On Basing One-Way Functions on NP-Hardness

Adi Akavia* Oded Goldreich† Shafi Goldwasser‡ Dana Moshkovitz§

April 8, 2005

Abstract

We consider the possibility of basing one-way functions on NP-Hardness; that is, we study possible reductions from a worst-case decision problem to the task of average-case inverting a polynomial-time computable function f (i.e., reductions that are supposed to establish that the latter function is one-way based on a worst-case assumption regarding the decision problem). Our main findings are the following two negative results:

1. If given y one can efficiently compute $|f^{-1}(y)|$ then the existence of a (randomized) reduction of $\mathcal{NP}$ to the task of average-case inverting f implies that $\mathcal{NP} \subseteq \text{co}\mathcal{AM}$. Thus, it follows that such reductions cannot exist (unless $\mathcal{NP} \subseteq \text{co}\mathcal{AM}$).

   The result extends to functions for which the preimage size is efficiently verifiable via an AM protocol. For example, this includes regular functions with efficiently recognizable range.

   We stress that this result holds for any reduction, including *adaptive* ones. We note that the previously known negative results regarding worst-case to average-case reductions were essentially confined to *non-adaptive* reductions, whereas known positive results (regarding computational problems in the geometry of numbers) use adaptive reductions.

2. For any function f, the existence of a (randomized) *non-adaptive* reduction of $\mathcal{NP}$ to the task of average-case inverting f implies that $\mathcal{NP} \subseteq \text{co}\mathcal{AM}$.

   This result improves over the previous negative results (for this case) that placed $\mathcal{NP}$ in non-uniform co$\mathcal{AM}$.

Our work builds on the previous works of Feigenbaum and Fortnow (*SIAM Journal on Computing*, 1993) and Bogdanov and Trevisan (*44th FOCS*, 2003), while capitalizing on the additional "computational structure" of the search problem associated with the task of inverting polynomial-time computable functions. We believe that our results illustrate the gain of directly studying the context of one-way functions rather than inferring results for it from a the general study of worst-case to average-case reductions.

Area: Cryptography and Complexity.

Keywords: One-Way Functions, Worst-Case to Average-Case reductions, Adaptive versus Non-adaptive reductions, Interactive Proof Systems.

Note: Section 3 (on page 11) is kind of discretionary.

*akavia@mit.edu

†oded.goldreich@weizmann.ac.il

‡shafi@theory.csail.mit.edu

§dana.moshkovitz@weizmann.ac.il

1 Introduction

One-way functions are functions that are easy to compute but hard to invert, where the hardness condition refers to the average-case complexity of the inverting task. The existence of one-way functions is the cornerstone of modern cryptography: almost all cryptographic primitives imply the existence of one-way functions, and most of them can be constructed based either on the existence of one-way functions or on related (but seemingly stronger) versions of this assumption.

As noted above, the hardness condition of one-way functions is an average-case complexity condition. Clearly, this average-case hardness condition implies a worst-case hardness condition; that is, the existence of one-way functions implies that $\mathcal{NP}$ is not contained in $\mathcal{BPP}$. A puzzling question of fundamental nature is whether or not the necessary worst-case condition is a sufficient one; that is, can one base the existence of one-way functions on the assumption that $\mathcal{NP}$ is not contained in $\mathcal{BPP}$.

More than two decades ago, Brassard [Br] observed that the inverting task associated with a one-way *permutation* (or, more generally, a 1-1 one-way function) cannot be $\mathcal{NP}$-hard, unless $\mathcal{NP} = \text{co}\mathcal{NP}$. The question was further addressed (indirectly), in the works of Feigenbaum and Fortnow [FeFo] and Bogdanov and Trevisan [BoTr], which focused on the study of worst-case to average-case reductions among decision problems.

1.1 Our Main Results

In this paper we re-visit the aforementioned question, but do so explicitly. We study possible reductions from a worst-case decision problem to the task of average-case inverting a polynomial-time computable function (i.e., reductions that are supposed to establish that the latter function is one-way based on a worst-case assumption regarding the decision problem). Specifically, we consider (randomized) reductions of $\mathcal{NP}$ to the task of average-case inverting a polynomial-time computable function f, and capitalize on the additional "computational structure" of the search problem associated with the inverting task. This allows us to strengthen previously known negative results, and obtain the following two main results:

1. If given y one can efficiently compute $|f^{-1}(y)|$ then the existence of a (randomized) reduction of $\mathcal{NP}$ to the task of average-case inverting f implies that $\mathcal{NP} \subseteq \text{co}\mathcal{AM}$.

   The result extends to functions for which the preimage size is efficiently verifiable via an AM protocol. For example, this includes regular functions (cf., e.g., [GKL]) with efficiently recognizable range. Recall that $\mathcal{AM}$ is the class of sets having two-round interactive proof systems, and that it is widely believed that co$\mathcal{NP}$ is not contained in $\mathcal{AM}$ (equiv., $\mathcal{NP}$ is not contained in co$\mathcal{AM}$). Thus, it follows that such reductions cannot exist (unless $\mathcal{NP} \subseteq \text{co}\mathcal{AM}$).

   We stress that this result holds for any reduction, including *adaptive* ones. We note that the previously known negative results regarding worst-case to average-case reductions were essentially confined to *non-adaptive* reductions (cf. [FeFo, BoTr], where [FeFo] also handles restricted levels of adaptivity).

2. For any (polynomial-time computable) function f, the existence of a (randomized) *non-adaptive* reduction of $\mathcal{NP}$ to the task of average-case inverting f implies that $\mathcal{NP} \subseteq \text{co}\mathcal{AM}$.

   This result improves over the previous negative results of [FeFo, BoTr] that placed $\mathcal{NP}$ in non-uniform co$\mathcal{AM}$ (instead of in *uniform* co$\mathcal{AM}$).

These negative results can be interpreted in several ways: see discussion in Section 3.

1.2 Relation to Feigenbaum-Fortnow and Bogdanov-Trevisan

Our work is inspired by two previous works. The first work, by Feigenbaum and Fortnow [FeFo], posed the question of whether or not $\mathcal{NP}$-complete problems can be *random self-reducible*. That is, can (worst case) instances of $\mathcal{NP}$-complete problems be reduced to one or more *random instances, where the latter instances are drawn according to a predetermined distribution*. The main result of [FeFo] is that if such (*non-adaptive*) reductions exist, then co$\mathcal{NP}$ is in a non-uniform version of $\mathcal{AM}$, denoted $\mathcal{AM}_{\rm poly}$. Non-uniformity was used in their work to encode statistics about the target distribution of the reduction.

Bogdanov and Trevisan [BoTr] start by viewing the result of [FeFo] as a result about the impossibility of worst-case to average-case reductions for $\mathcal{NP}$-complete problems. They note that even if one cares about the average-case complexity of a problem with respect to a specific distribution (e.g., the uniform one) then it needs not be the case that a worst-case to average-case reduction must make queries according to this distribution. Furthermore, the distribution of queries may depend on the input to the reduction, and so statistics regarding it cannot be given as advice. Nevertheless, combining the ideas of [FeFo] with additional ideas (some borrowed from the study of locally-decodable codes [KaTr]), Bogdanov and Trevisan showed that any *non-adaptive* reduction of (worst-case) $\mathcal{NP}$ to the average-case complexity of $\mathcal{NP}$ (with respect to any sampleable distribution) implies that co$\mathcal{NP} \subseteq \mathcal{AM}_{\rm poly}$.

Although a main motivation of [BoTr] is the question of basing one-way functions on worst-case $\mathcal{NP}$-hardness, its focus (like that of [FeFo]) is on *decision problems*. Using known reductions between search and decision problems in the context of distributional problems [BCGL, ImLe], Bogdanov and Trevisan [BoTr] also derive implications on the (im)possibility of basing one-way functions on $\mathcal{NP}$-hardness. In particular, they conclude that if there exists an $\mathcal{NP}$-complete set for which deciding any instance is *non-adaptively* reducible to *inverting a one-way function* (or, more generally, to a search problem with respect to a sampleable distribution), then co$\mathcal{NP} \subseteq \mathcal{AM}_{\rm poly}$.

We emphasize that the *techniques* of [BoTr] refer explicitly only to decision problems, and do not relate to the underlying search problems (e.g., inverting a supposedly one-way function). In doing so, they potentially lose twice: they lose the extra structure of search problems and they lose the additional structure of the task of inverting polynomial-time computable functions. To illustrate the latter aspect, we re-formulate the problem of inverting a polynomial-time computable function as follows (or rather spell out what it means in terms of search problems). The problem of (average-case) inverting f on the distribution $f(U_n)$, where U_n denotes the uniform distribution over $\{0,1\}^n$, has the following features:

1. The problem is in NP; that is, the solution is relatively short and given an instance of the problem (i.e., y) and a (candidate) solution (i.e., x), it is easy to verify that the solution is correct (i.e., $y = f(x)$).
2. We care about the average-case complexity of the problem; that is, the probability that an efficient algorithm given a random (efficiently sampled) instance y (i.e., $y \leftarrow f(U_n)$) finds $x \in f^{-1}(y)$.
3. There exists an efficient algorithm that generates random instance-solution pairs (i.e., pairs (y, x) such that $y = f(x)$, for uniformly distributed $x \in \{0,1\}^n$).

Indeed, the first two items are common to all average-case NP-search problems (with respect to sampleable distributions), but the third item is specific to the context of one-way functions (cf. [Go, Sec. 2.1]). A generic sampleable distribution of instances is not necessarily coupled with a corresponding sampleable distribution of random instance-solution pairs. Indeed, capitalizing on the third item is the source of our success to obtain stronger (negative) results regarding the possibility of basing one-way functions on $\mathcal{NP}$-hardness.

The works [BoTr, FeFo] fall short of a general impossibility result in two ways. First, they only consider *non-adaptive* reductions, whereas the celebrated worst-case to average-case reductions of lattice problems (cf. [Aj, MiRe]) are adaptive. Furthermore, the positive results seem to illustrate the power of adaptive versus non-adaptive reductions.[1] Second, [BoTr, FeFo] reach conclusions involving a *non-uniform* complexity class (i.e., $\mathcal{AM}_{\rm poly}$). Non-uniformity seems an artifact of their techniques, and

[1]We comment that the power of adaptive versus non-adaptive reductions has been studied in various works (e.g., [FFLS, HNOS, BaLa]). It is known that if $NP \not\subseteq BPE$, then there exists a set in $\mathcal{NP} \setminus \mathcal{BPP}$ that is adaptively random self-reducible but not non-adaptively random self-reducible.

one may hope to conclude that co$\mathcal{NP} \subseteq \mathcal{AM}$ rather than co$\mathcal{NP} \subseteq \mathcal{AM}_{\rm poly}$. (One consequence of the uniform conclusion is that it implies that the polynomial time hierarchy collapses to the second level, whereas the non-uniform conclusion only implies a collapse to the third level.)

1.3 The Benefits of Direct Study of One-Way Functions

As stated before, working directly with one-way functions allows us to remove both the aforementioned shortcomings. That is, we get rid of the non-uniformity altogether, and obtain a meaningful negative result for the case of general (adaptive) reductions. Specifically, working directly with one-way functions allows us to consider natural special cases of potential one-way functions, which we treat for general (i.e., possibly adaptive) reductions.

One special case of potential one-way functions, which received some attention in the past (e.g., [GKL, GIL+, DiIm, HHK+]), is that of *regular* one-way functions. Loosely speaking, in such a function f, each image of f has a number of preimages that is (easily) determined by the length of the image. We prove that any reduction (which may be *fully adaptive*) of $\mathcal{NP}$ to inverting a regular polynomial-time computable function that has an efficiently recognizable range (possibly via an AM-protocol) implies co$\mathcal{NP} \subseteq \mathcal{AM}$. More generally, this holds for any function f for which there is an AM-protocol for determining the number of inverses $|f^{-1}(y)|$ of each given y. We call such functions size-verifiable, and note that they contain all functions for which (given y) one can efficiently compute $|f^{-1}(y)|$.

As stated above, we believe that the study of the possibility of basing one-way functions on worst-case $\mathcal{NP}$-hardness is the most important motivation for the study of worst-case to average-case reductions for $\mathcal{NP}$. In such a case, one should consider the possible gain from studying the former question directly, rather than as a special case of a more general study. We believe that the results presented in this paper indicate such gains. Firstly, working directly in the context of one-way function enabled us to get rid of the non-uniformity in all our results (by replacing non-uniform advice that provide needed statistics with AM-protocols designed to provide these statistics). Secondly, the context of one-way function enabled us to consider meaningful types of one-way functions and to establish even stronger results for them. We hope that this framework may lead to resolving the general question of the possibility of basing *any* one-way function on worst-case $\mathcal{NP}$-hardness via *any* reduction. In light of the results of this paper, we are tempted to conjecture an impossibility result (pending, as usual, on co$\mathcal{NP} \not\subseteq \mathcal{AM}$).

Organization of the rest of this work. In Section 2, we provide an overview of our proofs as well as a formal statement of our main results. Detailed proofs can be found in the appendix's sections (i.e., preliminaries are in Appendix A, the treatment of adaptive reductions is in Appendix B, and the treatment of general functions is in Appendix C). In Section 3 we discuss possible interpretations of our negative results (as well as those of [FeFo, BoTr]).

2 Overview of Results and Proofs

Having observed the potential benefit of working explicitly with the inverting task of a function f, materializing this benefit is the bulk of the technical challenge and the technical novelty of this work.

Let us first clarify what we mean by saying that a decision problem L is (efficiently and randomly) reducible to the problem of inverting a one-way function f. We take the straightforward interpretation (while using several arbitrary choices, like in the threshold determining an inverting oracle):

Definition 1 (inverting oracles and reductions). *A function $\mathcal{O} : \{0,1\}^* \to \{0,1\}^*$ is called a* (average-case) f-inverting oracle *if, for every n, it holds that $\Pr[\mathcal{O}(f(x)) \in f^{-1}(f(x))] \geq 1/2$, where the probability is taken uniformly over $x \in \{0,1\}^n$. For a probabilistic oracle machine R, we denote by $R^{\mathcal{O}}(w)$ a random variable representing the output of R on input w and access to oracle $\mathcal{O}$, where the probability space is taken uniformly over the probabilistic choices of machine R* (i.e., its randomness)*. A probabilistic polynomial-time oracle machine R is called a* reduction of L to (average-case) inverting f *if, for every*

$w \in \{0,1\}^*$ *and any* f*-inverting oracle* $\mathcal{O}$*, it holds that* $\Pr[R^{\mathcal{O}}(w) = \chi_L(w)] \geq 2/3$*, where* $\chi_L(w) = 1$ *if* $w \in L$ *and* $\chi_L(w) = 0$ *otherwise.*

A reduction as in Definition 1 may only establish that f is a weak one-way function (i.e., that f cannot be inverted with probability exceeding 1/2 on every input length), which makes our impossibility results even stronger.[2] Throughout this work, the function f will always be polynomial-time computable, and for simplicity we will also assume that it is length preserving (i.e., $|f(x)| = |x|$ for all x).

Let us take a closer look at the reduction R. On input w, it may ask polynomially many queries to the inverting oracle. In adaptive reductions, later queries may depend on the oracle answers to earlier queries. In non-adaptive reductions all queries are computed in advance (based solely on the input w and the randomness r). For simplicity of presentation, we assume all queries are of length $|w|$.

High-level structure of our proofs and their challenges. Our proofs all work via the contrapositive. Suppose, that there exists a reduction R from deciding an (NP-complete language) L to inverting the function f. We aim to use this reduction to give an AM-protocol for $\overline{L}$. (A similar AM-protocol can be given for L itself, but there is no point in doing so because $L \in \mathcal{NP}$ by hypothesis.)

As in [FeFo, BoTr], the main backbone of our AM-protocol for $\overline{L}$ is for the verifier to emulate the reduction R on input w and decide whether or not $w \in \overline{L}$ according to R's output. Of course, the verifier cannot run the reduction fully on its own, because the reduction requires access to an f-inverting oracle. Instead, the prover will play the role of the inverting oracle, thus enabling the emulation of the reduction. Needless to say, the verifier will check that all answers are actually f-preimages of the corresponding queries (and for starters we will assume that all queries are in the image of f). Since we aim at a constant-round protocol, we send all queries to the prover in one round, which in the case of an adaptive reduction *requires* to send the randomness r of the reduction to the prover. Note that also in the non-adaptive case, we may as well just send r to the prover, because the prover may anyhow be able to determine r from the queries.

