p-\epsilon, p+\epsilon)$. Indeed, additive approximations are sufficient in all our applications.

These protocols also take as inputs (actual and auxiliary) probabilities of various events. We make the assumption that these probabilities are specified exactly, but in fact the completeness and soundness of the protocols are unaffected even if only approximations of these quantities were provided. The quality of approximation required is, in all cases, a fixed inverse polynomial function of the input length.

Statistics. We use the following formulation of the Law of Large Numbers to obtain sampling estimates for various probabilities. For completeness we provide a proof in Appendix A.1.

Lemma 10 (Sampling bound)**.** *Let Ω be a sample space, $\epsilon, \eta < 1$, $T \subseteq \Omega$ and $\mathcal{D}$ a distribution on Ω. Suppose that S is a random sample of Ω consisting of at least $3\log(2/\eta)/\epsilon^3$ elements chosen independently at random from the distribution $\mathcal{D}$. Then with probability at least $1-\eta$,*

$$\big||T \cap S|/|S| - \mathcal{D}(T)\big| \leq \epsilon.$$

High probability means probability $1 - o(1)$, where the $o(\cdot)$ notation is in terms of the length of the input of the protocol.

The notation $A \triangle B$ is for the symmetric difference of sets A and B. We use standard set facts about symmetric difference.

3.1 The heavy samples protocol

In this section we describe the protocol used for estimating the fraction of heavy samples generated by a circuit C. Recall that a string y is *α-heavy* for distribution $\mathcal{D}$ on $\{0,1\}^m$ if $\mathcal{D}(y) \geq \alpha 2^{-m}$. Given a distribution $\mathcal{D}$, the probability that a random sample of $\mathcal{D}$ is α-heavy is given by the quantity

$$h_{\mathcal{D},\alpha} = \Pr_{y\sim\mathcal{D}}[\mathcal{D}(y) \geq \alpha 2^{-m}].$$

The problem we are considering is the following: Given a circuit $C : \{0,1\}^n \to \{0,1\}^m$, a threshold α, a heaviness estimate p, and an error parameter ϵ, we want a protocol that accepts when $p = h_{\mathcal{D}_C,\alpha}$ and rejects when $|p - h_{\mathcal{D}_C,\alpha}| > \epsilon$. Here, as throughout the section, $\mathcal{D}_C$ denotes the output distribution of the circuit C.

A natural approach is to estimate the unknown $h_{\mathcal{D}_C,\alpha}$ by sampling: Suppose that for a random sample $y \sim \mathcal{D}_C$ we could decide with good probability if the sample was α-heavy or α-light. Then, given $k = O(1/\epsilon^3)$ samples $y_1, \ldots, y_k$ generated by running C on independent inputs, we could estimate $h_{\mathcal{D}_C,\alpha}$ as the fraction of samples that are α-heavy, and sampling bounds would guarantee that the answer is correct with good probability.

However, in general we have no way of efficiently deciding whether a sample $y \sim \mathcal{D}_C$ is heavy or light. However, we have at our disposal the upper and lower bound protocols of Section 2.4. For each sample y_i, the verifier first asks the prover to say if sample y_i is α-heavy or α-light. Then the prover is asked to prove the claims for the heavy samples using the lower bound protocol, and the claims for the light samples using the upper bound protocol. In both cases, the claims concern the size of the set $C^{-1}(y_i)$, which is the restriction of an NP set on $\{0,1\}^m$.

There are several difficulties with implementing this approach. First, the upper bound protocol requires the verifier to have a secret, random preimage r_i of the set $C^{-1}(y_i)$. Fortunately, the verifier obtains this secret for free in the course of generating y_i. When generating y_i, the verifier in fact chooses a random $r_i \in \{0,1\}^n$ and sets $y_i = C(r_i)$, so this r_i (which is kept hidden from the prover) is indeed a random pre-image of y_i.

Another difficulty is that the upper bound protocol guarantees an ϵ deviation from the true value of $C^{-1}(y_i)$ only with probability $1 - O(\epsilon)$. Therefore the prover has a good chance of getting away with a false upper bound claim on any particular y_i. But how many times can the prover play this trick? If the prover decides to submit t false upper bound claims, its success probability quickly falls to $(1 - O(\epsilon))^t$ (see Corollary 7), so for $t = \omega(1/\epsilon)$ the risk of detecting a false upper bound claim becomes quite high for the prover. On the other hand, $O(1/\epsilon)$ false upper bound claims make no real difference for the verifier. In the end, the verifier uses these claims to compute its estimate of $h_{\mathcal{D}_C,\alpha}$, so any set of $O(1/\epsilon)$ false claims among k samples will only change its estimate by $O(k/\epsilon) = O(\epsilon^2)$, much less than the tolerated deviation ϵ.

The final difficulty stems from the fact that both the upper and lower bound protocols are approximate. To illustrate this issue, consider the case of a circuit C for which the distribution $\mathcal{D}_C$ is

close to α-flat: Namely, every $y \in \{0,1\}^m$ has probability either 0 or $(1 \pm \epsilon)\alpha 2^{-m}$ under $\mathcal{D}_C$. Now for *every* sample y_i of the verifier, it happens that y_i is approximately both α-heavy and α-light. Hence the verifier has no soundness guarantee about the prover's claims for any y_i.

This issue appears quite difficult to resolve. We sidestep it by weakening the requirement on the protocol. Instead of requiring soundness for all α, we settle for soundness for a *random* choice of α. This avoids the "flatness" issue, because an almost flat distribution is very unlikely to be close to α-flat for a random α. More generally, given any distribution $\mathcal{D}_C$, if α is assigned a random value among any set of $1/\epsilon$ values that are spaced by a factor of at least $1 + 2\epsilon$ apart, then the expected probability mass under $\mathcal{D}_C$ of samples that are both $(1-\epsilon)\alpha$-heavy and $(1+\epsilon)\alpha$-light can be at most ϵ. Therefore, the fraction of samples for which the verifier fails to obtain a soundness guarantee cannot be much more than $O(\epsilon)$.

Choosing a heaviness threshold. We formalize the last observation in the following claim. The claim is also used in Sections 3.2 and 3.3. For every integer α_0 and fraction $\delta > 0$, define the distribution:

$$\mathcal{A}_{\alpha_0,\delta} = \text{Uniform distribution on } \{\alpha_0(1+3\delta)^i : 0 \le i \le 1/\delta\}. \tag{1}$$

Observe that every value in $\mathcal{A}_{\alpha_0,\delta}$ is in the range $[\alpha_0, e^3 \cdot \alpha_0]$. Since the intervals $((1-\delta)\alpha, (1+\delta)\alpha)$ are pairwise disjoint over the various $\alpha \in \mathcal{A}_{\alpha_0,\delta}$, the following result holds:

Claim 11 (Choosing a random threshold)**.** *For every $\alpha_0 > 0$ and $0 < \delta < 1/3$, and every distribution $\mathcal{D}$ on $\{0,1\}^m$,*

$$\mathrm{E}_{\alpha \sim \mathcal{A}_{\alpha_0,\delta}}\left[\Pr_{y\sim\mathcal{D}}[\mathcal{D}(y) \in ((1-\delta)\alpha 2^{-m}, (1+\delta)\alpha 2^{-m})]\right] \le \delta.$$

3.1.1 The protocol

We formalize the notion of a "protocol that works for a random heaviness threshold α" by defining a family of problems $\{\Pi_{HEAVY,\alpha}\}$, parametrized by the threshold α, and requiring that the protocol be complete and sound for a random problem in this family.

Inputs. (C, p, ϵ), where $C : \{0,1\}^n \to \{0,1\}^m$ is a circuit, and $p \in [0,1]$ is a probability, and $\epsilon > 0$ is an error parameter represented in unary.

Shared auxiliary input. (α, δ), where α is a threshold parameter represented in unary and $0 < \delta < 1/3$ is an error parameter.

Yes instances. (C, p, ϵ) such that $h_{\mathcal{D}_C,\alpha} = p$.

No instances. (C, p, ϵ) such that $|h_{\mathcal{D}_C,\alpha} - p| > \epsilon$.

The heavy samples protocol. On input (C, p, ϵ) and shared auxiliary inputs α and δ:

1. Verifier: Set $k = \lceil 3 \cdot 16^3 \log(2/\delta)/\delta^3 \rceil$. Choose $r_1, \ldots, r_k \sim \{0,1\}^n$. Compute $y_j = C(r_j)$. Send $y_1, \ldots, y_k$ to the prover.
2. Prover: Send a partition (H, L) of $[k]$. An honest prover sets $H = \{i : \mathcal{D}_C(y_i) \ge \alpha 2^{-m}\}$ and $L = \{i : \mathcal{D}_C(y_i) < \alpha 2^{-m}\}$.

3. Verifier and Prover: Run the parallel upper bound protocol (see Section 2.4) with auxiliary input $(r_i : i \in L)$ with shared auxiliary input (error parameter) δ for the claim "$|C^{-1}(y_i)| < \alpha 2^{n-m}$ for all $i \in L$."

   Run the parallel lower bound protocol (see Section 2.4) with shared auxiliary input (error parameter) δ for the claim "$|C^{-1}(y_i)| \geq \alpha 2^{n-m}$ for all $i \in H$."

   Accept iff $\big||H|/k - p\big| \leq \epsilon/2$.

3.1.2 Analysis of the protocol

Lemma 12. *For every integer α_0 and fractions ϵ, δ, with probability $1 - O(\delta/\epsilon)$ over α chosen uniformly from $\mathcal{A}_{\alpha_0,\delta}$, the heavy samples protocol (with input (C, p, ϵ) and auxiliary input (α, δ) satisfying the promise) is a protocol for $\Pi_{HEAVY,\alpha}$ with completeness $1 - O(\delta)$ and soundness $O(\delta)$.*

Proof. We denote by H' and L' the set of α-heavy and α-light samples, respectively:

$$H' = \{i : \mathcal{D}_C(y_i) \geq \alpha 2^{-m}\} \text{ and } L' = \{i : \mathcal{D}_C(y_i) < \alpha 2^{-m}\}.$$

The honest prover always chooses $H = H'$ and $L = L'$.

By the sampling bound, for every prover strategy, with probability $1 - O(\delta)$ over the randomness of the verifier, the fraction of α-heavy samples among $y_1, \ldots, y_k$ should closely approximate $h_{\mathcal{D}_C,\alpha}$, and in particular,

$$\big||H'|/k - h_{\mathcal{D}_C,\alpha}\big| \leq \epsilon/6. \tag{2}$$

Completeness. Completeness (for arbitrary α) follows from high probability estimate (2), together with completeness of the parallel lower bound protocol for $\Pi^{|L|,1}_{LB,\delta}$ (see Corollary 7) and completeness of the parallel upper bound protocol for $\Pi^{|H|,\cdot}_{UB,\delta}$ (see Corollary 9).