The fact that r is given (explicitly or implicitly) to the prover is the source of all difficulties that follow. It means that the prover need not answer the queries obliviously of other queries (or of r), but may answer the queries depending on r. In such a case, the prover's answers (when considering all possible r) are not consistent with any single oracle. Indeed, all these difficulties arise only in case f is not 1-1 (and indeed in case f is 1-1 the correct answer is fully determined by the query). We stress that the entire point of this study is the case in which f is not 1-1. In the special case that f is 1-1 (and length preserving), inverting f cannot be $\mathcal{NP}$-hard for rather obvious reasons (as has been well-known for a couple of decades; cf. [Br]).[3]

To illustrate what may happen in the general case, consider a 2-to-1 function f. Given an arbitrary reduction of L to inverting f, consider a modified reduction that tosses n additional coins $\rho_1, ..., \rho_n$, issues n additional queries, and halts without output if and only if for $i = 1, ..., n$ the i-th additional query is answered with the $(\rho_i + 1)$-st corresponding preimage (in lexicographic order). This reduction works with probability that is very close to the original one, but a cheating prover can always cause its emulation to halt without output.

[2] In contrast, the standard definition of one-way function requires that any efficient inverting algorithm succeeds with negligible probability (i.e., probability that is smaller than $1/\text{poly}(n)$ on all but finitely many n's). Here we relax the security requirement in two ways (by requiring more of a successful inverting algorithm): first, we require that the inverting algorithm be successful on any input length, and second that the success probability exceeds 1/2 rather than $1/\text{poly}(n)$.

[3] Intuitively, inverting such an f (which is a search problem in which each instance has a unique solution) corresponds to a decision problem in $\mathcal{NP} \cap \text{co}\mathcal{NP}$ (i.e., given (y, i) determine the i-th bit of $f^{-1}(y)$). Thus, the fact that inverting f cannot be $\mathcal{NP}$-hard (unless $\mathcal{NP} = \text{co}\mathcal{NP}$) is analogous to the fact that sets in $\mathcal{NP} \cap \text{co}\mathcal{NP}$ cannot be $\mathcal{NP}$-hard (again, unless $\mathcal{NP} = \text{co}\mathcal{NP}$). In contrast, in case f is not 1-1, the corresponding decision problems are either not known to be in $\mathcal{NP} \cap \text{co}\mathcal{NP}$ or are *promise problems* (cf. [ESY]) in the "promise problem class" analogue of $\mathcal{NP} \cap \text{co}\mathcal{NP}$. Recall that promise problems in the latter class *may be* $\mathcal{NP}$-hard even if $\mathcal{NP} \neq \text{co}\mathcal{NP}$ (see [ESY]).

A different way of looking at things is that the reduction guarantees that, for any adequate (f-inverting) oracle $\mathcal{O}$, with probability 2/3 over the choices of r, machine R decides correctly when given oracle access to $\mathcal{O}$. However, it is possible that for every r there exists an oracle $\mathcal{O}_r$ such that R, when using coins r, decides incorrectly when given oracle access to $\mathcal{O}_r$. If this is the case (which we cannot rule out) then the prover may cheat by answering like the bad oracle $\mathcal{O}_r$. In the rest of this section, we provide an outline of how we deal with this difficulty in each of the two cases (i.e., size-verifiable functions and non-adaptive reductions).

2.1 Size-Verifiable Functions (Adaptive Reductions)

Recall that our aim is to present an AM-protocol for $\overline{L}$, when we are given a general (adaptive) reduction R of the worst-case decision problem of L to average-case inverting f. We denote by q the number of queries made by R, by $R(w, r, a_1, ..., a_{i-1})$ the i-th query made by R on input w and randomness r after receiving the oracle answers $a_1, ..., a_{i-1}$, and by $R(w, r, a_1, ..., a_q)$ the corresponding final decision. Recall that for simplicity, we assume that all queries are of length $n \stackrel{\text{def}}{=} |w|$. In the bulk of this subsection we assume that, given y, one can efficiently determine $|f^{-1}(y)|$.

A very simple case: As a warm-up we first assume that $|f^{-1}(y)| \leq \text{poly}(|y|)$, for every y. In this case, on common input w, the parties proceed as follows.

1. The verifier selects uniformly coins r for the reduction, and sends r to the prover.
2. Using r, the prover emulates the reduction as follows. When encountering a query y, the prover uses the lexicographically first element of $f^{-1}(y)$ as the oracle answer (and uses $\perp$ if $f^{-1}(y) = \phi$). Thus, it obtains the corresponding list of queries $y_1, ..., y_q$, which it sends to the verifier along with the corresponding sets $f^{-1}(y_1), ..., f^{-1}(y_q)$.
3. Upon receiving $y_1, ..., y_q$ and $A_1, ..., A_q$, the verifier checks, for every i, that $|A_i| = |f^{-1}(y_i)|$ and that $f(x) = y_i$ for every $x \in A_i$. Letting a_i denote the lexicographically first element of A_i, the verifier checks that $R(w, r, a_1, ..., a_{i-1}) = y_i$ for every i. The verifier accepts w (as a member of $\overline{L}$) if and only if all checks are satisfied and $R(w, r, a_1, ..., a_q) = 0$.

Note that the checks performed by the verifier "force" the prover to emulate a uniquely determined (perfect) inverting oracle (i.e., one that answers each query y with the lexicographically first element of $f^{-1}(y)$). Thus, the correctness of the reduction implies the completeness and soundness of the above AM-protocol.

In general, however, the size of $f^{-1}(y)$, for y in the range of f may not be bounded by a polynomial in n (where $n = |y| = |w|$). In this case, we cannot afford to have $f^{-1}(y)$ as part of a message in the protocol (because it is too long). The obvious idea is to have the verifier send an adequate random hash function $h : \{0,1\}^n \to \{0,1\}^\ell$ and let the prover answer with $h^{-1}(0^\ell) \cap f^{-1}(y)$ (rather than with $f^{-1}(y)$), where $\ell = \lfloor (\log_2 |f^{-1}(y)|/\text{poly}(n)) \rfloor$. The problem is that in this case the verifier cannot check the "completeness" of the list of preimages (because it cannot compute $|h^{-1}(0^\ell) \cap f^{-1}(y)|$), which allows the prover to omit a few members of $h^{-1}(0^\ell) \cap f^{-1}(y)$ at its choice. Recall that this freedom of choice (of the prover) may obliterate the soundness of the protocol.

The solution is that, although we have no way of determining the size of $h^{-1}(0^\ell) \cap f^{-1}(y)$, we do know that its expected size is exactly $|f^{-1}(y)|/2^\ell$, where the expectation is taken over the choice of h (assuming indeed that a random h maps each point in $\{0,1\}^n$ uniformly on $\{0,1\}^\ell$). Furthermore, the prover cannot add elements to $h^{-1}(0^\ell) \cap f^{-1}(y)$ (because the verifier can verify membership in this set), it can only omit elements. But if the prover omits even a single element, it ends-up sending a set that is noticeably smaller than its expected size (because the expected size of $h^{-1}(0^\ell) \cap f^{-1}(y)$ is a polynomial in n). Thus, if we repeat the process many times, the prover cannot afford to cheat in most of these repetitions, because in that case the statistics will deviate from the expectation by too much.

Before turning to the specific implementation of this idea, we mention that the above reasoning mimics the main idea of Feigenbaum and Fortnow [FeFo] (also used by Bogdanov and Trevisan [BoTr]).

Similarly to their setting, we also have a situation in which the prover may cheat (without being detected) only in one direction. In their setting cheating was possible by claiming that a string that is in an NP-set (at the target of the reduction) is not in that set, whereas here it is in sending a proper subset of a set of preimages under f and h (which also has an efficient membership test). In both settings, it is impossible to cheat in the other direction (i.e., claim that a non-member of an NP-set is in that set, or send a set containing some non-preimages).

Protocol for the general case: In the following protocol we use families of hash functions of very high quality (e.g., poly(n)-wise independent ones). Specifically, in addition to requiring that a random $h : \{0,1\}^n \to \{0,1\}^\ell$ maps each point uniformly, we require that, for a *suitable polynomial* p and for any $S \subseteq \{0,1\}^n$ of size at least $p(n) \cdot 2^\ell$, with overwhelmingly high probability over the choice of h it is the case that $|h^{-1}(0^\ell) \cap S| < 2|S|/2^\ell$. In particular, the probability that this event does not occur is so small that, when conditioning on this event, the expected size of $|h^{-1}(0^\ell) \cap S|$ is $(1 \pm 2^{-n}) \cdot |S|/2^\ell$. (Thus, under this conditioning and for S as above, the variance of $2^\ell|h^{-1}(0^\ell) \cap S|/|S|$ is less than 2.)

1. The verifier selects uniformly $m = n \cdot q^2 p(n)^2 = \text{poly}(n)$ sequences of coins, $r^{(1)}, ..., r^{(m)}$ for the reduction, and sends them to the prover. In addition, for each $k = 1, ..., m$, $i = 1, ..., q$ and $\ell = 1, ..., n$, it selects and sends a random hash function $h_{k,i,\ell} : \{0,1\}^n \to \{0,1\}^\ell$.

   To streamline the following description, for $j \leq 0$, we artificially define $h_{k,i,j}$ such that $h_{k,i,j}^{-1}(0^j) \stackrel{\text{def}}{=} \{0,1\}^n$. In such a case, $S \cap h_{k,i,j}^{-1}(0^j) = S$, and so an instruction to do something with the former set merely means using the latter set.

2. For every $k = 1, ..., m$, the prover uses $r^{(k)}$ to emulate the reduction as follows. When encountering the i-th query, $y_i^{(k)}$, it determines $\ell_i^{(k)} = \lfloor (\log_2 |f^{-1}(y_i^{(k)})|/p(n)) \rfloor$, and uses the lexicographically first element of $f^{-1}(y_i^{(k)}) \cap h_{k,i,\ell_i^{(k)}}^{-1}(0^{\ell_i^{(k)}})$ as the oracle answer (and uses $\perp$ if the latter set is empty).

   Thus, it obtains the corresponding list of queries $y_1^{(k)}, ..., y_q^{(k)}$, which it sends to the verifier along with the corresponding sets $f^{-1}(y_1^{(k)}) \cap h_{k,1,\ell_1^{(k)}}^{-1}(0^{\ell_1^{(k)}}), ..., f^{-1}(y_q^{(k)}) \cap h_{k,q,\ell_q^{(k)}}^{-1}(0^{\ell_q^{(k)}})$.

   We assume that none of the latter sets has size greater than $4p(n)$. Note that the bad event occurs with negligible probability, and in such a case the prover halts and the verifier rejects. (Otherwise, all mq sets are sent in one message.)

3. Upon receiving $y_1^{(1)}, ..., y_q^{(1)}, ..., y_1^{(m)}, ..., y_q^{(m)}$ and $A_1^{(1)}, ..., A_q^{(1)}, ..., A_1^{(m)}, ..., A_q^{(m)}$, the verifier conducts the following checks:

   (a) For every $k = 1, ..., m$ and $i = 1, ..., q$, the verifier checks that for every $x \in A_i^{(k)}$ it holds that $f(x) = y_i^{(k)}$ and $h_{k,i,\ell_i^{(k)}}(x) = 0^{\ell_i^{(k)}}$, where $\ell_i^{(k)} = \lfloor (\log_2 |f^{-1}(y_i^{(k)})|/p(n)) \rfloor$. Letting $a_i^{(k)}$ be the lexicographically first element of $A_i^{(k)}$, it checks that $R(w, r^{(k)}, a_1^{(k)}, ..., a_{i-1}^{(k)}) = y_i^{(k)}$.

   (b) For every $i = 1, ..., q$, it checks that

   $$\frac{1}{m} \cdot \sum_{k=1}^{m} \frac{2^{\ell_i^{(k)}} \cdot |A_i^{(k)}|}{|f^{-1}(y_i^{(k)})|} > 1 - \frac{1}{100q \cdot p(n)} \tag{1}$$

   where $0/0$ is defined as 1.

   The verifier accepts w if and only if all the foregoing checks are satisfied and it holds that $R(w, r^{(k)}, a_1^{(k)}, ..., a_q^{(k)}) = 0$ for a uniformly selected $k \in \{1, ..., m\}$.

Analysis of the Protocol. We first note that the additional checks added to this protocol have a negligible effect on the *completeness* condition: the probability that either $|f^{-1}(y_i^{(k)}) \cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})| > 4p(n)$ for some i, k or that Eq. (1) is violated for some i is exponentially vanishing.[4] Turning to the soundness condition, we note that the checks performed by the verifier force the prover to use $A_i^{(k)} \subseteq T_i^{(k)} \stackrel{\text{def}}{=} f^{-1}(y_i^{(k)}) \cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})$. Also, with overwhelmingly high probability, for every $i = 1, ..., q$, it holds that

$$\frac{1}{m} \cdot \sum_{k=1}^{m} \frac{2^{\ell_i^{(k)}} \cdot |f^{-1}(y_i^{(k)}) \cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})|}{|f^{-1}(y_i^{(k)})|} < 1 + \frac{1}{100q \cdot p(n)} \tag{2}$$

Combining Eq. (1) and Eq. (2), and recalling that $A_i^{(k)} \subseteq T_i^{(k)}$ (and $|f^{-1}(y_i^{(k)})| < 2p(n) \cdot 2^{\ell_i^{(k)}}$), it follows that $(1/m) \cdot \sum_{k=1}^{m} (|T_i^{(k)} \setminus A_i^{(k)}|/2p(n)) < 2/(100q \cdot p(n))$ for every i. Thus, for each i, the probability over a random k that $A_i^{(k)} \neq T_i^{(k)}$ is at most $1/25q$. It follows that for a random k, the probability that $A_i^{(k)} = T_i^{(k)}$ for all i's is at least $1 - (1/25)$. In this case, the correctness of the reduction implies the soundness of the foregoing AM-protocol.

The above description presumes that the verifier can determine the size of the set of f-preimages of any string. The analysis can be easily extended to the case that the verifier can only check the correctness of the size claimed and proved by the prover. That is, we refer to the following definition.

Definition 2 (Size Verifiable). *We say that a function $f\colon \{0,1\}^* \to \{0,1\}^*$ is* size verifiable *if there is a constant-round proof system for the set $\{(y, |f^{-1}(y)|) : y \in \{0,1\}^*\}$.*

A natural example of a function that is size verifiable (and for which the relevant set is not known to be in $\mathcal{BPP}$) is the integer multiplication function. That is, we consider the function that maps pairs of integers (which are not necessarily prime or of the same length) to their product. In this case the set $\{(y, |f^{-1}(y)|) : y \in \{0,1\}^*\}$ is in $\mathcal{NP}$ (i.e., the NP-witness is the prime factorization) but is widely believed not to be in $\mathcal{BPP}$ (e.g., it is believed to be infeasible to distinguish product of two $(n/2)$-bit random primes from the product of three $(n/3)$-bit long random primes).

Theorem 3 (Adaptive Reductions). *Unless co$\mathcal{NP} \subseteq \mathcal{AM}$, there exists no reduction* (even not an adaptive one) *from deciding an NP-complete language to inverting a* size-verifiable *polynomial-time computable function.*

Namely, it is unlikely that size-verifiable one-way functions can be based on NP-hardness. We note that the result can be extended to functions that are "approximately size-verifiable" (covering the "approximable preimage-size" function of [HHK+] as a special case). For formal proofs see Appendix B.

2.2 Non-Adaptive Reductions (General Functions)

We now turn to outline the proof of our second main result.