Soundness. Fix an $\alpha \sim \mathcal{A}_{\alpha_0,\delta}$ such that

$$\Pr_{y\sim\mathcal{D}_C}\big[\mathcal{D}_C(y) \in ((1-\delta)\alpha 2^{-m}, (1+\delta)\alpha 2^{-m})\big] \leq \epsilon/16.$$

By Claim 11 and Markov's inequality, this holds with probability $1 - O(\delta/\epsilon)$ for a random α in $\mathcal{A}_{\alpha_0,\delta}$. For such a choice of α, let B denote the set of samples that are both $(1-\delta)\alpha$-heavy and $(1+\delta)\alpha$-light, that is

$$B = \{i : \mathcal{D}_C(y_i) \in ((1-\delta)\alpha 2^{-m}, (1+\delta)\alpha 2^{-m})\}.$$

By the sampling bound, the number of samples in B is not much larger than $\epsilon/16$ with high probability over the randomness of the verifier. Indeed,

$$\Pr\big[|B| > \epsilon k/8\big] = \Pr\big[|B|/k - \epsilon/16 > \epsilon/16\big] \leq \delta.$$

Now fix a prover strategy for which the verifier accepts instance (C, p, ϵ) with probability $\omega(\delta)$. Then there exists a setting of the verifier's randomness for which $\big||H|/k - p\big| \leq \epsilon/2$ (by the last step of the verifier), $\big||H'|/k - h_{\mathcal{D}_C,\alpha}\big| \leq \epsilon/6$ (by high probability estimate (2)), and the following conditions hold:

- For $t = \lceil \log(1/\delta)/\delta \rceil$ all but $t + \epsilon k/8$ samples in L are α-light, that is, $|L - L'| \leq t + \epsilon k/8$. Indeed, this is an event of probability $1 - O(\delta)$ over the randomness of the verifier: By soundness of the parallel upper bound protocol for $\Pi_{UB,\delta}^{|L|,t}$, fewer than t of the samples in L are $(1+\delta)\alpha$-heavy. Moreover, the number of samples in L that are α-heavy but $(1+\delta)\alpha$-light is upper bounded by the size of B. It follows that with probability $1-O(\delta)$, $|L-L'| \leq t+\epsilon k/8$.
- All but $\epsilon k/8$ samples in H are α-heavy, that is, $|H - H'| \leq \epsilon k/8$. This is also an event of probability $1 - O(\delta)$ over the randomness of the verifier: By soundness of the parallel lower bound Protocol for $\Pi_{LB,\delta}^{|H|,1}$, none of the samples in H are $(1-\delta)\alpha$-light. Moreover, the number of samples in H that are α-light but $(1-\delta)\alpha$-heavy is upper bounded by the size of B. It follows that with probability $1 - O(\delta)$, $|H - H'| \leq \epsilon k/8$.

It follows that

$$\big||H| - |H'|\big| \leq |H - H'| + |H' - H| = |H - H'| + |L - L'| \leq t + \epsilon k/4 < \epsilon k/3.$$

Therefore,

$$|h_{\mathcal{D}_C,\alpha} - p| \leq \big|h_{\mathcal{D}_C,\alpha} - |H'|/k\big| + \big||H'|/k - |H|/k\big| + \big||H|/k - p\big| < \epsilon/6 + \epsilon/3 + \epsilon/2 = \epsilon.$$

So, (C, p, ϵ) is a "yes" instance of $\Pi_{HEAVY,\alpha}$. □

3.2 The hiding protocol

In this section we describe the protocol for estimating the probability that a random sample generated by a circuit $C : \{0,1\}^n \to \{0,1\}^m$ is both a light sample and a "yes" instance of some NP language V. Let us denote by $\mathcal{D}_C$ the distribution of outputs of the circuit C and by $\mathcal{U}$ the uniform distribution on $\{0,1\}^m$. We assume that we are given as advice the probability p_Y that a random sample in $\{0,1\}^m$, is a "yes" instance of V.

For starters, let us assume that we are guaranteed that the distribution $\mathcal{D}_C$ is α-smooth (for some reasonably small α), that is, no output of C is α-heavy. Let us consider the following experiment. We choose among distributions $\mathcal{D}_C$ and $\mathcal{U}$ by flipping a coin biased towards $\mathcal{D}_C$ with probability b (the value of interest is around $b = 1/\alpha$), then generate a sample y according to either $\mathcal{D}_C$ or $\mathcal{U}$ depending on the outcome of the coin. We then give the sample y to the prover and ask which distribution it came from.

The best strategy for the prover to guess the origin of the sample is a maximum likelihood test: Compute the ratio $\mathcal{D}_C(y)/\mathcal{U}(y)$, and declare "$\mathcal{D}_C$" if the ratio exceeds $1/b$, and "$\mathcal{U}$" otherwise. However, when $b < 1/\alpha$, the maximum likelihood test will always guess that the sample came from $\mathcal{U}$. This effectively gives the verifier the ability to hide samples from $\mathcal{D}_C$ among samples from $\mathcal{U}$.

Suppose now that the verifier wants to determine membership in $V \cap \{0,1\}^m$ for a random sample from $\mathcal{D}_C$, but it has only the ability to determine membership for random samples from $\mathcal{U}$. The verifier can then use the prover as follows: Generate $\omega(\alpha)$ samples by choosing each one independently. The sample size is large enough so that at least one of the samples, call it z, will originate from $\mathcal{D}_C$. The prover is then asked to determine membership in V for all the samples. The verifier

then checks if the prover determined membership in V correctly for all the samples coming from $\mathcal{U}$. If this is the case, then chances are that the prover also determined membership in V correctly for z, since this sample was hidden among all the other ones.

In our case, the verifier has no way of determining membership in V on samples from $\mathcal{U}$. Instead, it only knows that the fraction of samples from $\mathcal{U}$ that fall in V should be roughly p_Y. Recall that the verifier is interested in determining the fraction $g_{\mathcal{D}_C,V}$ of samples from $\mathcal{D}_C$ that fall in V. It is tempting to try the following protocol: The verifier and prover run the hiding procedure on a sufficiently large sample. The verifier then checks that the fraction of samples originating from $\mathcal{U}$ claimed to be in V by the prover is within a small enough deviation δ of p_Y. If this is the case, the verifier estimates $g_{\mathcal{D}_C,V}$ as the fraction of samples originating from $\mathcal{D}_C$ that are claimed to be in V by the prover.

It is not difficult to see that this protocol is not sound: A prover can "convince" the verifier, for instance, that $g_{\mathcal{D}_C,V} = p_Y$. To do so, for each sample the prover answers independently "yes" with probability p_Y and "no" with probability $1 - p_Y$ about the membership of this sample in V.

However, since V is an NP set, the verifier can impose an additional requirement: Every time the prover makes a "yes" claim for a sample, an NP certificate for membership in V of the sample must be provided. The prover's cheating power now becomes *one-sided* as it can no longer provide a "yes" answer for a sample that is not in V; the only way the prover can now cheat is by providing "no" answers for samples in V. However, if the prover supplies such false answers on more than an $O(\delta)$ fraction of the samples (where $\delta > 0$ is an error parameter), it is likely that most of these falsely answered samples originated from $\mathcal{U}$, because the samples originating from $\mathcal{U}$ comprise an overwhelming majority of all the samples. Therefore it is likely that the fraction of samples from $\mathcal{U}$ claimed "yes" by the prover is smaller than $p_Y - \delta$. On the other hand, statistically it is very unlikely that fewer than a $p_Y - \delta$ fraction of samples from $\mathcal{U}$ are in V. This prevents the prover from providing false answers on more than an $O(\delta)$ fraction of *all* the samples. Since the number of samples from $\mathcal{D}_C$ is roughly a b-fraction of the total number of samples, the prover in particular cannot provide false answers on more than an $O(\delta/b)$ fraction of samples originating from $\mathcal{D}_C$. Therefore a cheating prover is unlikely to skew the verifier's estimate of $g_{\mathcal{D}_C,V}$ by more than $O(\delta/b)$. Setting $\delta = \epsilon b$ allows the verifier to obtain an additive ϵ approximation of $g_{\mathcal{D}_C,V}$ in time polynomial in $1/\epsilon$, α, and the sizes of C and V.

Handling the heavy samples. Notice that if we drop the smoothness restriction on $\mathcal{D}_C$, this argument fails because very heavy samples cannot be effectively hidden among uniform samples. However, our goal is to merely estimate the probability that a sample from $\mathcal{D}_C$ is both in V and light. To do so, we run the protocol as for smooth distributions, initially ignoring the heavy samples. The soundness of that protocol still shows that the prover must have answered most of the light samples from $\mathcal{D}_C$ correctly. What remains to be done is to weed out the heavy samples. By the protocol from Section 3.1, the verifier can estimate the fraction p_H of heavy samples. (To avoid duplication, we assume the protocol here is given p_H as auxiliary input.) At this point, the verifier reveals its sample to the prover, and asks the prover to give lower bound proofs for a $p_H - \epsilon$ fraction of samples from $\mathcal{D}_C$. The prover's cheating power here is also one-sided: By soundness of the parallel lower bound protocol, the prover cannot claim any of the light samples as heavy, so it can only cheat by claiming that some heavy samples are light. However, since the prover is

required to provide lower bound proofs for a $p_H - \epsilon$ fraction of samples from $\mathcal{D}_C$, and the number of truly heavy samples from $\mathcal{D}_C$ is likely to be upper bounded by $p_H + \epsilon$, the prover cannot cheat on more than an $O(\epsilon)$ fraction of the samples from $\mathcal{D}_C$. Therefore a cheating prover cannot skew the verifier's estimate of $g_{\mathcal{D}_C,V}$ by more than an additive term of $O(\epsilon)$.

We encounter the same issue regarding the choice of α as in the protocol for heavy samples: If too many samples from $\mathcal{D}_C$ have probability about α, the lower bound protocol provides no soundness guarantee. As in Section 3.1, we sidestep this problem by choosing a random α and arguing soundness with high probability over α.

3.2.1 The protocol

We give a protocol for the following family of promise problems, which we denote by $\{\Pi_{HIDE,\alpha}\}$.

Inputs. (C, V, p, ϵ), where $C : \{0,1\}^n \to \{0,1\}^m$ is a circuit, $V : \{0,1\}^m \times \{0,1\}^l \to \{0,1\}$ is a nondeterministic circuit,[6] $p \in [0,1]$ is a probability, and $\epsilon > 0$ is an error parameter represented in unary.

Shared auxiliary input. $(\alpha, \delta, p_Y, p_H)$, where $\alpha > 0$ is a threshold integer represented in unary, $0 < \delta < 1/3$ is an error parameter represented in unary, and p_Y, p_H satisfy:

$$p_Y = \Pr_{y \sim \mathcal{U}}[y \in V] \quad \text{and} \quad |p_H - p'_H| < \epsilon/32, \tag{3}$$

where $p'_H = \Pr_{y \sim \mathcal{D}_C}[\mathcal{D}_C(y) \geq \alpha 2^{-m}]$.

Yes instances. (C, V, p, ϵ) such that $p = g_{\mathcal{D}_C,V,\alpha}$, where

$$g_{\mathcal{D}_C,V,\alpha} = \Pr_{y \sim \mathcal{D}_C}[\mathcal{D}_C(y) < \alpha 2^{-m} \text{ and } y \in V].$$

No instances. (C, V, p, ϵ) such that $|p - g_{\mathcal{D}_C,V,\alpha}| > \epsilon$.

The hiding protocol. On input (C, V, p, ϵ) and auxiliary input $(\alpha, \delta, p_Y, p_H)$:

1. Verifier: Set $b = \delta/\alpha$ and $k = \lceil 6 \cdot 32^3 \log(2/\delta)/(b\epsilon)^3 \rceil$. Choose a set $T_{\mathcal{D}_C} \subseteq [k]$ by assigning each element of $[k]$ to $T_{\mathcal{D}_C}$ independently with probability b. Let $T_{\mathcal{U}} = [k] - T_{\mathcal{D}_C}$.

   Choose strings $y_1, \ldots, y_k \in \{0,1\}^m$ as follows. If $j \in T_{\mathcal{D}_C}$, choose $y_j \sim \mathcal{D}_C$. If $j \in T_{\mathcal{U}}$, choose $y_j \sim \mathcal{U}$. Send the sequence $y_1, \ldots, y_k$ to the prover.

2. Prover: Send sets $Y, H \subseteq [k]$ and strings $(w_j : j \in Y)$ to the prover. An honest prover sends

   (a) Y as the set of all j such that $y_j \in V$,

   (b) H as the set of all j such that y_j is α-heavy for $\mathcal{D}_C$, and

   (c) w_j such that $V(y_j; w_j)$ accepts for all $j \in Y$.

3. Verifier: Reject under any of the following circumstances:

[6] A *nondeterministic circuit* V computes a relation over $\{0,1\}^m \times \{0,1\}^l$, and we say $y \in \{0,1\}^m$ is accepted by V if there exists a $w \in \{0,1\}^l$ such that $V(y; w)$ accepts. Abusing notation, we also write V for the set of all y accepted by V.

(a) (Witness checks) $V(y_j; w_j)$ does not accept for some $j \in Y$.

(b) (Frequency of "yes" samples in $\mathcal{U}$) $\left||Y \cap T_{\mathcal{U}}|/|T_{\mathcal{U}}| - p_Y\right| \geq \epsilon b/32$.

(c) (Frequency of heavy samples in $\mathcal{D}_C$) $\left||H \cap T_{\mathcal{D}_C}|/|T_{\mathcal{D}_C}| - p_H\right| \geq \epsilon/16$.

(with Prover) Run the parallel lower bound Protocol with shared auxiliary input (error parameter) δ for the claim "$|C^{-1}(y_j)| \geq \alpha 2^{n-m}$ for all $j \in H \cap T_{\mathcal{D}_C}$."

Accept iff

$$\left||Y \cap \overline{H} \cap T_{\mathcal{D}_C}|/|T_{\mathcal{D}_C}| - p\right| < \epsilon/4. \tag{4}$$

Remark. The parameter b needs to satisfy two constraints. It has to ensure that the α-light queries are hidden well (as a choice of probability for membership in T) and also to guarantee that the fraction of "yes" samples in $T_{\mathcal{U}}$ is very close to p_U—not only within $O(\epsilon)$, but within $O(b\epsilon)$. This is necessary because $T_{\mathcal{D}_C}$ is smaller than $T_{\mathcal{U}}$ by a factor of b, so to obtain an $O(\epsilon)$ deviation bound for the fraction of light "yes" queries in $T_{\mathcal{D}_C}$, a stronger $O(b\epsilon)$ deviation bound must be assumed for the fraction of "yes" queries in $T_{\mathcal{U}}$.

3.2.2 Analysis of the protocol

Lemma 13. *For every integer α_0 and fractions ϵ, δ, with probability $1 - O(\delta/\epsilon)$ over α chosen uniformly from $\mathcal{A}_{\alpha_0,\delta}$, the hiding protocol (with input (C, V, p, ϵ) and auxiliary input $(\alpha, \delta, p_Y, p_H)$ satisfying the promise) is a protocol for $\Pi_{HEAVY,\alpha}$ with completeness $1 - O(\delta)$ and soundness $O(\delta)$.*

Proof. We denote by Y' and H' the set of actual "yes" samples and the set of actual heavy samples, respectively:

$$Y' = \{j : y_j \in V\} \text{ and } H' = \{j : \mathcal{D}_C(y_j) \geq \alpha 2^{-m}\}.$$

An honest prover always chooses $Y = Y'$ and $H = H'$.

The sampling bound implies that, for any fixed prover strategy, all of the following events hold with probability $1 - O(\delta)$ over the randomness of the verifier:

- The number of samples in $T_{\mathcal{D}_C}$ is large:

$$|T_{\mathcal{D}_C}| > bk/2 \geq 3 \cdot 32^3 \log(2/\delta)/\epsilon^3. \tag{5}$$

- The number of samples in $T_{\mathcal{U}}$ is large:

$$|T_{\mathcal{U}}| > (1 - b - \epsilon)k. \tag{6}$$

- About a p_Y fraction of samples in $T_{\mathcal{U}}$ are "yes" samples:

$$\left||Y' \cap T_{\mathcal{U}}|/|T_{\mathcal{U}}| - p_Y\right| < \epsilon b/32. \tag{7}$$

- About a p_H fraction of samples in $T_{\mathcal{D}_C}$ are heavy samples:

$$\left||H' \cap T_{\mathcal{D}_C}|/|T_{\mathcal{D}_C}| - p'_H\right| < \epsilon/32. \tag{8}$$

- About a $g_{\mathcal{D}_C,V,\alpha}$ fraction of samples in $T_{\mathcal{D}_C}$ are light "yes" samples:

$$\left||Y' \cap \overline{H}' \cap T_{\mathcal{D}_C}|/|T_{\mathcal{D}_C}| - g_{\mathcal{D}_C,V,\alpha}\right| < \epsilon/4. \tag{9}$$

Completeness. Completeness (for arbitrary α) follows from high probability estimates (7), (8) and (9), promise (3), and completeness of the parallel lower bound protocol for $\Pi_{LB,\delta}^{|H\cap T_{\mathcal{D}_C}|,1}$.

Soundness. Fix an $\alpha \sim \mathcal{A}_{\alpha_0,\delta}$ such that

$$\Pr_{y\sim\mathcal{D}_C}\left[\mathcal{D}_C(y) \in ((1-\delta)\alpha 2^{-m}, (1+\delta)\alpha 2^{-m})\right] \leq \epsilon/32.$$

By Claim 11 and Markov's inequality, this holds with probability $1 - O(\delta/\epsilon)$ for a random α in $\mathcal{A}_{\alpha_0,\delta}$. For such a choice of α, with probability $1 - O(\delta)$ over the randomness of the verifier, the number of samples in $T_{\mathcal{D}_C}$ that are both $(1-\delta)\alpha$-heavy and α-light is at most $\epsilon|T_{\mathcal{D}_C}|/16$. (This follows from the sampling bound and high probability estimate (5).)

Now fix a prover strategy for which the verifier accepts instance (C, V, p, ϵ) with probability $\omega(\delta)$. The analysis will be split into the following two parts:

- Show that the fraction of samples in $T_{\mathcal{D}_C}$ that were claimed both "yes" and "light" by the prover is within $O(\epsilon)$ of the fraction of truly light samples that were claimed "yes" by the prover. More generally, we show that for any set of samples $I \subseteq T_{\mathcal{D}_C}$, the fraction of samples in I that are claimed heavy by the prover is within $O(\epsilon)$ of the fraction of truly heavy samples in I. This will be shown in Claim 14.
- Show that if the fraction of false "no" claims for samples in $T_{\mathcal{U}}$ is small, then the fraction of false "no" claims for light samples in $T_{\mathcal{D}_C}$ is small. This will be shown in Claim 15, which formalizes the hiding property of the protocol, and contains the main idea of the soundness analysis.

Observe that step 3(a) of the hiding protocol ensures $Y \subseteq Y'$ whenever the verifier accepts. Let $Y^- = Y' - Y$.

Claim 14 (Heavy samples)**.** *With probability* $1 - O(\delta)$ *over the randomness of the verifier, if the verifier accepts then for every set* $I \subseteq T_{\mathcal{D}_C}$, $\left||I \cap \overline{H}| - |I \cap \overline{H}'|\right| \leq \epsilon|T_{\mathcal{D}_C}|/4$.

Claim 15 (Hiding Property)**.** *With probability* $1 - O(\delta)$ *over the randomness of the verifier,* $|Y^- \cap T_{\mathcal{U}}| > \frac{1}{2}|Y^- \cap \overline{H}'|$.

We give the proofs of both these claims at the end of the Section. Let us see first how these claims imply soundness. By a union bound, there exists an accepting transcript for the verifier where high probability estimates (5), (7), and (9) hold, and the properties in Claim 14 and Claim 15 both hold. Fix such an accepting transcript.

By the accepting condition (4) and high probability estimate (9),

$$\begin{aligned}
|p - g_{\mathcal{D}_C,V,\alpha}| &< \epsilon/4 + \epsilon/4 + \left|\frac{|Y \cap \overline{H} \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|} - \frac{|Y' \cap \overline{H}' \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|}\right| \qquad \text{by (4) and (9)} \\
&= \epsilon/4 + \epsilon/4 + \left|\frac{|Y \cap \overline{H} \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|} - \left(\frac{|Y \cap \overline{H}' \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|} + \frac{|Y^- \cap \overline{H}' \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|}\right)\right| \\
&\leq \epsilon/4 + \epsilon/4 + \left|\frac{|Y \cap \overline{H} \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|} - \frac{|Y \cap \overline{H}' \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|}\right| + \frac{|Y^- \cap \overline{H}' \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|} \\
&\leq \epsilon/4 + \epsilon/4 + \epsilon/4 + \frac{|Y^- \cap \overline{H}' \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|} \qquad \text{by Claim 14.}
\end{aligned}$$

We now apply the hiding property to bound the last term. First,

$$\frac{|Y^- \cap T_{\mathcal{U}}|}{|T_{\mathcal{U}}|} = \frac{|Y' \cap T_{\mathcal{U}}|}{|T_{\mathcal{U}}|} - \frac{|Y \cap T_{\mathcal{U}}|}{|T_{\mathcal{U}}|} = \left(\frac{|Y' \cap T_{\mathcal{U}}|}{|T_{\mathcal{U}}|} - p_Y\right) + \left(p_Y - \frac{|Y \cap T_{\mathcal{U}}|}{|T_{\mathcal{U}}|}\right) < \epsilon b/16,$$

by high probability estimate (7) and step 3(b) of the verifier. Now, using the hiding property and high probability estimate (5),

$$\frac{|Y^- \cap \overline{H}' \cap T_{\mathcal{D}_C}|}{|T_{\mathcal{D}_C}|} \leq \frac{|Y^- \cap \overline{H}'|}{|T_{\mathcal{D}_C}|} < 2 \cdot \frac{|Y^- \cap T_{\mathcal{U}}|}{|T_{\mathcal{D}_C}|} < 2 \cdot \frac{\epsilon bk/16}{bk/2} = \epsilon/4.$$

It follows that $|p - g_{\mathcal{D}_C,V,\alpha}| < \epsilon$, so (C, V, p, ϵ) is a yes instance of $\Pi_{HIDE,\alpha}$. □

Proof of Claim 14. For ease of notation, let $G = H \cap T_{\mathcal{D}_C}$ and $G' = H' \cap T_{\mathcal{D}_C}$. Denote by $\overline{G}$ and $\overline{G}'$ the complements of G and G' in $T_{\mathcal{D}_C}$, respectively.

Fix a transcript of the verifier for which high probability estimate (8) holds, none of the samples in $T_{\mathcal{D}_C} \cap H$ are $(1-\delta)\alpha$-light, and the number of samples whose weight is between $(1-\delta)\alpha$ and α in G is at most $\epsilon|T_{\mathcal{D}_C}|/16$. By soundness of the parallel lower bound protocol for $\Pi^{|G|,1}_{LB,\delta}$, Claim 11, and the sampling bound, all these events hold with probability $1 - O(\delta)$.

Suppose the verifier accepts. Since none of the samples in G are $(1-\delta)\alpha$-light and the number of samples whose weight is between $(1-\delta)\alpha$ and α in G is at most $\epsilon|T_{\mathcal{D}_C}|/16$, it follows that

$$|G - G'|/|T_{\mathcal{D}_C}| < \epsilon/16.$$

On the other hand, step 3(c) of the verifier, promise (3), and high probability estimate (8) give

$$\big||G| - |G'|\big|/|T_{\mathcal{D}_C}| \leq \big||G|/|T_{\mathcal{D}_C}| - p_H\big| + |p_H - p'_H| + \big|p'_H - |G'|/|T_{\mathcal{D}_C}|\big| < \epsilon/8.$$

The last two equations imply that the sets G and G' cannot differ on all but a few elements. Therefore, $\overline{G}$ and $\overline{G}'$ must also be very close, and so must be $I \cap \overline{G}$ and $I \cap \overline{G}'$.