Theorem 4 (General Functions). *Unless co$\mathcal{NP} \subseteq \mathcal{AM}$, there exists no* non-adaptive *reduction from deciding an NP-complete language to inverting a polynomial-time computable function.*

Considering the AM-protocol used in the adaptive case, we note that in the current case the verifier cannot necessarily compute (or even directly verify claims about) the size of sets of f-preimages of the reduction's queries. Indeed, known lower-bound protocols (cf. [GoSi]) could be applied to these sets, but known upper-bound protocols (cf. [AiHa]) cannot be applied because they require that the verifier has or can obtain a random (let alone secret) member of these sets. Fortunately, adapting the ideas

[4] Recall that here we refer to the case that $A_i^{(k)} = f^{-1}(y_i^{(k)}) \cap h^{-1}_{k,i,\ell_i^{(k)}}(0^{\ell_i^{(k)}})$. Thus, regarding Eq. (1), we note that the l.h.s is the average of m independent random variables, each having constant variance. Applying Chernoff bound, the probability that Eq. (1) is violated is upperbounded by $\exp(-\Omega(m/(100q \cdot p(n))^2)) = \exp(-\Omega(n))$.

of [BoTr] to the current setting, allows to overcome this difficulty and to obtain, not only non-uniform AM-protocols (for co$\mathcal{NP}$), but even uniform ones (thus, implying co$\mathcal{NP} \subseteq \mathcal{AM}$).

Here R is a *non-adaptive* reduction of some set $L \in \mathcal{NP}$ to the average-case inverting of an arbitrary (polynomial-time computable) function f, and our goal again is to show that $\overline{L} \in \mathcal{AM}$. We may assume, without loss of generality, that the queries of $R(w, \cdot)$ are *identically distributed* (but typically *not* independently distributed), and represent this distribution by the random variable R_w; that is, $\Pr[R_w = y] = |\{r \in \{0,1\}^{n'} : R(w,r) = y\}|/2^{n'}$, where n' denotes the number of coins used by $R(w, \cdot)$.

A simple case (analogous to [FeFo]): We first consider the case that R's queries are distributed identically to $F_n \stackrel{\text{def}}{=} f(U_n)$, where U_n denotes the uniform distribution over $\{0,1\}^n$. In this case, we ask the prover to provide $|f^{-1}(y_i^{(k)})|$ along with each query $y_i^{(k)}$ made in the emulation of $R(w, r^{(k)})$, and ask for lower-bound proofs (cf., [GoSi]) regarding the claimed sizes.[5] To prevent the prover from understating these sizes, we compare the value of $(1/qm) \cdot \sum_{i=1}^{q} \sum_{k=1}^{m} \log_2 |f^{-1}(y_i^{(k)})|$ to the expected value of $\log_2 |f^{-1}(f(U_n))|$, where here and below we define $\log_2 0$ as -1 (in order to account for the case of queries that have no preimages). Mimicking [FeFo], one may suggest that the latter value (i.e., $\text{Exp}[\log_2 |f^{-1}(F_n)|]$) be given as a non-uniform advice, but we can do better: We may ask the prover to supply $\text{Exp}[\log_2 |f^{-1}(f(U_n))|]$ and prove its approximate correctness using the following protocol.

> The verifier uniformly selects $x_1, ..., x_m \in \{0,1\}^n$, computes $y_i = f(x_i)$ for every i, sends $y_1, ..., y_m$ to the prover and asks for $|f^{-1}(y_1)|, ..., |f^{-1}(y_m)|$ along with lower and upper bound constant-round interactive proofs. (As usual, the lower-bound AM-protocol of [GoSi] (or [GVW]) can be applied because membership in the corresponding sets can be easily verified.) The (point is that the) upper-bound protocol of [AiHa] can be applied here, because the verifier has secret random elements of the corresponding sets.

Recall that the lower-bound protocol (of [GoSi] or [GVW]) guarantee that the prover cannot overstate any set size by more than an $\varepsilon = 1/\text{poly}(n)$ factor (without risking detection with overwhelmingly high probability). Thus, we will assume throughout the rest of this section that the prover never overstates set sizes (by more than such a factor). The analysis of understated set sizes is somewhat more delicate, firstly because (as noted) the execution of upper-bound protocols requires the verifier to have a secret random element in the set, and secondly because an understatement by a factor of ε is only detected with probability ε (or so). Still this means that the prover cannot significantly understate many sets sizes and go undetected. Specifically, if the prover understates the size of $f^{-1}(y_i)$ by more than an ε factor for at least n/ε of the y_i's then it gets detected with overwhelmingly high probability. Using a suitable setting of parameters, this establishes the value of $\text{Exp}[\log_2 |f^{-1}(f(U_n))|]$ up to a sufficiently small additive term, which suffices for our purposes. Specifically, as in Section 2.1, such a good approximation of $\text{Exp}[\log_2 |f^{-1}(f(U_n))|]$ forces the prover not to understate the value of $|f^{-1}(y_i^{(k)})|$ by more than (say) a $1/10p(n)$ factor for more than (say) $m/10$ of the possible pairs (i,k). (Note that, unlike in Section 2.1, here we preferred to consider the sum over all (i,k)'s rather than q sums, each corresponding to a different i.)[6]

[5] Furthermore, in this case the corresponding hashing functions (i.e., the $h_{k,i,\ell}$) can be sent after all set sizes has been claimed. In fact, it is important to do so (or introduce an alternative modification) in order to prevent possible control of the prover on the hashing function being used. Recall that in the adaptive case, the hashing function in use (for query y) is determined by $\ell = \lfloor(\log_2 |f^{-1}(y)|/p(n))\rfloor$, but the verifier knows $|f^{-1}(y)|$ and thus the prover has no control on the value of ℓ. In the current context, the prover may be able to cheat a little about the value of $|f^{-1}(y)|$, without being caught, and this may (sometimes) cause a change of one unit in the value of ℓ (and thus allow for a choice among two hash functions). Sending the hash function after ℓ is determines (by the prover) eliminates the potential gain from such a cheating. An alternative modification is to set $\ell = \lfloor(\rho + \log_2 s_y/p(n))\rfloor$, where s_y is prover's claim regarding the size of $|f^{-1}(y)|$ and $\rho \in [0,1]$ is a uniformly chosen randomization selected and sent in the verifier's initial step.

[6] We stress that in both cases both choices can be made. We note that, when analyzing the completeness condition, one may prefer to analyze the deviation of the individual sums (for each i).

A special case (analogous to one part of [BoTr]): We now allow R_w to depend on w, but restrict our attention to the natural case in which the reduction does not ask a query y with probability that exceeds $\Pr[F_n = y]$ by too much. Specifically, suppose that $\Pr[R_w = y] \le \mathrm{poly}(|y|) \cdot \Pr[F_n = y]$, for every y. In this case, we modify the foregoing protocol as follows.

> Here it makes no sense to compare the claimed value of $(1/qm) \cdot \sum_{i=1}^{q} \sum_{k=1}^{m} \log_2 |f^{-1}(y_i^{(k)})|$ against $\mathrm{Exp}[\log_2 |f^{-1}(F_n)|]$. Instead we should compare the former (claimed) average to $\mathrm{Exp}[\log_2 |f^{-1}(R_w)|]$. Thus, the verifier needs to obtain a good approximation to the latter value. This is done by generating many y_i's as before (i.e., $y_i = f(x_i)$ for uniformly selected $x_i \in \{0,1\}^n$) along with fewer but still many y_i's sampled from R_w, and sending all these y_i's (in random order) to the prover. Specifically, for $t \ge \max_{y \in \{0,1\}^*} \{\Pr[R_w = y]/\Pr[F_n = y]\}$, we generate t times more y_i's from F_n, and so each y_i received by the prover is at least as likely to come from F_n than from R_w.
>
> The prover will be asked to provide all $|f^{-1}(y_i)|$'s along with lower-bound proofs, and afterwards (i.e., only after committing to these $|f^{-1}(y_i)|$'s) the verifier will ask for upper-bound proofs for those y_i's generated via F_n (for which the verifier knows a secret and uniformly distributed $x_i \in f^{-1}(y_i)$).

Recall that the prover cannot significantly overstate the size of any $|f^{-1}(y_i)|$ (i.e., overstate it by more than an $\varepsilon = 1/\mathrm{poly}(n)$ factor). If the prover significantly understates the sizes of too many of the $|f^{-1}(y_i)|$'s, then it is likely to similarly understate also the sizes of many $|f^{-1}(y_i)|$'s such that y_i was generated by sampling F_n. But in this case, with overwhelmingly high probability, the prover will fail in at least one of the corresponding upper-bound proofs.

The general case (analogous to another part of [BoTr]): We now allow R_w to depend arbitrarily on w, without any restrictions whatsoever. For a threshold parameter t to be determined later, we say that a query y is t-heavy if $\Pr[R_w = y] > t \cdot \Pr[F_n = y]$. (In the *special case*, we assumed that there are no $\mathrm{poly}(n)$-heavy queries.) Observe that the probability that an element sampled according to F_n is t-heavy is at most $1/t$, and thus modifying an inverting oracle such that it answers t-heavy queries by $\perp$ affects the inverting probability of the oracle by at most $1/t$. Thus, for $t \ge 2$, if we answer t-heavy queries by $\perp$ (and answer other f-images with a preimage), then we emulate a legitimate inverting oracle (which inverts f with probability at least $1/2$) and the reduction R is still supposed to work well. Referring to y as t-light if it is not t-heavy, we note that t-light queries can be handled as in the foregoing *special case* (provided $t \le \mathrm{poly}(n)$), whereas t-heavy queries are accounted for by the previous discussion. The problem is to determine whether a query is t-heavy or t-light, and certainly we have no chance of doing so if many (reduction) queries are very close to the threshold (e.g., $\Pr[R_w = y] = (t \pm n^{-\log n}) \cdot \Pr[F_n = y]$ for all y's). Thus, as in [BoTr], we select the threshold at random (say, uniformly in the interval $[2,3]$). Next, we augment the foregoing protocol as follows.

- We ask the prover to provide for each query $y_i^{(k)}$, also the value of $\Pr[R_w = y_i^{(k)}]$, or equivalently the size of $\{r : R(w,r) = y_i^{(k)}\}$. In addition, we ask for lower-bound proofs of these sizes.
- Using lower and upper bound protocols (analogously to the *simple case*)[7], we get an estimate of $\mathrm{Exp}[\log_2 |\{r : R(w,r) = R_w\}|]$. We let the verifier check that this value is sufficiently close to the claimed value of $(1/qm) \cdot \sum_{i=1}^{q} \sum_{k=1}^{m} \log_2 |\{r : R(w,r) = y_i^{(k)}\}|$, thus preventing an understating of the size of almost all the sets $\{r : R(w,r) = y_i^{(k)}\}$.

  Hence, combining these two items, the verifier gets a good estimate of the size of $\{r : R(w,r) = y_i^{(k)}\}$ for all but few (i,k)'s. That is, the verifier can confirm that for almost all the (i,k)'s the claimed (by prover) size of $\{r : R(w,r) = y_i^{(k)}\}$ is approximately correct.

[7] In the simple case we got an estimate of $\mathrm{Exp}[\log_2 |f^{-1}(F_n)|]$, while relying on our ability to generate samples of F_n along with a uniformly distributed member of $f^{-1}(F_n)$. Here we rely on our ability to generate samples of R_w along with a uniformly distributed member of $\{r : R(w,r) = R_w\}$.

- Using the claimed (by the prover) values of $\Pr[R_w = y_i^{(k)}]$ and $\Pr[F_n = y_i^{(k)}]$, the verifier makes tentative decisions regarding which of the $y_i^{(k)}$'s is t-light.

  Note that for most (i,k), the prover's claim about $\Pr[R_w = y_i^{(k)}]$ is approximately correct, whereas the claim about $\Pr[F_n = y_i^{(k)}]$ can only be understated (by the lower-bound on $f^{-1}(y_i^{(k)})$).

  Using a protocol as in the *special case*, the verifier obtains an estimate of $\mathrm{Exp}[\log_2 |f^{-1}(R'_w)|]$, where R'_w denotes R_w conditioned on being t-light, and checks that this value is sufficiently close to the claimed average of $\log_2 |f^{-1}(y_i^{(k)})|$, taken only over t-light $y_i^{(k)}$'s. In addition, the verifier checks that the fraction of t-light $y_i^{(k)}$'s (among all $y_i^{(k)}$'s) approximates the probability that R_w is t-light.

  We note that estimating $\mathrm{Exp}[\log_2 |f^{-1}(R'_w)|]$ is done by generating y_i's as in the *special case*, but with $t \in [2,3]$ as determined above, and while asking for the value of both $\Pr[R_w = y_i]$ and $\Pr[F_n = y_i]$ for all y_i's, and afterwards requiring upper-bound proofs for one of these values depending on whether y_i was sampled from R_w or F_n. Needless to say, these values will serve as basis for determining whether each y_i is t-heavy or t-light, and will also yield an estimate of the probability that R_w is t-light.

Recall that the verifier accepts w if and only if all the foregoing checks (including the ones stated in the adaptive case) are satisfied.

Ignoring the small probability that we selected a bad threshold t as well as the small probability that we come across a query that is close to the threshold, we analyze the foregoing protocol as follows. We start by analyzing the queries y_i's used in the sub-protocol for estimating $\mathrm{Exp}[\log_2 |f^{-1}(R'_w)|]$. We first note that, due to the lower and upper bound proofs, for almost all queries y_i's generated by R_w, the sizes of $\{r : R(w,r) = y_i\}$ must be approximately correct. Next, employing a reasoning as in the *special case*, it follows that for almost all t-light queries y_i's we obtain correct estimates of the size of their f-image (i.e., we verify that the almost all the sizes claimed by the prover for the $|f^{-1}(y_i)|$'s are approximately correct). It follows that we correctly characterize almost all the t-light y_i's generated by R_w as such. As for (almost all) t-heavy queries y_i's generated by R_w, we may wrongly consider them t-light only if the prover has significantly overstated the size of their preimage, because we have a good estimate of $\{r : R(w,r) = y_i^{(k)}\}$ for (almost all) these y_i's. Recalling that an overstatement of $|f^{-1}(y_i^{(k)})|$ is detected with overwhelmingly high probability (by the lower-bound protocol), it follows that almost all t-heavy queries y_i's generated by R_w are correctly characterize as such. Thus, the characterization of almost all y_i's (generated by R_w) as t-light or t-heavy is correct, and so is the estimate of the probability that R_w is t-light. Recalling that for almost all the t-light y_i's generated by R_w we have a correct estimate of $|f^{-1}(y_i)|$, we conclude that the estimate of $\mathrm{Exp}[\log_2 |f^{-1}(R'_w)|]$ is approximately correct.

Next we employ parts of the foregoing reasoning to the $y_i^{(k)}$'s. Recalling that, for almost all queries $y_i^{(k)}$, we obtained correct estimates of the size of $\{r : R(w,r) = y_i^{(k)}\}$, and that $|f^{-1}(y_i^{(k)})|$ cannot be overstated, we conclude that we correctly characterize almost all t-heavy queries as such. The comparison to the estimated probability that R_w is t-light guarantees that the prover cannot claim too many t-light $y_i^{(k)}$'s as t-heavy, which implies that we have correctly characterize almost all $y_i^{(k)}$'s as t-light or t-heavy. Recalling that $|f^{-1}(y_i^{(k)})|$ can only be understated (due to the lower-bound proofs) and using the estimate of $\mathrm{Exp}[\log_2 |f^{-1}(R'_w)|]$ as an approximate lower-bound, it follows that the claims made regarding almost all the $|f^{-1}(y_i^{(k)})|$'s are approximately correct. Thus, as in the special case, the correctness of the reduction implies the completeness and soundness of the foregoing AM-protocol. A formal description of this result appears in Appendix C. (We remark that the description in the appendix differ from the above description on some technicalities.)

3 Discussion: interpretations of our negative results

Negative results of the type obtained in this work (as well as in [FeFo, BoTr]) can be interpreted in several ways: The straightforward view is that such results narrow down the means by which one can base one-way functions on $\mathcal{NP}$-hardness. Namely, under the assumption that $\mathcal{NP}$ is not contained in co$\mathcal{AM}$, our results show that (1) *non-adaptive* randomized reductions are not suitable for basing one-way functions on $\mathcal{NP}$-hardness, and (2) that one-way functions based on $\mathcal{NP}$-hardness can not be size verifiable (e.g., cannot be regular with an efficiently recognizable range).

Another interpretation is that these negative results are an indication that (worst-case) complexity assumptions regarding $\mathcal{NP}$ as a whole (i.e., $\mathcal{NP} \not\subseteq \mathcal{BPP}$) are not sufficient to base one-way functions on. But this does not rule out the possibility of basing one-way functions on the worst-case hardness of a subclass of $\mathcal{NP}$ (e.g., the conjecture that $\mathcal{NP} \cap \text{co}\mathcal{NP} \not\subseteq \mathcal{BPP}$). This is the case because our results (as previous ones) actually show that certain reductions of the (worst-case) decision problem of a set S to (average-case) inverting of f imply that $S \in \mathcal{AM} \cap \text{co}\mathcal{AM}$. But no contradiction is obtained if S belongs to $\mathcal{NP} \cap \text{co}\mathcal{NP}$ anyhow. Indeed, the decision problems related to lattices that are currently known to have worst-case to average-case reductions belong to $\mathcal{NP} \cap \text{co}\mathcal{NP}$ (cf. [Aj, MiRe] versus [AhRe]).

Yet another interpretation is that these negative results suggest that we should turn to a more relaxed notion of a reduction, which is uncommon in complexity theory and yet is applicable in the current context. We refer to "non black-box" reductions in which the reduction gets the code (of the program) of a potential probabilistic polynomial-time inverting algorithm (rather than black-box access to an arbitrary inverting oracle). The added power of such (security) reductions was demonstrated a few years ago by Barak [Ba01, Ba02].

Acknowledgments

Dana Moshkovitz is grateful to Muli Safra for supporting her visit to MIT, where this research has been initiated.

References

[AhRe] D. Aharonov and O. Regev. Lattice Problems in NP intersect coNP. In *45th IEEE Symposium on Foundations of Computer Science*, 2004.

[AiHa] W. Aiello and J. Hastad. Perfect Zero-Knowledge Languages can be Recognized in Two Rounds. In *28th IEEE Symposium on Foundations of Computer Science*, pages 439–448, 1987.

[Aj] M. Ajtai. Generating hard instances of lattice problems. In *28th ACM Symposium on the Theory of Computing*, pages 99–108, 1996.

[BaLa] L. Babai and S. Laplante. Stronger seperations for random-self-reducability, rounds, and advice. In *IEEE Conference on Computational Complexity 1999*, pages 98–104, 1999.

[Ba01] B. Barak. How to Go Beyond the Black-Box Simulation Barrier. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 106–115, 2001.

[Ba02] B. Barak. Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. In *43th IEEE Symposium on Foundations of Computer Science*, to appear, 2002.

[BCGL] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the Theory of Average Case Complexity. *Journal of Computer and System Science*, Vol. 44, No. 2, April 1992, pages 193–219.

[BoTr] A. Bogdanov and L. Trevisan. On worst-case to average-case reductions for NP problems. In *Proc. 44th IEEE Symposium on Foundations of Computer Science*, pages 308–317, 2003.

[Br] G. Brassard. Relativized Cryptography. In *20th IEEE Symposium on Foundations of Computer Science*, pages 383–391, 1979.

[DiIm] G. Di-Crescenzo and R. Impagliazzo. Security-preserving hardness-amplification for any regular one-way function In *31st ACM Symposium on the Theory of Computing*, pages 169–178, 1999.

[ESY] S. Even, A.L. Selman, and Y. Yacobi. The Complexity of Promise Problems with Applications to Public-Key Cryptography. *Inform. and Control*, Vol. 61, pages 159–173, 1984.

[FeFo] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22:994–1005, 1993. Extended Abstract appeared in Proc. of IEEE Structures'91.

[FFLS] J. Feigenbaum, L. Fortnow, C. Lund, and D. Spielman. The power of adaptiveness and additional queries in random-self-reductions. *Computational Complexity*, 4:158–174, 1994. First appeared in Proceedings of the 7th Annual IEEE Conference on Structure in Complexity Theory, 1992, pp. 338-346.

[Go] O. Goldreich. *Foundation of Cryptography – Basic Tools.* Cambridge University Press, 2001.

[GIL+] O. Goldreich, R. Impagliazzo, L.A. Levin, R. Venkatesan, and D. Zuckerman. Security Preserving Amplification of Hardness. In *31st IEEE Symposium on Foundations of Computer Science*, pages 318–326, 1990.

[GKL] O. Goldreich, H. Krawczyk and M. Luby. On the Existence of Pseudorandom Generators. *SIAM Journal on Computing*, Vol. 22-6, pages 1163–1175, 1993.

[GVW] O. Goldreich, S. Vadhan and A. Wigderson. On interactive proofs with a laconic provers. *Computational Complexity*, Vol. 11, pages 1–53, 2003. Extended abstract in *28th ICALP*, Springer, LNCS 2076, pages 334–345, 2001.

[GoSi] S. Goldwasser and M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems. *Advances in Computing Research: a research annual*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 73–90, 1989. Extended abstract in *18th STOC*, pages 59–68, 1986.

[HHK+] I. Haitner, O. Horvitz, J. Katz, C.Y. Koo, R. Morselli and R. Shaltiel. Reducing complexity assumptions for statistically-hiding commitment. To appear in *Eurocrypt*, 2005.

[HNOS] E. Hemaspaandra, A.V. Naik, M. Ogiwara, and A.L. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. *Journal of Computer and System Science*, Vol. 53 (2), pages 194–209, 1996.

[ImLe] R. Impagliazzo and L.A. Levin. No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random. In *31st IEEE Symposium on Foundations of Computer Science*, 1990, pages 812–821.

[KaTr] J. Katz and L. Trevisan. On The Efficiency Of Local Decoding Procedures For Error-Correcting Codes. In *32nd ACM Symposium on the Theory of Computing*, pages 80–86, 2000.

[MiRe] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th IEEE Symposium on Foundations of Computer Science*, pages 372–381, 2004.

Appendix

A Preliminaries

To simplify the presentation we assume, without loss of generality, that the reduction always makes exactly q distinct queries.[8] In addition, we assume (again without loss of generality), that the error probability of the reduction is exponentially small (because this can be achieved by standard amplification).

A.1 Verifying Size Estimates

Let us survey several fundamental protocols for verifying estimates on sizes of NP languages (more accurately, for any n, the size of the n'th slice of $L \subseteq \{0,1\}^*$, $L \cap \{0,1\}^n$, which, in short, we will refer to as $L \subseteq \{0,1\}^n$).

Lower Bounds The work of Goldwasser and Sipser, showing private-coins protocols are equivalent to public coins protocol, has presented a protocol for proving the size of L is at least some number s. The idea behind it is the following: pick a random hash function h mapping $\{0,1\}^n$ to a range Γ of size slightly smaller than s. If $|L| \geq s$, then, with high probability, any member of Γ will have a pre-image by h. If $|L|$ is significantly smaller than s, an arbitrary member of Γ is not likely to have a pre-image by h in L. The exact parameters are detailed in the following theorem.

Theorem 5 (Goldwasser-Sipser). *For any NP language $L \subseteq \{0,1\}^n$, any natural $1 \leq s \leq 2^n$, any estimate $0 < \rho < 1$, and any amplifying factor $u = \mathrm{poly}(n)$, there exists a constant-round private-coin protocol between a prover P and a verifier V such that:*

- *If $|L| \geq s$, then there exists a prover strategy P so that* $\Pr_r[V(r) = 1] \geq 1 - \frac{9}{\rho^2 u}$.
- *If $|L| \leq (1-\rho)s$, then for every prover strategy P',* $\Pr_r[V(r) = 1] \leq \frac{9}{\rho^2 u}$.

where the probability is taken over the random coin tosses r of the verifier.

We comment that stronger bounds (i.e., $u = \exp(\mathrm{poly}(n))$) are available via a related protocol of [GVW].

Upper Bounds It is not likely that protocols with constant number of rounds could verify that the size of an NP language L is at most some number s. Nevertheless, the work of Aiello and Hastad showed that if the verifier is able to sample uniformly a member x within the language, then the verifier can also upper-bound the size of the language. The idea is again to use hash functions: pick a random hash function h mapping $\{0,1\}^n$ to a range Γ of size slightly smaller than the claimed size, s. Given $h(x)$, let the prover guess a short list of candidates for x. If $|L| = s$, there exists such short list with high probability. On the other hand, if $|L|$ is much larger than s, there should many z's in L with $h(z) = h(x)$, and thus the prover has a not-very-high chance to output a list containing x.

This protocol uses significantly the fact it has a uniform x in L and the fact this x is *private* (*i.e.*, not known by the prover). By the work of Goldwasser and Sipser [GoSi], if the Aiello-Hastad protocol is plugged into an AM protocol in a setting where the specific NP-language L can indeed be sampled, the private sampling can be replaced by usage of public coins.

The exact parameters of the protocol are detailed in the following theorem:

[8] Repeated queries may be answered by the reduction itself (using the previously received answer), and dummy queries may be added in case they are less than q queries.

Theorem 6 (Aiello-Hastad). *For any NP language $L \subseteq \{0,1\}^n$, any natural $1 \le s \le 2^n$, any estimate $0 < \rho < 1$, and any amplifying factor u, there exists a constant-round private-coins protocol so that for a verifier $V(r,x)$ having a uniformly distributed $x \in L$,*

- *If $|L| \le s$, then there exists a prover strategy P, so that for every $x \in L$, $\Pr_r[V(r,x) = 1] \ge 1 - \frac{9}{\rho^2 u}$.*
- *If $|L| \ge (1+\rho)s$, then for every prover strategy P', $\Pr_r\left[\Pr_{x\in L}[V(r,x)=1] \le 1 - \frac{\rho}{6}\right] \ge 1 - \frac{9}{\rho^2 u}$.*

where the probability is taken over the random coin tosses r of the verifier, as well as over the x uniformly distributed within L.

Note that if L is much larger than s, the verifier may erroneously accept with a very high probability (roughly $1-\rho$). Nevertheless, it is sufficient for our needs. We will be interested in several NP-languages, $L_1, \ldots, L_k$, and our protocol will be of the following form:

1. P : sends alleged upper-bounds $s_1, \ldots, s_k$ for $L_1, \ldots, L_k$.
2. V, P : initiate the Aiello-Hastad protocol in parallel to verify $|L_i| \le (1+\rho)s_i$, $i = 1, \ldots, k$.

In this setting, as the set of *cheats*, $\{1 \le i \le k \mid |L_i| \ge (1+\rho)s_i\}$, is fixed in the first round, the probability it is large, but the verifier does not reject in the second round, is very small.

A.2 Sampling

Interestingly, given an estimate on the size of L, a probabilistic protocol with constant number of rounds is capable of sampling an almost-uniform member of L. This was already shown by Goldreich et al [GVW], and we slightly generalize the analysis so it handles the case in which we merely have an *estimate* on the size of L, $(1-\gamma)N \le |L| \le (1+\gamma)N$.

Let us give some intuition. We would like the verifier to pick a random index i in $1, 2, \ldots, N$ and ask the prover to provide x_i, the i'th member of L. This fails because the verifier can indeed verify that $x_i \in L$, but does not seem to have a way verifying x_i is indeed the i'th member of L. However, we can use *hash functions* to pull off a similar trick. The idea is to pick a random hash function, and ask the prover to provide an element whose hash is some pre-determined value. If the hash can be computed efficiently, then verification is resolved.

Diving deeper into the implementation, we consider a family of hash functions mapping $\{0,1\}^n$ into a range whose size is N *shrank* by some parameter. This way, when picking a hash function at random, each value of the hash is expected to have a short list of pre-images mapped to it. The prover can send this short list to the verifier, and let it choose a random pre-image within it.

We use $2d$-independent hash functions (for some d that depends on a confidence parameter δ), as to ensure the size of the short list is close to its expectation. We take $\epsilon \le \frac{1}{3}$ to be our deviation parameter.

Sampling Protocol:

1. V: sends hash function h picked uniformly and independently at random from a family of $2d$-wise independent functions $\{0,1\}^n \to [\tilde{N}]$, $\tilde{N} = \frac{\epsilon^3 N^d}{2d^2(1-\gamma)}$, $d = \lceil \log \frac{1}{\delta} \rceil$.
2. P: sends a list of elements $A \subseteq \{0,1\}^n$ [supposedly the pre-images of 0 by h].
3. V: rejects if any of the following does not hold:
   - $|A| \ge (1-\epsilon-\gamma)N/\tilde{N}$.
   - For every $x \in A$, $h(x) = 0$.
   - For every $x \in A$, $x \in L$.

4. V: picks x uniformly and independently at random from A.

The analysis is by bounding, for every $S \subseteq L$, the deviation of $\left|h^{-1}(0) \cap S\right|$ from its expectation. This (for $S = L$) implies that $\left|h^{-1}(0) \cap L\right|$ is large enough to prevent the verifier from rejecting. This also implies that a random element in $h^{-1}(0) \cap L$ is close to being uniform over L. Moreover, the elements the prover is required to provide constitute almost all $h^{-1}(0) \cap L$, hence a uniform x among them is still almost uniform in L. This argument is made formal in the following lemma:

Lemma 7 (adapted from [GVW]). *If $N(1-\gamma) \leq |L| \leq N(1+\gamma)$, for some $\gamma \leq \frac{1}{3} - \epsilon$, then*

1. *The verifier either rejects in step 3, or it picks an element $x_i \in L$ that is at least $(\delta + 3\epsilon + 3\gamma)$-statistically close to being uniform over L.*
2. *There exists a prover strategy (being truthful), so that with probability at least $1 - \delta$, the verifier does not reject in step 3.*

B Adaptive Reductions, Size-Verifiable Functions

In this section we complete the proof of Theorem 3. In fact, we will extend the proof to the case that the preiamge sizes can only be approximately verified, but we start with the cases of size-verifiable functions (as stated in Theorem 3).

B.1 Size Verifiable Functions

We generalize the description of Section 2.1 (which referred to "size computable functions") to the case of size-verifiable functions. Following is a detailed description of the resulting protocol.

Protocol for proving membership in $\overline{L}$ (for the case of size-verifiable functions f).
Parameters Setting for common input w: Let $n = |w|$. Let $\epsilon > 1/\text{poly}(n)$ be a deviation parameter, and $\delta > 1/\text{poly}(n)$ be the error probability. For each $s = 1, \ldots, n$, let $H_{n,\ell}$ be a family of $2t$-wise independent hash functions mapping $\{0,1\}^n$ to $\{0,1\}^\ell$, where $t \geq \max\left\{2n, \log \frac{4qm}{\delta}\right\}$. Denote $\Delta = \frac{2t^2}{\epsilon^3}$ and $\gamma = \frac{1}{2q}$; let $m \geq 2\sqrt{\frac{2q}{\delta}}\frac{\epsilon}{\gamma}$.

1. P, V: Run m times in parallel the "Emulation of the Reduction" sub-protocol:
   (a) V: Send randomness r, and nq hash functions. Specifically, select and send $h_{\ell,i} \in_R H_{n,\ell}$ for each query $i = 1, \ldots, q$ of the reduction, and each size $\ell = 1, \ldots, n$.
   (b) P: Send sizes $s_1, \ldots, s_q$, and sets of inverses $A_1, \ldots, A_q$; denote by a_i the minimal element in A_i (in lexicographic order).
   Supposedly, $s_i = \left|f^{-1}(y_i)\right|$ for $y_i = R(w, r, a_1, \ldots, a_{i-1})$, and A_i is a subset of the inverses of y_i. Specifically, if y_i has few inverses, *i.e.*, $s_i \leq 2\Delta$, then $A_i = f^{-1}(y_i)$, whereas, if y_i has many inverses, *i.e.*, $s_i > \Delta$, then $A_i = f^{-1}(y_i) \cap h_{\ell_i,i}(0)^{-1}$ for $\ell_i = \lfloor(\log \frac{\left|f^{-1}(y_i)\right|}{\Delta})\rfloor$.
   To streamline the following description, when $\ell_i \leq 0$, we artificially define h_{i,ℓ_i} such that $h^{-1}_{i,\ell_i}(0^{\ell_i}) \stackrel{\text{def}}{=} \{0,1\}^n$. In such a case, $f^{-1}(y_i) \cap h^{-1}_{k,\ell_i}(0^{\ell_i}) = f^{-1}(y_i)$, and so the above can be described succinctly by instructing the prover to send $A_i = f^{-1}(y_i) \cap h_{\ell_i,i}(0)^{-1}$.
   (c) P,V: Denote $y_i = R(w, r, a_1, \ldots, a_{i-1})$. Use the size-verification protocol to prove (in parallel) for each $i = 1, \ldots, q$, that $s_i = \left|f^{-1}(y_i)\right|$, with confidence $\delta' = \frac{\delta}{4qm}$; reject if any of the runs of the size-verification protocol rejects.