The rest are calculations formalizing these claims. First, $|G' - G|$ must also be small because

$$\big||G'| - |G|\big| = \big||G' - G| - |G - G'|\big| \geq |G' - G| - |G - G'|$$

so that $|G' - G|/|T_{\mathcal{D}_C}| < 3\epsilon/16$. It follows that

$$\begin{aligned}\big||I \cap \overline{G}| - |I \cap \overline{G}'|\big| &\leq \big||I \cap \overline{G}| - |I \cap \overline{G} \cap \overline{G}'|\big| + |I \cap (\overline{G}' - \overline{G})| \\ &= |I \cap (\overline{G} - \overline{G}')| + |I \cap (\overline{G}' - \overline{G})| \\ &\leq |\overline{G} - \overline{G}'| + |\overline{G}' - \overline{G}| \\ &\leq \epsilon |T_{\mathcal{D}_C}|/4.\end{aligned}$$

□

Proof of Claim 15. We will, in fact, reach the stronger conclusion

$$|Y^- \cap \overline{H}' \cap T_{\mathcal{U}}| > \tfrac{1}{2}|Y^- \cap \overline{H}'|$$

For every sample $j \in Y^- \cap \overline{H}'$, that is, every light sample for which the prover made a false "no" claim, consider the event "$j \in T_{\mathcal{D}_C}$" from the point of view of the prover. For a fixed prover strategy, the first message of the verifier completely determines the set $Y^- \cap \overline{H}'$. First, we show that for any first message $\mathbf{y} = (y_1, \ldots, y_k)$ of the verifier, the probability of each event "$j \in T_{\mathcal{D}_C}$" for $j \in Y^- \cap \overline{H}'$ (over the randomness of the verifier's first message) is less than any constant:

$$\begin{aligned}\Pr[j \in T_{\mathcal{D}_C} \mid \mathbf{y}] &= \Pr[j \in T_{\mathcal{D}_C} \mid y_j] \qquad \text{by independence of samples} \\ &= \frac{\Pr[y_j \mid j \in T_{\mathcal{D}_C}] \Pr[j \in T_{\mathcal{D}_C}]}{\Pr[y_j]} \\ &\leq \frac{\Pr[y_j \mid j \in T_{\mathcal{D}_C}] \Pr[j \in T_{\mathcal{D}_C}]}{\Pr[y_j \mid j \in T_{\mathcal{U}}] \Pr[j \in T_{\mathcal{U}}]} \\ &\leq \frac{(\alpha 2^{-m}) \cdot b}{2^{-m} \cdot (1-b)} \qquad \text{by lightness} \\ &\leq 2\delta \qquad \text{by choice of } b.\end{aligned}$$

For fixed $\mathbf{y}$, the quantity $|Y^- \cap \overline{H}' \cap T_{\mathcal{D}_C}|$ is a sum of indicator random variables for the events "$j \in T_{\mathcal{D}_C}$", one for each $j \in Y^- \cap \overline{H}'$, so it follows that

$$\mathrm{E}\big[|Y^- \cap \overline{H}' \cap T_{\mathcal{D}_C}| \mid \mathbf{y}\big] = \textstyle\sum_{j \in Y^- \cap \overline{H}'} \Pr[j \in T_{\mathcal{D}_C} \mid \mathbf{y}] \leq 2\delta \cdot |Y^- \cap \overline{H}'|.$$

by Markov's inequality, we have that

$$\Pr\big[|Y^- \cap \overline{H}' \cap T_{\mathcal{D}_C}| > \tfrac{1}{2}|Y^- \cap \overline{H}'| \mid \mathbf{y}\big] < 4\delta.$$

therefore

$$\Pr\big[|Y^- \cap \overline{H}' \cap T_{\mathcal{U}}| \leq \tfrac{1}{2}|Y^- \cap \overline{H}'| \mid \mathbf{y}\big] < 4\delta.$$

The claim follows by taking expectation over $\mathbf{y}$. □

3.3 Simulating the reduction

In this section we describe the protocol that simulates a querier circuit Q (describing an instantiation of the worst-to-average reduction on a particular input) querying an average-case membership oracle

for some NP set V. Let us assume that all the queries made by Q are identically distributed, and denote by $\mathcal{D}_Q$ the distribution of a single query. The average-case membership oracle is for the set

$$S = \{y \in \{0,1\}^* : y \in V \text{ and } \mathcal{D}_Q(y) \leq \alpha 2^{-|y|}\}.$$

Recall that if Q describes an instantiation of a worst-to-average reduction from some language L to V, distinguishing between the cases when Q accepts most of its inputs and when Q rejects most of its inputs allows us to determine membership in L.

We assume that the protocol is given as advice the probability p_H that a random query of Q is α-heavy, and the probability p_S that a random query of Q is in S. In reality, the protocol is only given approximations of these values, but for the sake of simplicity we ignore the distinction in this discussion. Suppose that Q on input $r \in \{0,1\}^n$, generates k queries $y_1, \ldots, y_k$ and a circuit $C : \{0,1\}^k \to \{0,1\}$. Moreover, Q satisfies the promise that $C(S(y_1), \ldots, S(y_k))$ either accepts or rejects with probability $1 - \eta$ for some $\eta > 0$.

Let us first consider the case when the distribution $\mathcal{D}_Q$ is α-smooth, that is, all queries are α-light. In this case, $S = V$ and $p_H = 0$, and the protocol of Feigenbaum and Fortnow can be used directly, as the advice p_S gives the probability that a random query generated by Q is a "yes" query, which is the advice needed for the Feigenbaum-Fortnow protocol. Let us recall how this protocol works. The verifier generates $l = \lceil 24 \cdot (k/\eta)^3 \log(2k/\delta) \rceil$ (where $\delta > 0$ is an error parameter) random strings $r_1, \ldots, r_l$, sends these strings to the prover, and asks the prover to simulate the computation of $Q(r_i)$ with oracle S for every i. To certify that most of the simulations are correct, the prover provides, with every "yes" query made in the simulations, an NP witness (for V) for this query. With high probability, for all query indices $j \in [k]$, among the jth queries made by Q, $(p_S \pm \eta/k)l$ of these queries must be "yes" instances of V, so the verifier can ask to see at least $p_S lk - \eta l$ "yes" answers without affecting completeness. But no prover can now make more than ηl false "no" claims, so if the verifier outputs the outcome of a random simulation, it will be correct with probability $1 - \eta$.

For a general distribution $\mathcal{D}_Q$, the difficulty is that the prover can no longer certify membership in S as in the Feigenbaum-Fortnow protocol, as S is not an NP-set.[7] Instead of certifying membership in S directly, the verifier will first approximately determine which of its queries are α-heavy. To do so, the verifier uses the fact that heaviness is a certifiable property, thus limiting the cheating power of the prover: Statistically, the fraction of heavy queries is within $p_H \pm \eta/k$ with high probability, and the verifier asks the prover to give proofs of heaviness (using the lower bound protocol) for at least $p_H kl - \eta l$ of its queries. Since the prover's cheating power is one-sided (the prover is likely to be caught cheating if it claims that a light query is heavy), it can fool the verifier about heaviness on at most $2\eta l$ queries.

Once the verifier knows approximately which queries are α-heavy, it can ask the prover to reveal which queries are in V among the ones that are α-light: For each query that the verifier thinks is α-light, the prover is asked to determine membership in V, and provide a certificate in case of a "yes" answer. Statistically, the fraction of queries that are light and in V is within $p_S \pm \eta/k$ with

[7] In fact, if S were defined as the set of y such that $y \in V$ *or* y is α-heavy, then it would have been an AM set (almost, save the fact that heaviness is an approximate AM property). This provides an alternate way of proving the Main Theorem: Modify the hiding protocol to calculate the fraction of samples that are either "yes" or heavy, then simulate the reduction using the Feigenbaum-Fortnow protocol.

high probability, and the verifier asks to see "yes" certificates for at least a $p_S kl - \eta l$ queries that it thinks are α-light. If the set of queries that the verifier thinks are α-light coincided exactly with the set of truly α-light queries, the prover would not be able to provide more than $2\eta l$ false answers about membership in V among the α-light queries. In general these two sets will not coincide exactly. However, the number of elements on which they differ is at most $2\eta l$, so that the total number of truly α-light queries on which the prover can cheat about membership in V is still $O(\eta l)$.

It follows that for a random $i \in [l]$, the verifier can correctly simulate membership in S with probability $1 - O(\delta + \eta)$.

Regarding the choice of α, we encounter the same issue as in Sections 3.1 and 3.2: If too many queries have probability about α, the lower bound protocol provides no soundness guarantee. Again, we sidestep this issue by arguing completeness and soundness for a random α.

3.3.1 The protocol

We give a protocol for the following family of promise problems, which we denote by $\{\Pi_{SIM,\alpha}\}$.

Inputs. (Q, V, η), where V is a nondeterministic polynomial-time machine, $Q : \{0,1\}^n \to \{0,1\}^{\mathrm{poly}(n)}$ is a querier circuit producing k queries for the set

$$S = S_{\mathcal{D}_Q,V,\alpha} = \{y \in \{0,1\}^* : V \text{ accepts } y \text{ and } \mathcal{D}_Q(y) < \alpha 2^{-|y|}\},$$

and $\eta > 0$ is an error parameter.

Shared auxiliary input. $(\alpha, \delta, p_H, p_S)$, where α is a threshold parameter represented in unary, $0 < \delta < 1/3$ is an error parameter represented in unary, and p_S, p_H satisfy:

$$|p_H - p'_H| < \eta/2k \quad \text{and} \quad |p_S - p'_S| < \eta/2k, \tag{10}$$

where $p'_H = \Pr_{y\sim\mathcal{D}_Q}[\mathcal{D}_Q(y) \geq \alpha 2^{-|y|}]$ and $p'_S = \Pr_{y\sim\mathcal{D}_Q}[y \in S]$.

Yes instances. (Q, V, η) such that $\Pr_r[Q^S(r) \text{ accepts}] > 1 - \eta/2$.

No instances. (Q, V, η) such that $\Pr_r[Q^S(r) \text{ accepts}] < \eta/2$.

The simulation protocol. On input (Q, V), and auxiliary input $(\alpha, \delta, p_H, p_S)$:

1. Verifier: Set $l = \lceil 24 \cdot (k/\eta)^3 \log(2k/\delta) \rceil$. Choose l random strings $r_1, \ldots, r_l \in \{0,1\}^n$ and send them to the prover. Denote the jth query of $Q(r_i)$ by y_{ij}.
2. Prover: Send sets $Y, H \subseteq [l] \times [k]$ and strings $(w_{ij} : (i,j) \in Y)$ to the prover. An honest prover sends:
   (a) Y as the set of all (i,j) such that $y_{ij} \in V$,
   (b) H as the set of all (i,j) such that y_{ij} is α-heavy for $\mathcal{D}_Q$, and
   (c) w_{ij} such that $V(y_{ij}; w_{ij})$ accepts for all $(i,j) \in Y$.
3. Verifier: Reject under any of the following circumstances:

(a) (Witness checks) $V(y_{ij}; w_{ij})$ does not accept for some $(i,j) \in Y$.

(b) (Frequency of heavy samples) $\big||H|/kl - p_H\big| > \eta/k$.

(c) (Frequency of light "yes" samples) $\big||Y \cap \overline{H}|/kl - p_S\big| > \eta/k$.