2. V: Accept iff all the conditions below hold:

   (a) In each run of the above sub-protocol:

      i. $\forall i,\ A_i \subseteq \{0,1\}^n$ (is non-empty) and has size less than $(1+\epsilon)2\Delta$. We denote by a_i the minimal element of A_i (in lexicographic order).

      ii. For every $i = 1, ..., q$ and every $a \in A_i$, it holds that $f(a) = R(w, r, a_1, ..., a_{i-1})$ and $h_{\ell_i,i}(a) = 0$ for $\ell_i = \lfloor (\log \frac{s_i}{\Delta}) \rfloor$.

      iii. It holds that $R(w, r, a_1, ..., a_q) = 0$.

   (b) On average (over all runs $k = 1, \ldots, m$ of the above sub-protocol), the size of the sets A_i is close to its expectation ($\mathbb{E}[|A_i|] = \frac{s_i}{2^{\ell_i}}$), namely, for all $i = 1, \ldots, q$,

$$\frac{1}{m}\sum_{k=1}^{m}\left(\left|A_i^{(k)}\right| \cdot \frac{2^{\ell_i}}{s_i}\right) > (1-\gamma)$$

   where $A_i^{(k)}$ denotes the set A_i the prover sent for the i-th query of run k.

Analysis of the Protocol We now prove the completeness and soundness of our protocol. For ease of notation, in the following we mark the communication in each run $k = 1, \ldots, m$ of the sub-protocol by a superscript '(k)', namely:

Notations: Denote by $r^{(k)}$ and $h_{\ell,1}^{(k)}, \ldots, h_{\ell,q}^{(k)}$ the randomness and hash functions sent by the verifier in run k. Denote by $s_1^{(k)}, \ldots, s_q^{(k)}$ and $A_1^{(k)}, \ldots, A_q^{(k)}$ the sizes and sets sent by the prover in run k, and let $a_i^{(k)}$ denote the minimal element in $A_i^{(k)}$. Let $\ell_i^{(k)} = \lfloor (\log \frac{\left|f^{-1}(y_i^{(k)})\right|}{\Delta}) \rfloor$ for $y_i^{(k)} = R(w, r^{(k)}, a_1^{(k)}, \ldots, a_{i-1}^{(k)})$. Last, when $\ell_i^{(k)} \leq 0$, we abuse the notation and define $h_{\ell_i^{(k)},i}^{(k)}(0)^{-1} \stackrel{def}{=} \{0,1\}^n$ (thus having $f^{-1}(y_i^{(k)}) \cap h_{\ell_i^{(k)},i}^{(k)}(0)^{-1} = f^{-1}(y_i^{(k)})$).

To show that our protocol is complete, we show that, for an honest prover and $w \in \overline{L}$, with high probability, both the reduction accepts and the statistical tests pass (namely, $\left|A_i^{(k)}\right|$ is close to its expectation, and its average deviation over all runs k is even more tightly concentrated). The reduction accepts w.p. at least $1 - err$ (the success probability of the reduction), because when the prover is honest, its answers are *independent* of the randomness used by the reductions (they are determined only by the hash functions[9]). The average size of $A_i^{(k)}$ is close to its expectation, by Chebyshev analysis relying on the fact that the size of each individual set A_i is concentrated close to its expectation with overwhelming probability. Thus, for $w \in \overline{L}$ and an honest prover, the verifier accepts, w.h.p..

To show that our protocol is sound, we prove that, for any prover's strategy, if $w \notin \overline{L}$, then the verifier rejects, w.h.p.. Specifically, we show that as long as $m = \mathrm{poly}(n)$ is large enough, if the average size of A_i's is close enough to its expectation so that the verifier does not reject on condition 2b, then there must be at least one run k of the sub-protocol, where the prover did not omit even a single element from any of the set $A_i^{(k)}$ $i = 1, \ldots, q$, namely:

$$\exists k \text{ s.t. } \forall i = 1, \ldots, q,\ A_i^{(k)} = f^{-1}(y_i^{(k)}) \cap h_{\ell_i,i}^{(k)}(0)^{-1}$$

On this run k, the answers of the prover are *independent* of the randomness used by the reductions (the answers are determined only by the hash functions), and hence the reduction returns the correct

[9]We remark that choosing the hash functions in advance, before the queries of the reduction are knows, should not obscures the fact that the hash functions are independent of the randomness of the reduction, and are chosen on random by the verifier.

output, w.p at least $1 - m \cdot err$, where err denotes the error probability of the reduction. (The factor of m in the error probability emanates from the prover's freedom to choose on which run $k \in 1, \ldots, m$ it is "honest".) Thus, for $w \notin \overline{L}$, the verifier rejects, w.h.p., even when the prover is dishonest. We remark that for larger m, not only that a single run is "good", but actually, most runs are good, and therefore, we could modify our protocol to check that *a random run* $k \in_R 1, \ldots, m$ outputs 0, instead of checking that *all runs* output 0. (This modified protocol is the one presented in the overview section.)

We now follow the above intuition to give the formal proof. We begin by proving the concentration results we use, and then turn to proving completeness and soundness.

Useful Concentration Results Denote by $T_i^{(k)}$ the sets $A_i^{(k)}$ that an honest prover sends (T stands for "Truth"), namely,

$$T_i^{(k)} = f^{-1}(y_i^{(k)}) \cap h^{(k)}_{\ell_i^{(k)},i}(0)^{-1}$$

Our proof relies on two properties of the the sets $T_i^{(k)}$. First, the size of the sets $T_i^{(k)}$ is concentrated close to their expectation. This is proven by high-moment analysis in [GVW]. Second, the average deviation of $\left|T_i^{(k)}\right|$ from its expectation is even more tightly concentrated around its expectation. This is proven using the Chebyshev inequality, while using the former property to bound the variance of this average. A formal statement of these properties follows.

Fact 8 (GVW). *For each i, k,* $\Pr\left[\left|\left|T_i^{(k)}\right| - \mathbb{E}[\left|T_i^{(k)}\right|]\right| \geq \epsilon\,\mathbb{E}[\left|T_i^{(k)}\right|]\right] \leq 2^{-t}$ *where the probability is taken over the random choice of* $h^{(k)}_{\ell_i^{(k)},i} \in H_{n,\ell_i^{(k)}}$.

Note that $\mathbb{E}[\left|T_i^{(k)}\right|] = \frac{\left|f^{-1}(y_i^{(k)})\right|}{2^{\ell_i^{(k)}}} \in (\Delta, 2\Delta)$, when $\ell_i^{(k)} > 0$, and $\mathbb{E}[\left|T_i^{(k)}\right|] = \left|f^{-1}(y_i^{(k)})\right|$, otherwise.

Lemma 9. *Let* $s_i^{(k)} = \left|f^{-1}(y_i^{(k)})\right|$ *for* $y_i^{(k)} = R(w, r^{(k)}, a_1^{(k)}, \ldots, a_{i-1}^{(k)})$, *namely, the prover is truthful when sending the set sizes. Then for the ϵ of our protocol and for any $\gamma > 0$,*

$$\Pr\left[\left|\frac{1}{m}\sum_{k=1}^{m}\left(\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right) - 1\right| > \gamma\right] \leq \frac{\delta}{4q}$$

where the probability is taken over the random choice of $h^{(k)}_{\ell_i^{(k)},i} \in H_{n,\ell_i^{(k)}}$.

Proof. The bound follows from Chebyshev inequality. In order to apply Chebyshev, we compute the expectation and variance of $\frac{1}{m}\sum_{k=1}^{m}\left(\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right)$. We first compute the expectation. By linearity of expectation, $\mathbb{E}\left[\frac{1}{m}\sum_{k=1}^{m}\left(\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right)\right] = \frac{1}{m}\sum_{k=1}^{m}\left(\mathbb{E}\left[\left|T_i^{(k)}\right|\right] \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right) = 1$ where the last equality holds since $\mathbb{E}\left[\left|T_i^{(k)}\right|\right] = \frac{s_i^{(k)})}{2^{\ell_i^{(k)}}}$. To bound the variance of $\frac{1}{m}\sum_{k=1}^{m}\left(\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right)$ we first bound the variance of $\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}$. Since $\left|T_i^{(k)}\right| \leq 2^n$ and $\mathbb{E}[\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}] = 1$, we have:

$$var\left[\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right] \leq \Pr\left[\left|\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}} - 1\right| \geq \epsilon\right] \cdot (2^n)^2 + \Pr\left[\left|\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}} - 1\right| < \epsilon\right] \cdot \epsilon^2 \leq 2\epsilon^2$$

where to reach the last inequality, we bound the first term by $2^{-t}2^{2n} \leq \epsilon^2$ (applying Fact 8 and the choice of $t \geq 2n$), and we (trivially) bound the second term by $1 \cdot \epsilon^2$. Now, since $H_{n,\ell_i^{(k)}}$ are $2t$-wise independent (and, in particular, pairwise independent), we get: $var\left[\frac{1}{m}\sum_{k=1}^{m}\left(\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right)\right] = \frac{1}{m^2}\sum_{k=1}^{m} var\left[\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right] \leq 2\left(\frac{\epsilon}{m}\right)^2$. Last, applying Chebyshev inequality, the lemma is obtained: $\Pr\left[\left|\frac{1}{m}\sum_{k=1}^{m}\left(\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right) - 1\right| > \gamma\right] \leq \frac{1}{\gamma^2} var[\frac{1}{m}\sum_{k=1}^{m}\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}] \leq 2\left(\frac{\epsilon}{\gamma m}\right)^2 = \frac{\delta}{4q}$ where the last equality is obtained by assigning parameters values as set in our protocol. $\square$

Completeness and Soundness We now prove the completeness and soundness of our protocol.

Theorem 10 (Completeness). *Let $w \in \overline{L}$, then when the prover is honest, the verifier accepts w.p. at least $1 - \delta$*

Proof. We now go over the conditions of the protocol and prove that they simultaneously hold, w.p. at least $1 - \delta$. By the properties of the size-verification protocols, the verifier accepts on step 1c of the sub-protocol simultaneously for all i, k, w.p. at least $1 - \delta$.

The size verification protocol simultaneously accepts for all i, k w.p. at least $(1 - \delta')^{qm} \geq 1 - \frac{\delta}{4}$ for $\delta' = \frac{\delta}{4qm}$ as in the protocol.

We next show that condition 2a holds, w.p. at least $1 - \frac{\delta}{2}$. Condition 2(a)i on the size of the $A_i^{(k)}$'s holds, w.p. at least $(1 - \frac{\delta}{4mq})^q \geq 1 - \frac{\delta}{4m}$ (because it holds for each individual index i, w.p. at least $1 - 2^{-t} = 1 - \frac{\delta}{4qm}$, by Fact 8 and the choice of $t \geq \log\frac{4qm}{\delta}$). In the following we condition on the fact that 2(a)i holds. Note that, in this case, the inverses $a_i^{(k)}$'s are well defined $a_i^{(k)} = \min\left\{A_i^{(k)}\right\}$. Moreover, $a_1^{(k)}, \ldots, a_q^{(k)}$ are independent of the randomness $r^{(k)}$ of the reduction (recall that the prover is honest and hence $A_i^{(k)} = T_i^{(k)}$). Condition 2(a)ii on the correctness of running the emulation trivially holds (for honest provers). Last, since $a_1^{(k)}, \ldots, a_q^{(k)}$ are independent of the randomness $r^{(k)}$ of the reduction, then condition 2(a)iii holds w.p. at least $1 - err \geq 1 - \frac{\delta}{4m}$ (where err denotes the error probability of R on the worst possible fixed oracle, and the inequality is true since, we assumed w.l.o.g $err = exp(-n)$). Put together we get that conditions 2(a)i-2(a)iii simultaneously hold in all m runs sub-protocol, w.p. at least $\left((1 - \frac{\delta}{4m})(1 - \frac{\delta}{4m})\right)^m \geq 1 - \frac{\delta}{2}$.

Last, Lemma 9 implies that condition 2b simultaneously holds for all $i = 1, \ldots, q$, w.p. at least $\left(1 - 2\left(\frac{\epsilon}{\gamma m}\right)^2\right)^q \geq (1 - \frac{\delta}{4q})^q \geq 1 - \frac{\delta}{4}$ (for $m \geq 2\sqrt{\frac{2q}{\delta}}\frac{\epsilon}{\gamma}$ as in the protocol).

Put together, we get that the verifier accepts, w.p. at least $(1 - \frac{\delta}{4})(1 - \frac{\delta}{4})(1 - \frac{\delta}{2}) \geq 1 - \delta$. $\square$

Theorem 11 (Soundness). *Let $w \notin \overline{L}$, then for any prover strategy, the verifier rejects w.p. at least $1 - \delta$.*

Proof. In the following we assume w.l.o.g that $s_i^{(k)} = \left|f^{-1}(y_i^{(k)})\right|$ $\forall i, k$ (otherwise the verifier rejects on the size verification protocol rejects, w.p. $1 - \delta' \geq 1 - \delta$, and the proof is completed). Note that this implies $\ell_i^{(k)} = \lfloor(\log\frac{\left|f^{-1}(y_i^{(k)})\right|}{\Delta})\rfloor$ $\forall i, k$. Also, assume w.l.o.g that condition 2(a)ii of the sub-protocol always holds, namely, the prover follows the syntactic instructions of the protocol (otherwise, the verifier rejects and the proof is completed).

In the lemma below we show that if the verifier does no reject on condition 2b of the protocol, then there is a run k of the sub-protocol where the prover does not omit any inverse, namely, $A_i^{(k)} = T_i^{(k)}$ $\forall i$, w.p. at least $1 - \frac{\delta}{4}$. This concludes the proof, since, in this case, the inverses $a_i^{(k)}$ used in the reduction in this k-th run are *independent* of its randomness $r^{(k)}$, and hence the reduction outputs 1 w.p. at least $1 - m \cdot err$ (where err denotes the error probability of R on the worst possible fixed oracle). Thus the verifier rejects, w.p. at least $1 - (\frac{\delta}{4} + m \cdot err) \geq 1 - \delta$. (Note, that the error probability of the reduction might be $m \cdot err$ –in contrast to err– because the prover might choose to be truthful on the the worst possible run $k \in 1, \dots, m$.)

Lemma 11.1. *Assume* $s_i^{(k)} = \left|f^{-1}(y_i^{(k)})\right|$ $\forall i, k$ *and condition 2(a)ii holds in all runs of the sub-protocol. Then, if the verifier does not reject on condition 2b of the protocol, then there exists a run* k *of the sub-protocol where* $A_i^{(k)} = T_i^{(k)}$ *simultaneously hold for all* $i = 1, \dots, q$*, w.p. at least* $1 - \frac{\delta}{4}$.

Proof. To prove the lemma assume the contrary, namely, for each run $k = 1, \dots, m$ there is an index i s.t. $\left|A_i^{(k)}\right| \leq \left|T_i^{(k)}\right| - 1$. By the pigeon hole principle, this implies there exists an index i_0 s.t. $\left|A_{i_0}^{(k)}\right| \leq \left|T_{i_0}^{(k)}\right| - 1$ at least $\frac{m}{q}$ times. Therefore,

$$\frac{1}{m}\sum_{k=1}^{m}\left|A_{i_0}^{(k)}\right| \cdot \frac{2^{\ell_{i_0}^{(k)}}}{s_{i_0}^{(k)}} \leq \left(\frac{1}{m}\sum_{k=1}^{m}\left|T_{i_0}^{(k)}\right| \cdot \frac{2^{\ell_{i_0}^{(k)}}}{s_{i_0}^{(k)}}\right) - \frac{1}{q}$$

Now, by Lemma 9, $\frac{1}{m}\sum_{k=1}^{m}\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}} \leq 1 + \gamma$ holds simultaneously for all $i = 1, \dots, q$ (and thus, in particular, applies to i_0), w.p. at least $1 - \frac{\delta}{4}$. In this case,

$$\frac{1}{m}\sum_{k=1}^{m}\left|A_{i_0}^{(k)}\right| \cdot \frac{2^{\ell_{i_0}^{(k)}}}{s_{i_0}^{(k)}} \leq (1+\gamma) - \frac{1}{q} = 1 - \gamma$$

(where the equality is derived by assigning $\gamma = \frac{1}{2q}$ as in our protocol). Thus the verifier rejects on condition 2b, w.p. at least $1 - \frac{\delta}{4}$. ■ □

B.2 Extension to Approximate-Size Verifiable Functions

So far we handles functions where the *exact* number of pre-images was efficiently verifiable (*i.e.*, size verifiable functions). A natural extension is to functions where only an *approximate* value for the number of pre-images can be efficiently verified. We call such functions *approximate-size verifiable*. We show that there there exists no reduction (not even an adaptive one) from deciding an NP-complete language to inverting a polynomial-time computable function which approximate-size verifiable, (unless co$\mathcal{NP} \subseteq \mathcal{AM}$).

An interesting applications of the extension to approximate-size verifiable functions is showing that one cannot base on NP the existence of OWFs that are regular and range recognizable (possibly via an AM-protocol), *even when the regularity parameter of the function is unknown.* To prove this we show that such functions are approximate-size verifiable. We comment that the approximate-size verification protocol that we present relies on the ability to sample in the range of f. This is another demonstration of the benefit in directly addressing OWFs rather than reducing the decision problems.

Definition 12 (Approximate-Size Verifiable)**.** *We say that a function* $f\colon \{0,1\}^* \to \{0,1\}^*$ *is* approximate-size verifiable *if there is a constant-round proof system that, given common inputs* $y \in \{0,1\}^*$ *and* $N \in \mathbb{R}$*, an approximation parameter* ρ *and confidence parameter* δ*, w.p. at least* $1 - \delta$*, the verifier accepts iff* $\left|f^{-1}(y)\right| \in (1 \pm \rho)N$.

With a slight adaptation of our protocol and analysis, our result also hold to approximate-size verifiable functions.

Theorem 13. *Unless* co$\mathcal{NP} \subseteq \mathcal{AM}$*, there exists no reduction* (even not an adaptive one) *from deciding an NP-complete language to inverting a polynomial-time computable function f which is* approximate-size verifiable*.*

To prove the theorem we assume the contra-positive (namely, that there exists a polynomial-time reduction (possibly adaptive) from deciding L to inverting f), and present an AM protocol for deciding $\overline{L}$; thus proving co$\mathcal{NP} \subseteq \mathcal{AM}$. The AM-protocol for $\overline{L}$ as well as its analysis are similar to the the former case when f was *exactly* size-verifiable. Let us highlight the differences.

Highlighting the differences in the protocol. The AM-protocol for $\overline{L}$ is an adaptation of our former protocol that we gave for *exact* size-verifiable functions f. Let us highlight the differences between the two protocols. We change the parameter γ to be $\gamma = \frac{1}{4q}$, and set the approximation parameter ρ to be $\rho = \gamma = \frac{1}{4q}$. Note that $\rho + \gamma = \frac{1}{2q}$ which is the value of γ in the former protocol. In step 1c we run the approximate-size verification protocol (instead of the exact one) to prove for each i, that $\left|f^{-1}(y_i)\right| \in (1 \pm \rho)s_i$, with the same confidence $\delta' = \frac{\delta}{4qm}$ as in the former protocol. In step 2(a)i we check the A_i's (are non-empty and) have size less than $(1+\rho)\cdot(1+\epsilon)2\Delta$. Last, in step 2b we check that $\frac{1}{m}\sum_{k=1}^{m}\left(\left|A_i^{(k)}\right| \cdot \frac{2^{\ell_i}}{s_i}\right) > 1 - (\gamma + \rho)$.

Highlighting the main changes in the analysis. Let $T_i^{(k)} = f^{-1}(y_i^{(k)}) \cap h^{(k)}_{\ell_i^{(k)},i}(0)^{-1}$ (as before). Our bound on $\mathbb{E}[\left|T_i^{(k)}\right|]$ is now slightly weaker:

$$\mathbb{E}[\left|T_i^{(k)}\right|] = \frac{\left|f^{-1}(y_i^{(k)})\right|}{2^{\ell_i^{(k)}}} \in \frac{(1 \pm \rho)s_i^{(k)}}{2^{\ell_i^{(k)}}} \in ((1-\rho)\Delta, (1+\rho)\Delta)$$

This is the reason for the additional $1+\rho$ factor on the condition of the size of A_i in step 2(a)i. Similarly, we now only know that $\mathbb{E}[\left|T_i^{(k)}\right| \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}] \in 1 \pm \rho$, and therefore, Lemma 9 is adapted to be:

Adaptation of Lemma 9: *Let* $\left|f^{-1}(y_i^{(k)})\right| \in (1 \pm \rho)s_i^{(k)}$ *for* $y_i^{(k)} = R(w, r^{(k)}, a_1^{(k)}, \ldots, a_{i-1}^{(k)})$. *Then for the ϵ of our protocol and for any $\gamma > 0$,*

$$\Pr\left[\left|\frac{1}{m}\sum_{k=1}^{m}\left(\left|T_i^{(k)}\right| \cdot \frac{2^{\ell_i^{(k)}}}{s_i^{(k)}}\right) - 1\right| > (\gamma + \rho)\right] \leq \frac{\delta}{4q}$$

where the probability is taken over the random choice of $h^{(k)}_{\ell_i^{(k)},i} \in H_{n,\ell_i^{(k)}}$.

This is the reason for the change is step 2b. Last, for Lemma 11.1 to hold, we need $\rho + \gamma = \frac{1}{2q}$. This explains our choice of parameters. The complete analysis is deferred to the full version of this paper.

Application to Regular Functions with Unknown Regularity Parameter A special case of size-verifiable functions are regular functions with efficiently recognizable range (possibly via an AM-protocol). Recall, that loosely speaking, for regular functions the number of pre-image is efficiently determined by the input length. We show that our negative results also hold for functions where –though the number of pre-image is determined by the length– it *cannot be computed efficiently*. Specifically, we show that such functions are *approximate-size verifiable* as long as their range is efficiently recognizable (possibly via an AM-protocol). This implies that such functions cannot be proved one-way based on $\mathcal{NP}$.

Let us sketch an approximate-size verification protocol for such functions. Since the number of pre-images is equal over all y's, it suffices to estimate $\left|f^{-1}(y)\right|$ on any arbitrary y. So, the verifier may choose a random x and send $y = f(x)$, for which the prover claims and proves the size of $\left|f^{-1}(y)\right|$ by utilizing the lower and upper bound protocols of [GoSi, AiHa] respectively. (Note that when using the upper bound protocol, the verifier relies on the knowledge of the pre-image x of y, namely, on the ability to efficiently sample pairs $x, f(x)$.)

C General Functions, Non-Adaptive Reductions

In this section we complete the proof of Theorem 4. We remark that the presentation in this section is somewhat different from the presentation in Section 2.2. The difference relates to the specific way in which it is verified that for almost all (i,k)'s the set sizes claimed for $y_i^{(k)}$'s (i.e., both $\{r : R(w,r) = y_i^{(k)}\}$ and $|f^{-1}(y_i^{(k)})|$) are not understated (by more than a $(1-\varepsilon)$ factor). Both in Section 2.2 and here, a significant understatement is detected by comparing the claimed values to corresponding statistics gathered for independently selected y_i (for which we can approximately verify thee set sizes). The difference is in how exactly the comparison is done. In Section 2.2 we considered the average of the logarithm of the set sizes (where multiplicative factors in set sizes get translated to additive error terms). In the current section, we "group" the $y_i^{(k)}$'s (resp., the y_i's) according to the set sizes and compare the relative fraction of elements in each group (i.e., compare the weight of the j-th group of $y_i^{(k)}$'s to the j-th group of y_i's, for every j). Furthermore, the alternative procedure employed here requires the selection of randomized thresholds for the sizes in each group.

Let us illustrate the situation as follows. Suppose that Alice got samples $s_1, ..., s_t$ from some distribution X over $[N]$ and she is supposed to send these to Bob. Further suppose that Alice can only understate the value of the samples she got (i.e., send $c_1, ..., c_t$ such that $c_i \le s_i$) and that Bob can sample X by himself. We want to prevent Alice from sending Bob too many c_i's that are significantly smaller than the corresponding s_i's. In Section 2.2, we achieve this goal by letting Bob estimates $E[\log_2 X]$ and having him compare this estimate to $(1/t)\sum_{i=1}^{t} \log_2 c_i$. Clearly, if Alice understates α fraction of the c_i's by more than than a $1-\beta$ factor then the calculated average is highly likely to be $\alpha\beta - (\log_2 N)/t^{1/3}$ smaller than Bob's estimate of $E[\log_2 X]$. In this section, for every $j = 1, ..., N/\varepsilon$, we let Bob compare $\Pr[X > \rho \cdot (1+\varepsilon)^j]$ to $|\{i : c_i > \rho \cdot (1+\varepsilon)^j\}|/t$, where $\rho \in (0.9, 1)$ is uniformly chosen (such that the probability that X is too close to some threshold value $\rho \cdot (1+\varepsilon)^j$ is very small). Indeed, also here, if Alice understates α fraction of the c_i's by more than than a $1-\varepsilon^2$ factor then some of the values $|\{i : c_i > \rho \cdot (1+\varepsilon)^j\}|/t$ will be too small when compared to $\Pr[X > \rho \cdot (1+\varepsilon)^j]$.

C.1 Fixing of Parameters

Let us assume there exists a *non-adaptive* randomized worst-case to average-case reduction R from some NP language L to Δ inverting an efficiently computable family $\{f_n\}$.

For some constant $0 < \delta < 1$, let us construct an AM protocol, that given $x \in \{0,1\}^n$ decides whether $x \in \bar{L}$ with probability at least $1 - 2\delta$. We assume w.l.o.g that there exists an efficient reduction from deciding L to inverting f that uses n' random bits to perform q queries identically distributed according to some efficiently samplable distribution $S(U)$, and has error probability η small enough to satisfy $\eta \le \frac{\delta}{10q}$.

Define ε to be some parameter small enough with respect to η, $\varepsilon \leq \frac{\eta}{20}$. Our AM-protocol for $\overline{L}$ is composed of two phases. In the first phase, *the emulation phase*, the reduction R is simulated by the verifier $m \stackrel{def}{=} \left\lceil \frac{10^6 n'^3 q^3}{\varepsilon^2 \log^3(1+\varepsilon)} \right\rceil$ times independently in parallel, using the prover to simulate an f-inverting oracle. This phase is guaranteed to work well under an assumption, which is, roughly, that the prover's answers are ε-approximations for the true answers. The purpose of the second phase, *the verification phase*, is to ensure this assumption indeed holds, if the verifier does not reject, with high probability.

C.2 Phase I - Emulation

C.2.1 On The Distribution of Queries

Our first observation is that we can focus on oracles that invert only queries that are likely to be asked according to $f(U)$.

Formally, we say y is *ok* if the ratio between its probability according to the reduction, *i.e.*, according to $S(U)$, and its probability according to $f(U)$, is small. For every positive number n, we define $\frac{n}{0} = \infty$, $n \cdot \infty = \infty$, $n < \infty$, as to handle the case where a query of the reduction has no pre-images at all.

Definition 14 (relative weight). *The* relative weight *of a query y in the reduction is*

$$\hat{w}(y) \stackrel{def}{=} \frac{\left|S^{-1}(y)\right| / 2^{n'}}{\left|f^{-1}(y)\right| / 2^{n}}$$

We say y is t-ok (or simply ok , when t is clear from context), if $\hat{w}(y) \leq t$.

If t is large enough, we can focus exclusively on oracles that invert only t-ok queries. Let $\mathbb{O}^t$ denote the set of all f-inverting oracles O that satisfy the following: if y is t-ok then $O(y) \in f^{-1}(y)$, and, otherwise, $O(y) = \perp$.

Proposition 15. *Let $t \geq \frac{1}{1-\Delta}$. For every $O \in \mathbb{O}^t$,*

$$\Pr_r \left[R^O(x,r) \neq L(x)\right] \leq \eta$$

Proof.

$$\mathbf{E}_{y \sim f(U)}\left[\hat{w}(y)\right] = \sum_y \frac{\left|f^{-1}(y)\right|}{2^n} \cdot \frac{\left|S^{-1}(y)\right| / 2^{n'}}{\left|f^{-1}(y)\right| / 2^n} = \sum_y \frac{\left|S^{-1}(y)\right|}{2^{n'}} = 1$$

By Markov inequality,

$$\Pr_{y \sim f(U)}\left[\hat{w}(y) \geq t\right] \leq \frac{1}{t}$$

That is, $\Pr_{x \in U}\left[f(x) \text{ is } t\text{-ok }\right] \geq 1 - \frac{1}{t} \geq \Delta$. That is, O is a Δ f-inverting oracle. The proposition follows from the definition of the reduction. ■

Description of The Emulation Phase

Fix the input $x \in \{0,1\}^n$, on which the AM protocol should decide whether $x \in \overline{L}$. The verifier first picks at random $t^{ok} \geq \frac{1}{1-\Delta}$. The reason for the randomized choice is that this way, with high probability, if a query is picked according to $S(U)$, then it is not likely to have a relative weight which is close to the threshold. This issue will be addressed in the verification phase.

Then, the verifier picks independently uniformly distributed randomness strings $r^1, \ldots, r^m$ for the reduction, and computes the queries that the reduction would have generated for the f-inverting oracle on those randomness strings. The prover is required to provide the following information, Info(y), regarding each query y:

- n_y – supposedly the number of pre-images of y by f.
- r_y – supposedly the number of randomness strings on which S generates query y.

Given this information, the verifier can classify which of the queries are ok , as well as generate an almost-random pre-image for each of the queries via the sampling protocol. This allows the verifier to define a *partial oracle*. For input $x \in \{0,1\}^n$ and randomness $r \in \{0,1\}^{n'}$, the behavior of the reduction is determined solely by the oracle's answers to $y_1, \ldots, y_q$, the queries generated for x and r. Let $\mathbb{O}^t_{x,r}$ denote the set of such partial oracles, assigning each y_i either some $x_i \in f^{-1}(y_i)$ if y_i is ok , or $\perp$, otherwise. The verifier can now simulate the reduction to decide whether $x \in \overline{L}$.

Formally, the protocol is as follows:

Phase I - Emulating the Reduction:

1. V: picks uniformly at random $\ell_{coin} \in \{1, \ldots, \ell_{max}\}$ for $\ell^{max} = \frac{100}{\delta^2 qm}$, and sets

$$t^{ok} = \left(1 + \frac{\varepsilon}{\ell^{ok}}\right) \cdot \frac{1}{1-\Delta}$$

2. V: picks independently at random $r^k \in \{0,1\}^{n'}$ for $k = 1, \ldots, m$.
3. V: computes $y_1^k, \ldots, y_q^k$, the queries of R on randomness r^k, $k = 1, \ldots, m$.
4. P: sends Info(y_1^k),…,Info(y_q^k), $k = 1, \ldots, m$.
5. V,P: initiate in parallel the sampling protocol with confidence parameter ε and deviation parameter ε to produce x_j^k for $j = 1, \ldots, q$, $k = 1, \ldots, m$; use $n_{y_j^k}$ as the size parameter.
6. V: defines the following partial oracle, $k = 1, \ldots, m$:

$$\forall 1 \leq j \leq q \quad O_k(y_j^k) \stackrel{def}{=} \begin{cases} x_j^k & \frac{r_{y_j^k}/2^{n'}}{n_{y_j^k}/2^n} \leq t^{ok} \\ \\ \perp & otherwise \end{cases}$$

7. V: rejects if there exists $1 \leq k \leq m$, for which the reduction R on randomness r^k and oracle O_k returns "$x \in L$".

Analysis of The Emulation Phase

Hitting Rates Our arguments crucially rely on the well known fact that whenever one samples enough elements from some distribution, for every property, the fraction of elements that satisfy this property in the sample, is almost the same as the probability this property is satisfied according to the distribution.