(with Prover) Run the parallel lower bound protocol with shared auxiliary input (error parameter) δ for the claim:

$$|\{r : \text{The first query of } Q(r) \text{ is } y_{ij}\}| > \alpha 2^{n-|y_{ij}|} \text{ for all } (i,j) \in H.$$

Choose a random $i \in [l]$. Accept if the decider of $Q(r_i)$ accepts the input $(a_1, \ldots, a_k)$, where $a_j =$ "yes" if $(i,j) \in Y \cap \overline{H}$, and $a_j =$ "no" otherwise.

3.3.2 Analysis of the protocol

Lemma 16. *For every integer α_0, querier circuit Q that produces k queries, and fractions η, δ, with probability $1 - O(\delta k/\eta)$ over α chosen uniformly from $\mathcal{A}_{\alpha_0,\delta}$, the hiding protocol (with input (Q, V, η) and auxiliary input $(\alpha, \delta, p_H, p_S)$ satisfying the promise) is a protocol for $\Pi_{SIM,\alpha}$ with completeness $1 - O(\delta + \eta)$ and soundness $O(\delta + \eta)$.*

Proof. We denote by H' and Y' the set of actual heavy samples, and the set of actual "yes" samples, respectively:

$$H' = \{(i,j) : \mathcal{D}_Q(y_{ij}) \geq \alpha 2^{-|y_{ij}|}\} \text{ and } Y' = \{(i,j) : y_{ij} \in V\}.$$

The honest prover always chooses $H = H'$ and $Y = Y'$.

First, we observe the following high probability estimates over the randomness of the verifier (for any prover strategy), which follow directly from the sampling bound:

$$(Q, V, \eta) \text{ is a yes instance} \implies \Pr\big[|\{i : Q^S(r_i) \text{ accepts}\}| > (1-\eta)l\big] = 1 - O(\delta) \tag{11}$$

$$(Q, V, \eta) \text{ is a no instance} \implies \Pr\big[|\{i : Q^S(r_i) \text{ accepts}\}| < \eta l\big] = 1 - O(\delta). \tag{12}$$

Let T denote an arbitrary fixed (that is, independent of both the verifier's randomness and the prover's strategy) subset of $\{0,1\}^*$. The key observation of Feigenbaum and Fortnow is that for any T, with probability $1 - O(\delta)$ over the randomness of the verifier, the fraction of queries y_{ij} that fall inside T is $|T|/kl \pm \eta/k$, even though there are dependencies among the queries. To see this, divide the queries into k sets, where the jth set consists of queries $q_{1j}, \ldots, q_{lj}$. Within each set, the queries are independent, so by the choice of $l = \lceil 24 \cdot (k/\eta)^3 \log(2k/\delta) \rceil$ and by the sampling bound, the fraction of queries in the jth set that fall inside T is $|T|/kl \pm \eta/k$ with probability $1 - O(\delta/k)$. By a union bound over j, it follows that with probability $1 - O(\delta)$ the total fraction of queries y_{ij} that fall inside T is within η/k of $|T|/kl$.

Specifically, in the case when T is the set of α-heavy queries, we obtain that with probability $1 - O(\delta)$ over the randomness of the verifier, it holds that

$$\big||H'|/kl - p_H\big| < \eta/2k. \tag{13}$$

When T is the set of α-light queries that are in V, again with probability $1 - O(\delta)$ over the randomness of the verifier, it holds that

$$\left||Y' \cap \overline{H}'|/kl - p_S\right| < \eta/2k. \tag{14}$$

Completeness. Completeness (for arbitrary α) follows from high probability estimates (11), (13), (14), promise (10), and completeness of the parallel lower bound protocol for the promise problem $\Pi_{LB,\delta}^{|H|,1}$.

Soundness. Fix an $\alpha \sim \mathcal{A}_{\alpha_0,\delta}$ such that

$$\Pr_{y\sim\mathcal{D}_Q}\left[\mathcal{D}_Q(y) \in ((1-\delta)\alpha 2^{-|y|}, (1+\delta)\alpha 2^{-|y|})\right] \leq \eta/2k.$$

By Claim 11 and Markov's inequality, this holds with probability $1 - 2\delta k/\eta$ for a random α in $\mathcal{A}_{\alpha_0,\delta}$. For such a choice of α, it follows from the sampling bound that with probability $1 - O(\delta)$ over the randomness of the verifier, the number of queries that are both $(1-\delta)\alpha$-heavy and α-light is at most ηl.

Fix a prover strategy for which the verifier accepts instance (Q, V, η) with probability $\omega(\delta) + 11\eta$. We will show that at least a 11η fraction of the transcripts are accepting, satisfy high probability estimate (12), and have the property that the prover is honest on all but at most $10\eta l$ answers provided in step 2 of the protocol: Namely, they satisfy the condition

$$|(Y \cap \overline{H}) \triangle (Y' \cap \overline{H}')| < 10\eta l. \tag{15}$$

Now consider all prefixes of transcripts consisting of the prover-verifier interaction before the verifier's choice of index i in step 3. There must exist at least one such prefix that satisfies estimate (12) and condition (15), and for which at least an 11η fraction of choices for i yield accepting transcripts. For this prefix, condition (15) implies that for at least an $1 - 10\eta$ fraction of indices i, the verifier correctly simulates the computation $Q^S(r_i)$ using the claims received from the prover in step 2. Therefore, for at least an η fraction of indices i, the transcript resulting from this choice of i is accepting. By condition (12), it follows that (Q, V, η) must be a "yes" instance.

We now show that at least an 11η fraction of transcripts are accepting, satisfy high probability estimate (12), and satisfy condition (15). For an accepting transcript, step 3(a) of the verifier guarantees that $Y \subseteq Y'$. Let $Y^- = Y' - Y$. If $\overline{H}$ were equal to $\overline{H}'$ with high probability, we would have $(Y \cap \overline{H}) \triangle (Y' \cap \overline{H}') = Y^- \cap \overline{H}'$, so the claim would follow from the verifier's step 3(c) and estimates (12) and (14). The only complication is that $\overline{H}$ and $\overline{H}'$ are not equal (and the difference between them can be two-sided), but the set difference is small: $|H \triangle H'| \leq 4\eta l$ for a $1 - O(\delta)$ fraction of transcripts because with probability $1 - O(\delta)$ each,

- By soundness of the parallel lower bound protocol for $\Pi_{LB,\delta}^{|H|,1}$, none of the samples in H are $(1-\delta)\alpha$-light. Also, the number of samples whose weight is between $(1-\delta)\alpha$ and α is at most ηl, so it follows that $|H - H'| \leq \eta l$.
- Step 3(b) of the verifier, promise (10), and high probability estimate (13) give

$$\left||H| - |H'|\right| \leq \left||H| - p_H kl\right| + |p_H - p'_H|kl + \left|p'_H kl - |H'|\right| < 2\eta l.$$

Then $|H' - H| \leq 3\eta l$ because

$$\big||H'| - |H|\big| = \big||H' - H| - |H - H'|\big| \geq |H' - H| - |H - H'|.$$

It follows that

$$|H \mathbin{\triangle} H'| = |H - H'| + |H' - H| \leq 4\eta l. \tag{16}$$

By a union bound, at least an 11η fraction of transcripts are accepting, satisfy high probability estimates (12), (14), and condition (16). For such transcripts, we have

$$\begin{aligned}
\big|(Y \cap \overline{H}) \mathbin{\triangle} (Y' \cap \overline{H}')\big| &= \big|(Y \cap \overline{H}) \mathbin{\triangle} ((Y \cap \overline{H}') \mathbin{\triangle} (Y^- \cap \overline{H}'))\big| \\
&= \big|(Y \cap (\overline{H} \mathbin{\triangle} \overline{H}')) \mathbin{\triangle} (Y^- \cap \overline{H}')\big| \\
&\leq \big|\overline{H} \mathbin{\triangle} \overline{H}'\big| + |Y^- \cap \overline{H}'| \\
&\leq 4\eta l + |Y^- \cap \overline{H}'|.
\end{aligned}$$

The last line follows from the fact that the symmetric difference stays the same if the sets are complemented. Therefore,

$$\begin{aligned}
|Y^- \cap \overline{H}'| &\leq \big||Y^- \cap \overline{H}'| + (|Y \cap \overline{H}'| - |Y \cap \overline{H}|)\big| + \big||Y \cap \overline{H}'| - |Y \cap \overline{H}|\big| \\
&\leq \big||Y' \cap \overline{H}'| - |Y \cap \overline{H}|\big| + \big|(Y \cap \overline{H}') \mathbin{\triangle} (Y \cap \overline{H})\big| \\
&= \big||Y' \cap \overline{H}'| - |Y \cap \overline{H}|\big| + \big|Y \cap (\overline{H}' \mathbin{\triangle} \overline{H})\big| \\
&\leq \big(\big||Y' \cap \overline{H}'| - p'_S kl\big| + |p'_S - p_S|kl + \big|p_S kl - |Y \cap \overline{H}|\big|\big) + |H \mathbin{\triangle} H'| \\
&< (\eta l/2 + \eta l/2 + \eta l) + 4\eta l. \quad \text{by (14), (10), verifier step 3(c), and (16)}
\end{aligned}$$

so that $|(Y \cap \overline{H}) \mathbin{\triangle} (Y' \cap \overline{H}')| < 4\eta l + 6\eta l = 10\eta l$. □

4 Main theorem and proof

Theorem 17 (Main Theorem)**.** *For any two languages L and L' such that $L' \in$ NP and every constant c, if there is a n^{-c} non-adaptive worst-to-average reduction from L to L', then $L \in$ NP/poly $\cap$ coNP/poly.*

In particular, if L were hard for NP, then coNP $\subseteq$ NP/poly, therefore $\Sigma_3 = \Pi_3$.

Proof. Fix an arbitrarily small constant $\eta > 0$. We will assume that there exist polynomials $k(n)$ and $m(n)$ such that for every n and every input x of length n, the reduction makes exactly $k(n)/m(n)$ queries of every length between 1 and $m(n)$, so that the total number of queries made by the reduction is $k(n)$. We will also assume that the queries made by the reduction are identically distributed, and that (when provided access to an average-case oracle) the reduction either accepts or rejects with probability at least $1 - \eta/2$. In Section 2.2 we explained why all these assumptions can be made without loss of generality.

Observe that if there is a n^{-c} worst-to-average reduction from L to L', then there is also a worst-to-average reduction from $\overline{L}$ to L', so it suffices to prove that $L \in \mathrm{NP/poly}$. We describe an AM protocol for L with advice (in which completeness and soundness only hold when the advice is correct) and private coins with completeness $1 - O(\eta)$ and soundness $O(\eta)$. By the remarks in Section 2.3, the existence of such a protocol for L shows that $L \in \mathrm{NP/poly}$.

Let R_n denote the circuit computing the worst-to-average reduction from L to L' on inputs of length n. Let V be a non-deterministic machine for L'. Set $\alpha_0 = n^{-c}$.

Input. A string x of length n.

Advice. For every $1 \leq i \leq m(n)$, the probability $p_{Y,i} = \Pr_{y \sim \{0,1\}^i}[y \in L']$, and the circuit R_n.

Let Q be the circuit obtained by hard-wiring the input x to R_n. Thus Q takes as input a random string r and produces as output $k(n)$ queries and a decider circuit. For every $1 \leq i \leq m(n)$, let C_i denote the circuit that generates a random query of Q of length i: The circuit C_i simulates the circuit Q, then uses additional randomness to select uniformly one of the $m(n)$ outputs of Q of length i. Let $\mathcal{D}_{C_i}$ be the distribution of a sample of C_i, and $\mathcal{D}_Q$ be the distribution of the first query of Q. Finally, let V_i be the nondeterministic circuit describing the computation of $M_{L'}$ on an input of length i.

The protocol.

1. Verifier: Set the error parameters δ, ϵ_1 and ϵ_2 so that $\delta = \min(\epsilon_1/m(n), 1/3)$, $\epsilon_1 = \epsilon_2/32$, and $\epsilon_2 = \eta/2k(n)$. Choose a random α from the distribution $\mathcal{A}_{\alpha_0,\delta}$ (see (1)). Send α to the prover.

2. Prover: For every $1 \leq i \leq m(n)$, send two probabilities $h_{\mathcal{D}_{C_i},\alpha}$ and $g_{\mathcal{D}_{C_i},V,\alpha}$ to the verifier. An honest prover sends

$$h_{\mathcal{D}_{C_i},\alpha} = \Pr_{y \sim \mathcal{D}_{C_i}}[\mathcal{D}_{C_i}(y) \geq \alpha 2^{-i}] \text{ and } g_{\mathcal{D}_{C_i},V,\alpha} = \Pr_{y \sim \mathcal{D}_{C_i}}[\mathcal{D}_{C_i}(y) < \alpha 2^{-i} \text{ and } y \in V].$$

3. Verifier and Prover: For every $1 \leq i \leq m(n)$, run (in parallel) the heavy samples protocol (see Section 3.1) on input $(C_i, h_{\mathcal{D}_{C_i},\alpha}, \epsilon_1)$ and shared auxiliary input (α, δ).

   For every $1 \leq i \leq m(n)$, run (in parallel) the hiding protocol (see Section 3.2) on input $(C_i, V_i, g_{\mathcal{D}_{C_i},V_i,\alpha}, \epsilon_2)$ and shared auxiliary input $(\alpha, \delta, p_{Y,i}, h_{\mathcal{D}_{C_i},\alpha})$.

4. Verifier: Let

$$p_H = \sum_{i=1}^{m(n)} \frac{1}{m(n)} \cdot h_{\mathcal{D}_{C_i},\alpha} \text{ and } p_S = \sum_{i=1}^{m(n)} \frac{1}{m(n)} \cdot g_{\mathcal{D}_{C_i},V,\alpha}.$$

5. Verifier and Prover: Run the simulation protocol (see Section 3.3) on input $(Q, M_{L'}, \eta)$ and shared auxiliary input $(\alpha, \delta, p_H, p_S)$.

Observe that when the prover is honest, the probability p_H is exactly the probability that a random query of Q is α-heavy regardless of its length. Conversely, if p_H deviates from the probability that a random query of Q is α-heavy by more than ϵ, it must be that at least one of the claims $h_{\mathcal{D}_{C_i},\alpha}$

deviates from the value $\Pr_{y\sim\mathcal{D}_{C_i}}[\mathcal{D}_{C_i}(y) \geq \alpha 2^{-i}]$. Similar considerations apply to the probability p_Y.

Analysis of the protocol. The protocol intends to simulate a run of the reduction R when given oracle access to the set L^*_α for some α, where

$$L^*_\alpha = \{y \in \{0,1\}^* : y \in L' \text{ and } \mathcal{D}_Q(y) < \alpha 2^{-|y|}\}$$

Observe that for every $\alpha \in \mathcal{A}_{\alpha_0,\delta}$, the languages L' and L^*_α are $1/\alpha_0$-close: The distance between L' and L^*_α equals the measure of the set of α-heavy samples for $\mathcal{D}_Q$ under the uniform distribution. Since the number of α-heavy samples of $\mathcal{D}_Q$ of length i cannot exceed $\alpha^{-1} \cdot 2^{-i}$, the two sets are $1/\alpha \leq 1/\alpha_0$-close under the uniform distribution.

Observe that our choice of parameters guarantees that for a random choice of $\alpha \sim \mathcal{A}_{\alpha_0,\delta}$, with probability $1-O(\eta)$ over the choice of α, all runs of the heavy samples protocol, the hiding protocol, and the query simulation protocol satisfy the completeness and soundness conditions guaranteed by Lemmas 12, 13, and 16. For such a choice of α, completeness and soundness of the protocol follow by inspection. The completeness error is $O(\eta)$, which we obtain by adding the completeness errors of all the component protocols. The soundness error is also $O(\eta)$, which we obtain by observing that parallel composition does not increase the soundness error (see Section 2.3), and adding the soundness errors from steps 3 and 5 of the protocol. □

Remarks.

- If the worst-to-average reduction were uniform, the proof of Theorem 17 actually gives the stronger conclusion $L \in \mathrm{AM}^{\log}$. This requires small modification to the protocol: Instead of requiring that the protocol be given as advice the values $p_{Y,i}$ for all i between 1 and $m(n)$, we only ask that the advice consist of the average $p_Y = \sum_{i=1}^{m(n)} p_{Y,i}/m(n)$, which can be represented using $O(\log n)$ bits. As a preliminary step of the modified protocol, the prover sends claims for the actual values $p_{Y,i}$, and the verifier checks that p_Y is the average of these values. To check that these claims are correct (within an arbitrarily small additive term ϵ), for each i, the verifier generates $\omega(\log(m(n))/\epsilon^3)$ uniformly random samples of length i and asks the prover to provide certificates for membership in V for at least a $p_{Y,i} - \epsilon$ fraction of them. An honest prover can provide sufficiently many certificates with high probability, and the power of the cheating prover is one-sided: Such a prover cannot understate any $p_{Y,i}$ by more than 2ϵ, so to preserve the average p_Y it cannot overstate any $p_{Y,i}$ by more than $2\epsilon m(n)$. Choosing ϵ small enough provides the verifier with sufficiently good approximations of the values $p_{Y,i}$.

- The condition $L' \in \mathrm{NP}$ can be weakened to $L' \in \mathrm{NP/poly}$, as the advice for the verifier of L' can be incorporated as advice to the protocol.

- The conclusion of the theorem holds even under Levin's notion of hardness for efficient on average algorithms. This notion makes the additional requirement that L^* be an "errorless" approximation of L in the proof of Theorem 17: That is, L^* now takes values in $\{0, 1, \text{"fail"}\}$ and it is required that if $L^*(x) \neq L'(x)$, then $L^*(x) =$ "fail". Accommodating this change

requires merely a slight modification of the simulation protocol: Instead of simulating answers to heavy queries by "no", the modified protocol simulates answers to heavy queries by "fail".

5 Search problems and samplable distributions

In this section, we generalize our results to reductions from worst-case hard languages to average-case search problems (instead of languages) whose average-case complexity is measured with respect to arbitrary samplable distributions (instead of the uniform distribution).

Observe that if a decision problem L in NP is hard on average with respect to some distribution $\mathcal{D}$, then the search version of L is also hard with respect to $\mathcal{D}$. However, the converse is not evident. Thus even though Theorem 17 shows that non-adaptive worst-case to average-case reductions from an NP-hard problem to decision problems in NP are unlikely to exist, it is conceivable that reductions to search problems in NP are possible. In this section we rule out this possibility, showing that reductions to arbitrary search problems in distributional NP are no more powerful than reductions to decision problems in NP with respect to the uniform distribution.

The idea of the proof is to show that if there exists a worst-to-average reduction from some language L to a search problem in distributional NP, then this reduction can be composed with known reductions from average-case complexity of Impagliazzo and Levin [IL90] and Ben-David et al. [BCGL89] to obtain a worst-to-average reduction from L to some language L' with respect to the uniform distribution, thus reducing this to the special case studied in Theorem 17. A crucial property for our purpose of the reductions in [BCGL89, IL90], which is implicit in those works, is that both reductions are non-adaptive.

We begin by defining the type of reduction under consideration, as well as the types of reductions implicit in [BCGL89, IL90] that will be used in the proof of the main theorem of this section.

5.1 Average-case reductions for search problems

The notion of a "worst-case to average-case reduction" can be generalized in several ways. Such generalizations are needed in order to extend our impossibility result for worst-case to average-case reductions to the case when the average-case problem is a distributional search problem. To obtain this result, we will need to compose worst-to-average reductions with *average-case reductions*.

We begin with the notion of a heuristic NP-search algorithm that not only works well on average, but also provides witnesses for "yes" instances.

Let V be an NP-relation. We denote by L_V the NP-language corresponding to V, i.e., $L_V(x) = 1$ iff there exists a w such that $V(x; w) = 1$. A family of random functions $F_n : \{0,1\}^n \to \{0,1\}^m$ is a *δ-approximate witness oracle* for V with respect to the ensemble of distributions $\mathcal{D}$ if for all n,[8]

$$\Pr_{x \sim \mathcal{D}, F}[V(x; F_{|x|}(x)) = L_V(x)] > 1 - \delta.$$

[8]Technically, a witness oracle is an ensemble of distributions over function families $\{F_n\}$, but to simplify notation we will identify samples from this ensemble with the ensemble itself.

We will omit the subscript of F when it is implicitly determined by the input length. Note that the definition implies the existence of a set S of measure $\mathcal{D}(S) = 1 - 3\delta$ such that all $x \in S$,

$$\Pr_F[V(x; F_{|x|}(x)) = L_V(x)] > 2/3.$$

Intuitively, S is the set of inputs where the oracle has a good chance of producing a witness for the input, when such a witness exists. As usual, the constant $2/3$ is arbitrary, since if one has access to F, it can be queried k times independently in parallel to obtain a good witness with probability $1 - 1/3^k$.

Just as languages in NP represent decision problems, witness oracles represent search problems. For example, inverting a one-way function $f : \{0,1\}^n \to \{0,1\}^n$ on a $1 - \delta$ fraction of inputs amounts to finding an algorithm $A : \{0,1\}^n \to \{0,1\}^n$ that is δ-approximate for the relation $V(y; x) \iff y = f(x)$ with respect to the distribution $f(\mathcal{U}_n)$.

Using witness oracles, we can formalize the notion of nonadaptive reductions between search problems, as well as reductions from search to decision problems. Let us focus on the case of a reduction between two search problems $(V, \mathcal{D})$ and $(V', \mathcal{D}')$. As in the case of languages, we want the property that the reduction transforms any heuristic algorithm for $(V', \mathcal{D}')$ into a heuristic algorithm for $(V, \mathcal{D})$. Moreover, if $x \in L_V$, we want that the reduction, on most inputs $x \sim \mathcal{D}$, recovers a witness for x based on the answers provided by the witness oracle for V'.

Definition 18 (Reduction between search problems)**.** *Let V, V' be* NP *relations and $\mathcal{D}, \mathcal{D}'$ be polynomial-time samplable distribution ensembles. A* δ-to-δ' search-to-search reduction *for search problems from $(V, \mathcal{D})$ to $(V', \mathcal{D}')$ is a family of polynomial-size circuits $R = \{R_n\}$ such that on input $x \in \{0,1\}^n$, randomness r, $R_n(x; r)$ outputs strings $y_1, \ldots, y_k$ and a circuit C such that for any witness oracle F^* that is δ'-approximate for V' with respect to $\mathcal{D}'$, it holds that*

$$V(x, C(F^*(y_1), \ldots, F^*(y_k))) = L_V(x)$$

with probability $1 - \delta$ over the choice of $x \sim \mathcal{D}$, F^*, and the randomness used by the reduction.