Lemma 16 (Chernoff). *Fix some sample space Y, and some subset $Y' \subseteq Y$. For every $\delta > 0$, if $y_1, \ldots, y_m$ were drawn independently at random from a distribution D over Y, then with probability at least $1-\delta$,*

$$\left| \Pr_{k \in_R [m]} \left[y_k \in Y' \right] - \Pr_{y \sim D} \left[y \in Y' \right] \right| \leq \frac{\lambda}{\sqrt{m}} \cdot \Pr_{y \sim D} \left[y \in Y' \right]$$

for $\lambda = 3\sqrt{\ln \frac{1}{\delta}}$.

Proof. Fix a threshold i. For every $1 \leq k \leq m$, let X_k be an indicator variable for the event $y_k \in Y'$. Clearly, for any $1 \leq k \leq m$, $\mathbf{E}[X_k] = \Pr_{y \sim D}[y \in Y']$. Let $X = \sum_{k=1}^{m} X_k$. In other words, $X = m \cdot \Pr_{k \in_R [m]}[y_k \in Y']$. By linearity of expectations, $\mathbf{E}[X] = m \cdot \Pr_{y \sim D}[y \in Y']$. Moreover, using the independence between the $0-1$ variables, $\mathbf{Var}[X] \leq \mathbf{E}[X]$. Applying Chernoff's bound, we have

$$\Pr\left[|X - \mathbf{E}[X]| > \lambda\sqrt{\mathbf{E}[X]}\right] \leq 2e^{-\lambda^2/4} \leq \delta$$

■

We show that if among the m emulations of the reduction, in some trial,

1. The classification whether the queries are ok is right.
2. For ok queries, the number of f- pre-images claimed by the prover ε-approximates the true number.

Then, the protocol operates correctly on the input.

The proof is done in three steps. The first step is to establish that for the randomness strings the verifier samples throughout the protocol, the set of oracles yielding to an erroneous outcome of the reduction, *i.e.*, $R^O(x,r) \neq L(x)$, is small, with high probability. The next step is to discuss almost-uniform responses to queries, and bound the bias bad oracles might benefit from. Next, we use the analysis of the sampling protocol as well as the assumption to deduce the completeness and soundness of the protocol.

Theorem 17 (correctness of emulation phase). *For every $x \in \{0,1\}^n$, if there exists a trial $1 \leq k \leq m$ in which,*

1. *for every $1 \leq j \leq q$, $2^n r_{y_j^k}/2^{n'} n_{y_j^k} \leq t^{ok}$ if and only if $\hat{w}(y_j^k) \leq t^{ok}$* (ok classification).
2. *for every $1 \leq j \leq q$, if $\hat{w}(y_j^k) \leq t^{ok}$, then $n_{y_j^k}(1-\varepsilon) \leq \left|f^{-1}(y_j^k)\right| \leq n_{y_j^k}(1+\varepsilon)$* (pre-images approximation).

Then the protocol operates correctly on input x, that is:

1. *If $x \in \overline{L}$, then there exists a prover strategy that makes the verifier accept with probability at least $1-\delta$* (completeness).
2. *If $x \notin \overline{L}$, then for any prover strategy, the verifier accepts with probability at most δ* (soundness).

Proof. For randomness r, consider the set of bad oracles for r, $B_r = \left\{ O \in \mathbb{O}_{x,r}^{t^{ok}} \;\middle|\; R^O(x,r) \neq L(x) \right\}$. By proposition 15, for every f-inverting oracle $O \in \mathbb{O}^{t^{ok}}$, $\Pr_r[O \in B_r] \leq \eta$. Hence, $\mathbf{E}_r[|B_r|] \leq \eta \left|\mathbb{O}_{x,r}^{t^{ok}}\right|$. By Markov inequality, for every $c > 0$, and, in particular, for $c = 2$,

$$\Pr_r[|B_r| \geq c\mathbf{E}_r[|B_r|]] \leq \frac{1}{c}$$

Therefore, by lemma 16, for all r^k's sampled by the verifier, B_{r^k} is small, with high probability, *i.e.*,

$$\Pr\left[\bigwedge_{k=1}^{m} |B_{r^k}| \leq c\eta \left|\mathbb{O}_{x,r^k}^{t^{ok}}\right|\right] \geq \Pr\left[\left|\Pr_{k \in_R [m]}\left[|B_{r^k}| \geq c\eta \left|\mathbb{O}_{x,r^k}^{t^{ok}}\right|\right] - \Pr_r\left[B_r \geq c\eta \left|\mathbb{O}_{x,r}^{t^{ok}}\right|\right]\right| > \frac{1}{c}\right] \geq 1 - \frac{\delta}{4}$$

Let us condition on this event in the rest of the analysis (We will later use a union-bound to get rid of this conditioning).

Assume some $k \in [m]$ is such that for every $j \in J \stackrel{def}{=} \left\{1 \le j \le q \mid y_j^k \text{ is ok}\right\}$, x_j^k is ϵ-almost uniform within $f^{-1}(y_j^k)$. Let us bound the probability the partial oracle O assigning x_j^k to all ok queries $j \in J$, and $\perp$ to every other query, is in B_{r^k}.

Denote $J = \{j_1, \ldots, j_t\}$. For $i \in [t]$, and a set of responses $a_1, \ldots, a_{i-1}$ to queries $y_{j_1}^k, \ldots, y_{j_{i-1}}^k$, define the good responses to the j_i'th query as those that preserve small expectation (over the remaining responses) to land in B_{r^k}, *i.e.*,

$$G_i \stackrel{def}{=} \left\{a_i \in f^{-1}(y_{j_i}^k) \,\middle|\, \mathbf{E}_{a_{i+1},\ldots,a_t}\left[\text{ the oracle defined by } a_1, \ldots, a_t \text{ is in } B_{r^k}\right] \le c\eta\right\}$$

By the ϵ-almost uniformity,

$$\Pr\left[x_{j_i}^k \in G_i \mid \bigwedge_{i'=1}^{i-1} x_{j_{i'}}^k \in G_{i'}\right] \ge 1 - c\eta - \epsilon$$

By the chain rule and our conditioning, the probability the induced partial oracle is not in B_{r^k} is at least $(1 - c\eta - \epsilon)^t$, which is, by Bernoulli inequality, at least $1 - t(c\eta - \epsilon)$. Now, let us use this discussion to prove the completeness and soundness of the protocol.

If $x \in \overline{L}$, consider an honest prover, *i.e.*, one that supplies true information for each of the y_j^k's. Hence, for every $1 \le k \le m$, $O_k \in \mathbb{O}_{x,r^k}^{t^{ok}}$. Moreover, by lemma 7, for every $1 \le k \le m$, with probability at least $(1 - \varepsilon)^q \ge 1 - \frac{\delta}{8}$, the verifier does not reject during the q applications (with independent randomness) of the sampling protocol. By the above discussion, the probability that, in every trial $1 \le k \le m$, $O_k \in B_{r^k}$, is at most $\frac{\delta}{2}$. Applying union bound, we get that the probability the verifier rejects is at most δ.

If $x \notin \overline{L}$, let k be the instantiation whose existence is ensured by the premise. Assume the verifier does not reject. Thus, $O_k \in \mathbb{O}_{x,r^k}^{t^{ok}}$. By the above discussion, the probability that $O_k \in B_{r^k}$ is at most $\frac{\delta}{2}$. Applying union bound, we get the probability the verifier accepts is at most δ. ∎

C.3 Phase II – Verification

Description of The Verification Phase

The purpose of the verification phase is to ensure the assumption required for the emulation phase indeed holds, with high probability. The assumption is the existence of a trial $1 \le k \le m$ in which the verifier has reliable information regarding all reduction's queries $y_1^k, \ldots, y_q^k$, that is, the classification of ok queries is correct and, for all ok queries, the estimate of the number of pre-images is accurate. To this end, the verifier also ensures r_y is a good estimate for the weight of a query y according to the reduction's distribution, for most queries y generated in the emulation phase.

The idea behind the verification is the two meta-arguments: *confidence by comparison* and *confidence by association*.

First, for all the queries generated in the emulation phase, the verifier checks that $\left|f^{-1}(y_j^k)\right|$ and $\left|S^{-1}(y_j^k)\right|$ are truly lower-bounded by $n_{y_j^k}$ and $r_{y_j^k}$, respectively. This is done by applying the Goldwasser-Sipser protocol.

Next, the verifier generates a testing series of y's by repeating the following: either picking a random x and computing $f(x)$, or – with much lower probability – by picking a random u and computing $S(u)$. The verifier uses *private coins* for the generation process, and asks the prover to supply Info for each query. This generation process allows the verifier to perform two important tests:

1. For the $S(u)$'s, the verifier is able to apply, not only the Goldwasse-Sipser protocol, but also the Aiello-Hastad protocol, using u as its secret[10].

2. For the $f(x)$'s, the verifier is able to apply the Aiello-Hastad protocol using x as its secret.

Now, the verifier picks at random thresholds $t_1 < t_2 < \cdots$ and considers properties of the form $\geq t_i$ (*e.g.*, whether a query y satisfies $\left|S^{-1}(y)\right| \geq t_i$). The random choice is made as to guarantee that having a property can be one-sidedly verified by the protocol of Goldwasser-Sipser and the protocol of Aiello-Hastad, with high probability, despite their slight inaccuracy.

The verifier verifies that the fraction of queries satisfying each of the properties is roughly the same in the two ways it is sampled from each relevant distribution: for every $1 \leq j \leq q$, it checks that the fraction of k's for which $r_{y_j^k} \geq t_i$ is approximately the fraction of $y_k = S(u_k)$ that satisfy this property; it checks the fraction of k's for which $n_{y_j^k} \geq t_i$ among ok queries is approximately the fraction of ok $y_k = S(u_k)$'s that satisfy this property (note: the prover cannot identify the ok $S(u)$'s among the $f(x)$'s; and for the latter the upper bound on the number of pre-images is verifiable); it also checks that the fraction of very ok y_j^k's is approximately the same as the fraction of very ok $y_k = S(u_k)$.

Formally, the protocol is as follows:

Phase II - Verification:
Parameters:

- $\rho = m^{-\frac{2}{3}}$: estimate parameter in Goldwasser-Sipser and Aiello-Hastad protocols.

1. **Verifying Lower Bounds**:
   (a) P,V: For every $1 \leq k \leq m$, for every $1 \leq j \leq q$, verify $\left|f^{-1}(y_j^k)\right| \geq (1-\rho)n_{y_j^k}$ using the Golwasser-Sipser protocol with amplification parameter $u = \frac{1000qm}{\delta\rho^2}$.
   (b) P,V: For every $1 \leq k \leq m$, for every $1 \leq j \leq q$, verify $\left|S^{-1}(y_j^k)\right| \geq (1-\rho)r_{y_j^k}$ using the Golwasser-Sipser protocol with amplification parameter $u = \frac{1000qm}{\delta\rho^2}$.
2. **Performing Testing Queries**:
   (a) V: sends a random hybrid $y_1, \ldots, y_m$ generated as follows: fix $\alpha = \frac{\delta}{t^{ok} \cdot 1000}$. for every $1 \leq k \leq m$ independently,
      - With probability α, set $k \in Q_1$: choose uniformly $x_k \in \{0,1\}^n$, and let $y_k = f(x_k)$.
      - With probability $1-\alpha$, set $k \in Q_2$: choose uniformly a randomness string u_k, and let $y_k = S(u_k)$.

   (b) P: sends [supposedly] Info(y_1), … ,Info(y_m).
   (c) P,V: For every $k \in Q_1$, check $\left|f^{-1}(y_k)\right| \leq (1+\rho)n_{y_k}$ using the Aiello-Hastad protocol with amplifying parameter $u = \frac{1000m}{\delta\epsilon^2}$.
   (d) P,V: For every $k \in Q_2$, check $\left|S^{-1}(y_k)\right| \leq (1+\rho)r_{y_k}$ using the Aiello-Hastad protocol with amplifying parameter $u = \frac{1000m}{\delta\epsilon^2}$.
   (e) P,V: For every $k \in Q_2$, check $\left|S^{-1}(y_k)\right| \geq (1-\rho)r_{y_k}$ using the Goldwasser-Sipser protocol with amplifying parameter $u = \frac{1000m}{\delta\epsilon^2}$.
3. **Comparing Statistics**:

[10] note: the verifier could not have done that for y_j^k, $j = 1, \ldots, q$, because of the correlations between different queries in the same trial; some unwanted information regarding the verifier's "secrets" could have leaked to the prover this way.

(a) V: picks uniformly at random $\ell_{pre} \in \{1, \ldots, \ell_{max}\}$, for $\ell^{max} = \frac{100}{\delta^2 qm}$. Denote $t_{pre} = \left\lceil \frac{n}{\log(1+\varepsilon)} \right\rceil$. Set thresholds

$$t_0^{pre} = 0 \text{ and } t_i^{pre} = \left(1 + \frac{\varepsilon}{\ell_{pre}}\right) \cdot (1+\varepsilon)^{i-1} \text{ for } i = 1, \ldots, t_{pre}$$

(b) V: picks uniformly at random $\ell_{coin} \in \{1, \ldots, \ell_{max}\}$, for $\ell^{max} = \frac{100}{\delta^2 qm}$. Denote $t_{coin} = \left\lceil \frac{n'}{\log(1+\varepsilon)} \right\rceil$. Set thresholds

$$t_0^{coins} = 0 \text{ and } t_i^{coin} = \left(1 + \frac{\varepsilon}{\ell_{coin}}\right) \cdot (1+\varepsilon)^{i-1} \text{ for } i = 1, \ldots, t_{coin}$$

(c) V:

i. rejects if there exists query $1 \leq j \leq q$ and $1 \leq i \leq t_{coin}$, such that

$$\Pr_{k \in_R [m]} \left[r_{y_j^k} \geq t_i^{coin} \right] < \Pr_{k \in_R Q_2} \left[r_{y_k} \geq t_i^{coin} \right] - \frac{1}{100 q t_{coin}}$$

ii. rejects if there exists query $1 \leq j \leq q$, such that

$$\Pr_{k \in_R [m]} \left[\frac{2^n}{2^{n'}} \cdot \frac{r_{y_j^k}}{n_{y_j^k}} \leq t^{ok} \right] < \Pr_{k \in_R Q_2} \left[\frac{2^n}{2^{n'}} \cdot \frac{r_{y_k}}{n_{y_k}} \leq t^{ok} \right] - \frac{1}{10q}$$

iii. rejects if there exists query $1 \leq j \leq q$ and $1 \leq i \leq t_{pre}$, such that

$$\Pr_{k \in_R [m]} \left[n_{y_j^k} \geq t_i^{pre} \wedge \frac{2^n}{2^{n'}} \cdot \frac{r_{y_j^k}}{n_{y_j^k}} \leq t^{ok} \right] < \Pr_{k \in_R Q_2} \left[n_{y_k} \geq t_i^{pre} \wedge \frac{2^n}{2^{n'}} \cdot \frac{r_{y_k}}{n_{y_k}} \leq t^{ok} \right] - \frac{1}{100 q t_{pre}}$$

Note that by our choice of parameters, the protocol is efficient.

Analysis of The Verification Phase

The analysis is done in several steps. First we address the issue of choosing the thresholds. We explain the problem with arbitrary choice of thresholds, and prove the randomized manner in which we choose them in the protocol solves it. Then, we analyze the behavior of the prover in light of the applications of the Goldwasser-Sipser and Aiello-Hastad protocols. In this context we apply the *confidence by association* argument. Finally, we resolve the assumption of the emulation phase by obtaining the following:

1. The estimates for the r_y's are mostly accurate, *by comparison.*

2. The classification of ok queries is mostly accurate, *by comparison.*

3. The estimates for the n_y's are mostly accurate for ok queries, *by comparison.*

Marginal Issues The protocols of Goldwasser-Sipser and Aiello-Hastad only allow the verifier to check the bounds it has are *approximately* the right bounds, *i.e.*, within factor of either $(1-\rho)$ or $(1+\rho)$, respectively, of a true bound. This slight inaccuracy may pull the rug under the *confidence by comparison* argument: it may cause many y's to be misclassified in both directions, where the argument was based on the assumption that there are only one-sided misclassifications, with high probability.