This definition subsumes the case of a worst-to-average reduction: A δ' worst-to-average reduction is simply a 0-to-δ' average-to-average reduction. The other type of reduction used in the analysis—the search-to-decision reduction—is formalized in a similar way:

Definition 19 (Search-to-decision reduction)**.** *Let V be an* NP *relation, L' be an* NP *language, and $\mathcal{D}, \mathcal{D}'$ be polynomial-time samplable distribution ensembles. A* δ-to-δ' search-to-decision reduction *for search problems from $(V, \mathcal{D})$ to $(L', \mathcal{D}')$ is a family of polynomial-size circuits $R = \{R_n\}$ such that on input $x \in \{0,1\}^n$, randomness r, $R_n(x; r)$ outputs strings $y_1, \ldots, y_k$ and a circuit C such that for any L^* that is δ'-close to L' with respect to $\mathcal{D}'$, it holds that*

$$V(x, C(L^*(y_1), \ldots, L^*(y_k))) = L_V(x)$$

with probability $1 - \delta$ over the choice of $x \sim \mathcal{D}$ and the randomness used by the reduction.

5.2 Worst-case to average-case reductions to distributional search problems

We now state the main result of this section.

Theorem 20. *Let L be a language, V' be an* NP*-relation, $\mathcal{D}'$ be an arbitrary polynomial-time samplable ensemble of distributions, and c be a constant. If there is a non-adaptive n^{-c} worst-to-average reduction from L to $(V', \mathcal{D}')$, then $L \in \mathrm{NP/poly} \cap \mathrm{coNP/poly}$.*

To understand the meaning of Theorem 20, consider a polynomial time computable function f and a samplable ensemble of inputs $\mathcal{D} = \{\mathcal{D}_n\}$, and suppose that we want to prove that f is a one-way function with respect to the distribution $\mathcal{D}$. (That is, for a random $x \sim \mathcal{D}_n$, it is hard on average to find a preimage of $f(x)$.) We may set our aim low, and only try to prove that f is just infinitely often a weak one-way function. This means that there is a polynomial p such that, for every polynomial time inverter A, the computation $A(f(x))$ fails with probability at least $1/p(n)$ to output a preimage of $f(x)$, where the probability is over the coin tosses of A and the sampling of x from $\mathcal{D}_n$, and the statement is true for infinitely many n. We could try to provide evidence for the hardness of f by giving a reduction showing that an adversary that inverts A with probability better than $1 - 1/p(n)$ on all input lengths would imply a BPP algorithm for a presumably hard language L. Theorem 20 implies that if such a reduction is non-adaptive, then $L \in \mathrm{coNP/poly}$, and if L were NP-hard we would have a collapse of the polynomial hierarchy. Specifically, in order to apply Theorem 20 to our setting, consider the NP relation V' made of pairs $(f(x), x)$, and define the distribution $\mathcal{D}'$ as the ensemble $\{f(x)\}$ when x is sampled from $\mathcal{D}$. Then solving the search problem of V' on a random instance of $\mathcal{D}'$ is the same as inverting $f(x)$ on a random x taken from $\mathcal{D}$. A non-adaptive reduction of a decision problem L to such a problem implies that $L \in \mathrm{coNP/poly}$.

Theorem 20 is an immediate consequence of Theorem 17 and the following two lemmas:

Lemma 21. *For every $\delta = \delta(n)$ and* NP*-relation $V \subseteq \cup_n \{0,1\}^n \times \{0,1\}^{m(n)}$ there exists an* NP*-language L' for which there is a $O(\delta(m(n))^2)$-to-δ average-to-average reduction from $(V, \mathcal{U})$ to $(L', \mathcal{U})$.*

Lemma 22. *For every $\delta = \delta(n)$,* NP*-relation V and polynomial-time samplable ensemble of distributions $\mathcal{D}$ there exists a constant c and an* NP*-relation V' for which there is a $O(\delta n^c)$-to-δ average-to-average reduction from $(V, \mathcal{D})$ to $(V', \mathcal{U})$.*

Analogues of Lemmas 21 and 22 are known in the context of the distributional hardness of NP-problems. A variant of Lemma 21 appears Ben-David et al. [BCGL89], while a variant of Lemma 22 was proved by Impagliazzo and Levin [IL90]. Our proofs are in essence a recasting of these arguments in the formalism of nonadaptive average-to-average reductions. These proofs are presented in Appendix A.3 and Appendix A.4, respectively.

Acknowledgments

We thank Madhu Sudan for suggesting the relevance of [IL90], Oded Goldreich for stressing the relevance of our result to the question of basing cryptography on NP-hardness, and Amit Sahai for helpful discussions. We also thank Oded Goldreich and an anonymous reviewer for many useful comments on the presentation. The hiding protocol was suggested by Manikandan Narayanan.

References

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 284–293, 1997. 6

[AFK89] Martn Abadi, Joan Feigenbaum, and Joe Kilian. On hiding information from an oracle. *Journal of Computer and System Sciences*, 39(1):21–50, 1989. 2

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42:327–345, 1991. 16

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 99–108, 1996. 4, 6

[BCGL89] Shai Ben-David, Benny Chor, Oded Goldreich, and Michael Luby. On the theory of average-case complexity. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, pages 204–216, 1989. 5, 6, 10, 37, 39, 44

[BF90] Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *Proceedings of the 7th Symposium on Theoretical Aspects of Computer Science*, pages 37–48, 1990. 2

[BFKR97] Donald Beaver, Joan Feigenbaum, Joe Kilian, and Phillip Rogaway. Locally random reductions: Improvements and applications. *Journal of Cryptology*, 10(1):17–36, 1997. 2

[BK95] Manuel Blum and Sampath Kannan. Designing programs that check their work. *Journal of the ACM*, 41(1):269–291, 1995. Also in STOC'89. 5

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993. 5

[Blu88] Manuel Blum. Designing programs to check their work. Technical Report 88-09, ICSI, 1988. 5

[Bra79] Gilles Brassard. Relativized cryptography. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, pages 383–391, 1979. 3

[BT05] Andrej Bogdanov and Luca Trevisan. Average-case complexity: A survey. In preparation, 2005. 5

[CGKS98] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–982, 1998. 6

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, I22(6):644–654, 1976. 2, 3

[ESY84] Shimon Even, Alan L. Selman, and Yacob Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61:159–173, 1984. 14

[EY80] Shimon Even and Yacob Yacobi. Cryptography and NP-completeness. In *Proceedings of the 7th ICALP*, volume 85 of *LNCS*, pages 195–207. Springer-Verlag, 1980. 3

[FF93] Joan Feigenbaum and Lance Fortnow. On the random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22:994–1005, 1993. 2, 4, 6, 7, 8, 13

[FKN90] Joan Feigenbaum, Sampath Kannan, and Noam Nisan. Lower bounds on random-self-reducibility. In *Proceedings of the 5th IEEE Conference on Structure in Complexity Theory*, pages 100–109, 1990. 2

[For87] Lance Fortnow. The complexity of perfect zero-knowledge. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pages 204–209, 1987. 9, 16

[GG98] Oded Goldreich and Shafi Goldwasser. On the possibility of basing cryptography on the assumption that $P \neq NP$. Unpublished manuscript, 1998. 3, 6

[GKST02] Oded Goldreich, Howard Karloff, Leonard Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *Proceedings of the 17th IEEE Conference on Computational Complexity*, pages 175–183, 2002. 7

[Gol97] Oded Goldreich. Notes on Levin's theory of average-case complexity. Technical Report TR97–058, Electronic Colloquium on Computational Complexity (ECCC), 1997. 5

[Gol99] Oded Goldreich. *Modern cryptography, probabilistic proofs and pseudorandomness*, volume 17 of *Algorithms and combinatorics series*. Springer, 1999. 15

[Gol05] Oded Goldreich. On promise problems. Technical Report TR05–018, Electronic Colloquium on Computational Complexity (ECCC), 2005. 14

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the 18th ACM Symposium on Theory of Computing*, pages 59–68, 1986. 9, 13, 16

[GVW01] Oded Goldreich, Salil Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. In *Proceedings of the 28th ICALP*, volume 2076 of *LNCS*, pages 334–345. Springer-Verlag, 2001. 17

[HVV04] Alexander Healy, Salil Vadhan, and Emanuele Viola. Using nondeterminism to amplify hardness. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 192–201, 2004. 11

[IL90] Russell Impagliazzo and Leonid Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 812–821, 1990. 5, 6, 9, 37, 39

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the 10th IEEE Conference on Structure in Complexity Theory*, pages 134–147, 1995. 2, 5, 11

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error correcting codes. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 80–86, 2000. 7

[Lem79] Abraham Lempel. Cryptology in transition. *ACM computing surveys*, 11(4):285–303, 1979. 3

[Lev86] Leonid Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986. 4, 10, 11

[Lip89] Richard Lipton. New directions in testing. In *Proceedings of DIMACS Workshop on Distributed Computing and Cryptography*, 1989. 5

[Mic04] Daniele Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM Journal on Computing*, 34(1):118–169, 2004. 4, 6

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995. 42

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measure. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pages 372–381, 2004. 4, 6

[O'D02] Ryan O'Donnell. Hardness amplification within NP. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pages 751–760, 2002. 11

[Reg03] Oded Regev. New lattice based cryptographic constructions. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 407–416, 2003. 6

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001. 3

[Tre05] Luca Trevisan. On uniform amplification of hardness in NP. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, 2005. 6, 11

[VV86] Leslie Valiant and Vijay Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(1):85–93, 1986. 44

[Yap83] Chee K. Yap. Some consequences of nonuniform conditions on uniform classes. *Theoretical Computer Science*, 26:287–300, 1983. 13

A Appendix

A.1 Additive bounds for random sampling

Proof of Lemma 10. Let $N = |S|$ and $p = \mathcal{D}(T)$. We use the following form of the Chernoff bound (see [MR95, Section 4.1]):

$$\Pr[|T \cap S| < (1-\xi)Np] < \exp(-\xi^2 Np/2), \text{ for } \xi < 1$$

and

$$\Pr[|T \cap S| > (1+\xi)Np] < \begin{cases} (4/e)^{-(1+\xi)Np}, & \text{for } \xi > 1, \\ \exp(-\xi^2 Np/3), & \text{for } \xi \leq 1. \end{cases}$$

If $p < \epsilon$, the lower bound holds trivially, and for the upper bound we set $\xi = \epsilon/p > 1$ to obtain

$$\Pr[|T \cap S| > (p+\epsilon)N] < (4/e)^{-(p+\epsilon)N} < \eta.$$

If $p \geq \epsilon$, we set $\xi = \epsilon$ to obtain:

$$\Pr[|T \cap S| \notin (1 \pm \epsilon)Np] < 2\exp(-\epsilon^2 Np/3) \leq 2\exp(-\epsilon^3 N/3) < \eta. \quad \square$$

A.2 Proof sketches for the lower and upper bound protocols

Proof Sketch for Lemma 6. Let $S = C^{-1}(1)$. For $r \in S$, let I_r be an indicator for the event $h(r) = 0$, and let $R = h^{-1}(0)$, so that $|R| = \sum_{r \in S} I_r$. By the pairwise independence of h, we have $\mathrm{E}[|R|] = |S|k/s$, $\mathrm{Var}[|R|] \leq |S|k/s$, so by Chebyshev's inequality

$$\Pr[|R| \notin (1 \pm \epsilon/3)|S|k/s] \leq \frac{9}{\epsilon^2} \cdot \frac{s}{|S|k}$$

The bounds now follow by direct calculation. $\square$

Proof Sketch for Lemma 8. Let $S = C^{-1}(1)$. Fix r and let $S' = S - \{r\}$, $k' = |S'|/|\Gamma|$, $R = h^{-1}(h(r))$, $R' = R - \{r\}$. For every $r' \in S'$, let $I_{r'}$ be an indicator for the event $r' \in R'$, so that $|R'| = \sum_{r' \in S'} I_{r'}$. By the 3-wise independence of h, the $I_{r'}$ are pairwise independent conditioned on $h(r)$, so that $\mathrm{E}[|R'|] = k'$, $\mathrm{Var}[|R'|] = k'(1 - 1/|\Gamma|) < k'$ and by Chebyshev's inequality, for every $\xi > 0$:

$$\Pr[|R'| \notin (1 \pm \xi)k'] < 1/\xi^2 k'.$$

Suppose $|S| \leq s$. Without loss of generality we may assume $|S| = s$, since for larger values of s the acceptance probability may only increase. In this case $k' = k$, so that

$$\Pr[|R| > (1+\epsilon/3)k] = \Pr[|R'| \geq (1+\epsilon/3)k] < 9/\epsilon^2 k.$$

Given that $|R| \leq (1+\epsilon/3)k$, the prover can list all elements of R, so that $R = \{r_1, \ldots, r_l\}$ and $l \leq (1+\epsilon/3)k$. In particular, this ensures that $r \in R$ and the verifier accepts.

Now suppose $|S| \geq (1+\epsilon)s$, so that $k' > (1+\epsilon)k$. Then

$$\Pr[|R'| < (1+\epsilon/2)k] < \Pr\left[|R'| < \frac{1+\epsilon/2}{1+\epsilon}k'\right] < \Pr[|R'| < (1-\epsilon/3)k'] < 9/\epsilon^2 k' < 9/\epsilon^2 k.$$

Given that $|R| > (1+\epsilon/2)k$, what is the best strategy for a prover to make the verifier accept? Conditioned on $(h, h(r))$, r is uniformly distributed in R, so the best the prover can do is set $l = (1+\epsilon/3)k$ and pick $\{r_1, \ldots, r_l\}$ to be an *arbitrary* subset of R. In this case,

$$\Pr[r \in \{r_1, \ldots, r_l\}] = l/|R| < \frac{(1+\epsilon/3)k}{(1+\epsilon/2)k} < 1 - \epsilon/6. \quad \square$$

A.3 Average-case reductions from decision to search problems

Proof of Lemma 21. As in [BCGL89], we first reduce the search problem V to a search problem with a unique witness, then encode the bits of the witness in the language L'. The first step is based on the hashing argument of Valiant and Vazirani [VV86]. The reduction, as described below, only succeeds with probability 1/16, but this can be amplified to 2/3 by applying the reduction three times.

The inputs of the language L' are of the form (x, k, j, h), where x is an instance of L_V, k and j are integers between 0 and $m(|x|)$, and h is a pairwise independent hash function mapping $|x|$ bits to k bits (padded appropriately so the length of (x, k, j, h) is a fixed polynomial of $|x|$ only). Let w_i denote the ith bit of a string w. We define

$$(x, k, j, h) \in L' \text{ if there exists a } w \text{ of length } m(|x|) \text{ for which } h(w) = 0 \text{ and } w_i = 1.$$

It is immediate that $L' \in \mathrm{NP}$.

The reduction works as follows: On input $x \in \{0,1\}^n$, choose a random h uniformly at random and generate the queries $q_{kj} = (x, k, j, h)$ for all $1 \le k \le m$ and $1 \le j \le m$. Let $a_{kj} \in \{0,1\}$ denote the claimed answer to query q_{kj} and w_k be the concatenation $a_{k1} \dots a_{km}$. The decider looks for an index k such that w_k is a witness for x for all $1 \le j \le m$ and outputs w_k; if no such k is found the decider returns an arbitrary answer.

Let L^* be an arbitrary decision oracle that is δ-close to L'. Say an input x is *good* in L^* if for all $1 \le k \le m$, $1 \le j \le m$,

$$\Pr_{h, L^*}[L^*(x, k, j, h) = L(x, k, j, h)] > 15/16.$$

By a pigeonhole argument, $x \sim \mathcal{U}_n$ is good with probability at least $1 - O(\delta(m(n))^2)$. We show that the reduction succeeds on a good input with probability 1/16. By the Valiant-Vazirani argument, for $k = \lfloor \log_2 |\{w : V \text{ accepts } (x; w)\}| \rfloor$, with probability 1/8 there exists a unique w such that $h(w) = 0$. It follows that whenever x is good and $x \in L_V$, with probability at least 1/16, $L^*(x, k, j, h) = w_k$ for all $1 \le j \le m$, so the decider encounters the witness w_k. □

Remark. The argument can be strengthened to obtain a $O(\delta m(n))$-to-δ average-to-average reduction from (V, U) to (L', U) by applying a Hadamard code to the witness w in L' instead of revealing its bits.

A.4 Average-case reductions for arbitrary samplable distributions

Proof of Lemma 22. Let S denote the sampler that yields the distribution ensemble $\mathcal{D}$: S is a polynomial-time computable function $\{0,1\}^n \times \{0,1\}^{s(n)} \to \{0,1\}^n$, such that $S(1^n, \mathcal{U}_{s(n)}) = \mathcal{D}_n$.

We want to be able to map instances of V into instances of V' in such a way that witnesses for V can be recovered from witnesses for V', and so that for most x, the probability of an image of x in the uniform distribution is polynomially related to the probability of x in distribution $\mathcal{D}$.

Let V' be an NP-relation for language L', whose inputs are of the form (n, k, h_1, z, h_2) where

1. The integer n will denote the length of the input to the reduction coming from $\mathcal{D}$,
2. The integer $k \in [s]$ ($s = s(n)$) will encode the approximate likelihood of the input to the reduction according to $\mathcal{D}$,
3. $h_1 : \{0,1\}^n \to \{0,1\}^{k+6}$ is a pairwise independent hash function, and $z \in \{0,1\}^{k+6}$ is an element in the range of h_1;
4. $h_2 : \{0,1\}^s \to \{0,1\}^{s-k-3}$ is another pairwise independent hash function.

Suppose that the inputs (n, k, h_1, z, h_2) are padded appropriately so that their length depends on n only.

A pair (w, r) is an NP-witness for input (n, k, h_1, z, h_2) in V' if the following three conditions are satisfied: (1) $V(S(1^n, r); w) = 1$; (2) $h_1(S(1^n, r)) = z$; (3) $h_2(r) = 0$.

On input x, where $|x| = n$, the reduction produces queries $(n, k, h_1, h_1(x), h_2)$, for all possible values of k by choosing h_1 and h_2 uniformly at random. The decider looks at all answers (w_k, r_k), and returns w_k if $V(S(1^n, r_k); w_k) = 1$ for some k. If no such k is found, the decider returns the string 0^m.

Suppose F^* is a δ-approximate oracle for V' with respect to the uniform ensemble. Given $x \in \{0,1\}^n$, we call an instance (n, k, h_1, z, h_2) *good* for x if the following three conditions are satisfied:

1. $|x| = n$, $\lfloor -\log_2 \mathcal{D}(x) \rfloor = k$, and $h_1(x) = z$
2. There exists an r such that $h_1(S(1^n, r)) = z$ and $h_2(r) = 0$
3. If, for some r, $h_1(S(1^n, r)) = z$ and $h_2(r) = 0$, then $S(1^n, r) = x$.

Let $G(x)$ denote the set of all queries in L' that are good for x. It is immediate that the sets $G(x)$ are pairwise disjoint over all $x \in \{0,1\}^n$. On the one hand, we will show that, on input x, the reduction has a constant probability of producing a query that lands in $G(x)$. Moreover, conditioned on k, this query is uniformly distributed in $G(x)$. If $x \in L$ and F^* and V' agree on the query that falls within $G(x)$, then $F^*(x) = (w, r)$ with $S(1^n, r) = x$, so $V(x; w) = 1$. In addition, we will show that when $x \sim \mathcal{D}$, with probability at least $1 - \delta s$, F^* and V' do agree on a constant fraction of $G(x)$ for every k, so that the reduction has a constant probability of producing a query on which F^* and V' agree.

Claim 23. *Suppose $|x| = n$ and $\lfloor -\log_2 \mathcal{D}(x) \rfloor = k$. With probability $3/4$ over the choice of h_1 and h_2, the instance (n, k, h_1, z, h_2) is in $G(x)$.*

Proof of Claim. We first show that, with probability $7/8$, the instance satisfies the second condition for goodness, i.e., there exists an r such that $S(1^n, r) = x$ and $h_2(r) = 0$. If $S(1^n, r) = x$, let I_r be an indicator for the event $h_2(r) = 0$. By our choice of k, $|\{r : S(1^n, r) = x\}| \geq 2^{s-k}$, so that

$$\mathrm{E}\left[\sum_{r:S(1^n,r)=x} I_r\right] \geq 2^{s-k} \mathrm{E}[I_r] = 8.$$

As the I_r are pairwise independent, the variance of this sum is at most the expectation, so by Chebyshev's inequality at least one $I_r = 1$ with probability 7/8.

Now we look at the probability of satisfying the third condition for goodness. Fix r such that $S(1^n, r) \neq x$. By pairwise independence, $\Pr_{h_1}[h_1(S(1^n, r)) = h_1(x)] = 2^{-k-6}$, and independently, $\Pr_{h_2}[h_2(r) = 0] = 2^{-s+k+3}$. It follows that

$$\begin{aligned}\Pr[\exists r : S(1^n, r) \neq x \text{ and } h_1(S(1^n, r)) = h_1(x) \text{ and } h_2(r) = 0] \\ &\leq \sum\nolimits_{r:S(1^n,r)\neq x} \Pr[h_1(S(1^n, r)) = h_1(x)] \Pr[h_2(r) = 0] \\ &\leq \sum\nolimits_{r\in\{0,1\}^s} 2^{-k-6}2^{-s+k+3} = 1/8.\end{aligned}$$

It follows that both conditions for goodness are satisfied with probability at least 3/4. □

Claim 24. *For every x, $\mathcal{U}(G(x)) \geq \frac{3}{64}\mathcal{D}(x)/ns$.*

Proof of Claim. Consider a random string (n, k, h_1, z, h_2). With probability $1/ns$, $n = |x|$ and $k = \lfloor -\log_2 \mathcal{D}(x) \rfloor$. Conditioned on this, $z = h_1(x)$ with probability 2^{-k-3}. By the last Claim, with probability 3/4 over h_1 and h_2, $(n, k, h_1, h_1(x), h_2)$ is in $G(x)$. Putting this together,

$$\Pr[(n, k, h_1, z, h_2) \in G(x)] \geq \frac{1}{ns} \cdot \frac{3}{4} \cdot 2^{-k-3} \geq \frac{3}{64} \cdot \frac{\mathcal{D}(x)}{ns}. \quad \square$$

Let Z denote the set of all $x \in L_V$ for which

$$\Pr_{y\sim\mathcal{U},F^*}[V'(y, F^*(y)) = 1 \mid y \in G(x)] > 8/9,$$

so that if the k-th query q_k lands into $G(x)$, the answer (w_k, r_k) has a 8/9 probability of being a good witness for the query. It follows that, unconditionally, $V(q_k, F^*(q_k)) = 1$ with probability at least $8/9 \cdot 3/4 = 2/3$, and the decider is successful on the queries that come from S.

On the other hand, by the disjointness of the sets $G(x)$,

$$\begin{aligned}\delta &\geq \sum\nolimits_{x\in L_V} \mathcal{U}(G(x)) \Pr_{y\sim\mathcal{U}}[V'(y, F^*(y)) = 0 \mid y \in G(x)] \\ &> \sum\nolimits_{x\in\overline{Z}} \mathcal{U}(G(x)) \cdot \frac{1}{9} \\ &\geq \sum\nolimits_{x\in\overline{Z}} \frac{1}{9} \cdot \frac{3}{64} \cdot \frac{\mathcal{D}(x)}{ns} \text{ by Claim 24} \\ &= \Omega(\mathcal{D}(\overline{Z})/ns),\end{aligned}$$

so that $\mathcal{D}(\overline{Z}) = O(\delta ns)$. □