The idea is to show that the properties considered in the protocol are such that there is not much weight on elements being misclassified due to this inaccuracy, with high probability. The properties in question are of the form *some function of y (how many pre-images it has? what is the probability the reduction queries it? what is the ratio between the two expressions?) is greater than some threshold.* Hence, let us analyze the following generic setting: $\mathcal{F} : Y \to \mathbb{R}^+$ is some bounded non-negative function, $N \stackrel{def}{=} \sup_{y \in Y} \mathcal{F}(y)$. $\mathcal{D}$ is some distribution over Y. $t = \lceil (\frac{N}{\log(1+\varepsilon)}) \rceil$ thresholds are chosen by picking ℓ at random from $\{1, \ldots, \ell_{max}\}$ for some large natural number ℓ_{max} and setting

$$t_i^\ell = \left(1 + \frac{\varepsilon}{\ell}\right) \cdot (1+\varepsilon)^{i-1} \text{ for } i = 1, \ldots, t$$

We say y is $(\mathcal{F}, \rho)$-*marginal* for threshold t_i^ℓ if $\mathcal{F}(y) \in [\frac{1}{1+\rho} t_i^\ell, \frac{1}{1-\rho} t_i^\ell]$.

For small i's, the margins are very narrow. For larger i's, the margins get wider, however, still, all possible margins are disjoint, as argued in the following proposition, whose proof is brought in the appendix:

Proposition 18 (disjoint margins). *If $\rho \leq \frac{\varepsilon\mu^2}{24}$, then for all $(\ell, i_1) \neq (\ell', i_2)$,*

$$\left[\frac{1}{1+\rho} t_{i_1}^\ell, \frac{1}{1-\rho} t_{i_1}^\ell\right] \bigcap \left[\frac{1}{1+\rho} t_{i_2}^{\ell'}, \frac{1}{1-\rho} t_{i_2}^{\ell'}\right] = \phi$$

Proof. The proof is omitted and will appear in the full version of this paper. ■ This proposition immediately allows us to deduce that only few choices of ℓ induce much $\mathcal{D}$-weight on the margins:

Lemma 19. *For any function $\mathcal{F} : Y \to \mathbb{R}^+$, for any distribution $\mathcal{D}$ over Y, if $\rho \leq \frac{\varepsilon\mu^2}{24}$, then*

$$\Pr_\ell \left[\Pr_{y \sim \mathcal{D}} \left[\bigvee_{i=1}^{t} y \textit{ is } (\mathcal{F}, \rho)\textit{-marginal for } t_i^\ell \right] \geq \mu \right] \leq \frac{1}{\mu \ell_{max}}$$

Proof. By proposition 18, $\mathcal{F}(y)$ can be marginal with respect to at most one threshold t_i^ℓ, hence, at most $\frac{1}{\mu}$ of the ℓ_{max} possible ℓ's can satisfy $\Pr_{y \sim \mathcal{D}} \left[\bigvee_{i=1}^t y \text{ is } (\mathcal{F}, \rho)\text{-marginal for } t_i^\ell\right] \geq \mu$. ■

We fix $\mu \stackrel{def}{=} \frac{1}{10qt}$. Hence, for the value of ρ set in the protocol, the premise of the lemma holds. Moreover, for the value of ℓ_{max} set in the protocol, the resulting probability is at most $\frac{\delta}{100}$.

Cheating in Bounds Protocols We first show that the lower-bound and upper-bound protocols of [AiHa, GoSi] do not make the verifier reject in any of the trials with high probability when interacting with an honest prover, while are likely to cause the verifier to reject otherwise.

The first proposition relates to application of the Goldwasser-Sipser protocol.

Proposition 20. *The following hold:*

- *If for every $1 \leq k \leq m$, $1 \leq j \leq q$, $n_{y_j^k} = \left|f^{-1}(y_j^k)\right|$, and $r_{y_j^k} = \left|S^{-1}(y_j^k)\right|$, the verifier does not reject in step 1 with probability at least $1 - \frac{\delta}{50}$.*
- *The probability the verifier does not reject in step 1, but*

$$Y_1 \stackrel{def}{=} \left\{ y_j^k \,\middle|\, \left|f^{-1}(y_j^k)\right| < (1-\rho) n_{y_j^k} \vee \left|S^{-1}(y_j^k)\right| < (1-\rho) r_{y_j^k} \right\} \neq \phi$$

is at most $\frac{\delta}{50}$.

- *The probability the verifier does not reject in step 2e, but*

$$Y_2 \stackrel{def}{=} \left\{ y_j^k \,\middle|\, \left|S^{-1}(y_j^k)\right| < (1-\rho) r_{y_j^k} \right\} \neq \phi$$

  is at most $\frac{\delta}{100}$.

Proof. Recall the fixing of the amplifying parameter in the protocol $u = \left\lceil \frac{1000qm}{\delta\rho^2} \right\rceil$. By theorem 5, the probability the verifier does not reject in any of the independent executions of the Goldwasser-Sipser protocol, if given true answers, is at least

$$\left(1 - \frac{9}{\rho^2 u}\right)^{qm} \geq 1 - \frac{9qm}{\rho^2 u} \geq 1 - \frac{\delta}{100}$$

where the first inequality is Bernoulli inequality (where we used Bernoulli inequality, $(1-x)^n \geq 1 - nx$, for $x > 0$, $n \in \mathbb{N}$). We use a union bound to conclude the first item.

By theorem 5, if $|Y_1| \neq 0$ or $|Y_2| \neq 0$, the probability the prover does not reject in step 1 is at most:

$$\left(\frac{9}{\rho^2 u}\right)^{|Y|} \leq \left(\frac{\delta}{100qm}\right)^{|Y|} \leq \frac{\delta}{100}$$

The last two items follow. ■

The second proposition relates to the size estimates of the testing queries. Here we get pretty accurate estimates in both respects (number of pre-images, weight according to distribution) not because we can check both, but because the prover cannot identify the ok reduction queries from the $f(x)$'s. Note - in steps 2c and 2d the verifier reveals the secret, and exposes which y's were reduction queries and which were $f(x)$'s. The crucial point in our analysis will be that, in step 2b, when the prover has to supply the information about the y's, there is not enough information to identify reduction queries with good probability.

Lemma 21. *The following hold:*

- *If for every* $1 \leq k \leq m$, $n_{y_k} = \left|f^{-1}(y_k)\right|$ *and* $r_{y_k} = \left|S^{-1}(y_k)\right|$, *the verifier does not reject in step 2d with probability at least* $1 - \frac{\delta}{50}$.
- *The probability the verifier does not reject in step 2d, but either*

$$Y_1 = \left\{ y_k \mid \hat{w}(y_k) \leq t^{ok} \wedge \left|f^{-1}(y_k)\right| > (1+\rho) n_{y_k} \right\}$$

  satisfies $|Y_1| \geq \frac{12 \ln \frac{100}{\delta}}{\rho}$, *or*

$$Y_2 = \left\{ y_k \mid k \in Q_2 \wedge \left|S^{-1}(y_k)\right| > (1+\rho) r_{y_k} \right\}$$

  satisfies $|Y_2| \geq \frac{12 \ln \frac{100}{\delta}}{\rho}$, *is at most* $\frac{\delta}{50}$.

Proof. Recall the fixing of the amplifying parameter in the protocol $u = \left\lceil \frac{1000m}{\delta\rho^2} \right\rceil$. By theorem 6, the probability the verifier does not reject in any of the independent executions of the Aiello-Hastad protocol, if given true answers, is at least

$$\left(1 - \frac{9}{\rho^2 u}\right)^{m} \geq 1 - \frac{9m}{\rho^2 u} \geq 1 - \frac{\delta}{100}$$

where the first inequality is Bernoulli inequality.

By theorem 6, the probability the prover does not reject in step 2d, when $|Y_2| \geq \frac{12\ln\frac{100}{\delta}}{\rho}$, is at most:

$$\left(1-\frac{\rho}{6}+\frac{9}{\rho^2 u}\right)^{|Y_2|} \leq \left(1-\frac{\rho}{6}+\frac{\delta}{100m}\right)^{|Y_2|} \leq e^{-|Y_2|\left(\frac{\rho}{6}-\frac{\delta}{m}\right)} \leq \frac{\delta}{100}$$

[where we used $\frac{\rho}{6}-\frac{\delta}{m} \geq \frac{\rho}{12}$]

It remains to examine $|Y_1|$. For this purpose, let us show that given the prover's view in step 2b, there is a small probability any ok query in the hybrid belongs to Q_2. Note that the event $k \in Q_2$ is independent of events $y_i = y$, for any y, for every $i \neq k$. Hence it suffices to show the following:

Claim 21.1. *For every $1 \leq k \leq m$, for every ok query y,*

$$\Pr\left[k \in Q_2 | y_k = y\right] \leq \frac{\delta}{1000}$$

Proof. By Bays' formula,

$$\begin{aligned}
\Pr\left[k \in Q_2 | y_k = y\right] &= \frac{\Pr\left[y_k = y | k \in Q_2\right]\Pr\left[k \in Q_2\right]}{\Pr\left[y_k = y\right]} \\
&\leq \frac{\Pr\left[y_k = y | k \in Q_2\right]\Pr\left[k \in Q_2\right]}{\Pr\left[y_k = y | k \in Q_1\right]\Pr\left[k \in Q_1\right]} \\
&= \frac{2^{-n'}\left|S^{-1}(y)\right|\alpha}{2^{-n}\left|f^{-1}(y)\right|(1-\alpha)} \\
&\leq t^{ok}\cdot\frac{\alpha}{1-\alpha} \\
&\leq \frac{\delta}{1000}
\end{aligned}$$

■(of claim 21.1)

Thus, by linearity of expectations, $\mathbf{E}\left[|Q_2 \cap Y_1|\right] \leq \frac{\delta}{1000}|Y_1|$. Hence, by Markov's inequality, $\Pr\left[|Q_2 \cap Y_1| \geq \frac{1}{5}|Y_1|\right] \leq \frac{\delta}{200}$. As above, by theorem 6, the probability the prover does not reject in step 2c, is at most:

$$\left(1-\frac{\rho}{6}+\frac{9}{\rho^2 u}\right)^{|Q_1\cap Y_1|} \leq \left(1-\frac{\rho}{6}+\frac{\delta}{100m}\right)^{|Q_1\cap Y_1|} \leq e^{-|Q_1\cap Y_1|\left(\frac{\rho}{6}-\frac{\delta}{m}\right)}$$

Therefore, either $|Y_1|$ is not as large as stated in the lemma, or the probability the prover does not reject in step 2c is at most $\frac{\delta}{100}$. The lemma follows. ■(of lemma 21)

Confidence By Comparison First we argue that by comparison, the verifier has accurate estimates for the weights of the reduction's queries.

Lemma 22 (coins mean). *For every $1 \leq j \leq q$, for every $1 \leq i \leq t_{coin}$, if we denote:*

- $e_{i,j} \stackrel{def}{=} \Pr_{k\in_R[m]}\left[\left|S^{-1}(y_j^k)\right| \geq t_i^{coin}\right]$
- $s_i \stackrel{def}{=} \Pr_{k\in_R Q_2}\left[\left|S^{-1}(y_k)\right| \geq t_i^{coin}\right]$

then $|e_{i,j} - s_i| \leq \frac{1}{100qt_{coin}}$, with probability at least $1-\frac{\delta}{100qt_{coin}}$.

Proof. Fix some $1 \le j \le q$ and $1 \le i \le t$. As the members of $\{y_1, \ldots, y_m\}$ and $\left\{y_j^1, \ldots, y_j^k\right\}$ are independently distributed according to $S(U)$, lemma 16 implies that, with probability at least $1 - \frac{\delta}{100qt_{coin}}$, if $p_i \stackrel{def}{=} \Pr_{y \sim S(U)}\left[\left|S^{-1}(y)\right| \ge t_i^{coin}\right]$, then

$$\begin{aligned} |e_{i,j} - s_i| &= |(e_{i,j} - p_i) + (p_i - s_i)| \\ &\le |e_{i,j} - p_i| + |p_i - s_i| \\ &\le 2 \cdot \sqrt{\frac{9 \ln \frac{1}{\delta}}{m}} \\ &\le \frac{1}{100qt_{coin}} \end{aligned}$$

■

Lemma 23 (estimates for coins). *With probability at least $1 - \frac{\delta}{10}$, at least one of the following two events occur:*

- *the verifier rejects during the verification phase.*
- *for every $1 \le j \le q$,*

$$\Pr_{k \in_R [m]}\left[r_{y_j^k}(1-\varepsilon) \le \left|S^{-1}(y_j^k)\right| \le r_{y_j^k}(1+\varepsilon)\right] \ge 1 - \frac{1}{10q}$$

Proof. By proposition 20, with probability at least $1 - \frac{\delta}{100}$, either the verifier rejects in step 1b, or for every trial $1 \le k \le m$, for every query $1 \le j \le q$,

$$\left|S^{-1}(y_j^k)\right| \ge (1-\rho) r_{y_j^k} \ge (1-\varepsilon) r_{y_j^k}$$

Let us focus on the following event, and show it occurs with low probability:

- the verifier does not reject during the verification phase.
- for every trial $1 \le k \le m$, for every query $1 \le j \le q$, $\left|S^{-1}(y_j^k)\right| \ge (1-\varepsilon) r_{y_j^k}$.
- there exists $1 \le j_0 \le q$, for which

$$\Pr_{k \in_R [m]}\left[\left|S^{-1}(y_{j_0}^k)\right| > r_{y_{j_0}^k}(1+\varepsilon)\right] > \frac{1}{10q}$$

The last item implies the existence of a threshold $t_{i_0}^{coin}$ such that for at least $\frac{1}{10qt_{coin}}$ of the $y_{j_0}^k$'s, $r_{j_0}^k < t_{i_0}^{coin}$, however $\left|S^{-1}(y_{j_0}^k)\right| \ge t_{i_0}^{coin}$.

In addition, if $r_{y_{j_0}^k} \ge t_{i_0}^{coin}$, then $\left|S^{-1}(y_{j_0}^k)\right| \ge t_{i_0}^{coin}$, unless $\frac{1}{1+\rho} \cdot t_{i_0}^{coin} \le \left|S^{-1}(y_{j_0}^k)\right| \le \frac{1}{1-\rho} \cdot t_{i_0}^{coin}$. By lemma 19 and lemma 16, with probability at least $1 - \frac{\delta}{50}$,

$$\Pr_{k \in_R [m]}\left[\frac{1}{1+\rho} \cdot t_{i_0}^{coin} \le \left|S^{-1}(y_{j_0}^k)\right| \le \frac{1}{1-\rho} \cdot t_{i_0}^{coin}\right] \le \frac{1}{5qt_{coin}}$$

Hence, with probability at least $1-\frac{3\delta}{100}$,

$$\begin{aligned}\Pr_{k\in_R[m]}\left[\left|S^{-1}(y_{j_0}^k)\right|\geq t_{i_0}^{coin}\right] &\geq \Pr_{k\in_R[m]}\left[r_{y_{j_0}^k}\geq t_{i_0}^{coin}\right]-\frac{1}{5qt_{coin}}+\Pr_{k\in_R[m]}\left[\left|S^{-1}(y_{j_0}^k)\right|\geq t_{i_0}^{coin}\wedge r_{y_{j_0}^k}<t_{i_0}^{coin}\right]\\ &\geq \Pr_{k\in_R[m]}\left[r_{y_{j_0}^k}\geq t_{i_0}^{coin}\right]+\frac{4}{5qt_{coin}}\end{aligned}$$

Since the verifier does not reject in step 3(c)i, it must be that

$$\Pr_{k\in_R[m]}\left[r_{y_{j_0}^k}\geq t_{i_0}^{coin}\right]\geq \Pr_{k\in_R Q_2}\left[r_{y_k}\geq t_{i_0}^{coin}\right]-\frac{1}{100qt_{coin}}$$

Let us lower bound $\Pr_{k\in_R Q_2}\left[r_{y_k}\geq t_{i_0}^{coin}\right]$. If $\left|S^{-1}(y_k)\right|\geq t_{i_0}^{coin}$, then $r_{y_k}\geq t_{i_0}^{coin}$, unless k belongs to one of the following:

- (margins) $Y_1\stackrel{def}{=}\left\{k\in Q_2\mid \frac{1}{1+\rho}\cdot t_{i_0}^{coin}\leq \left|S^{-1}(y_k)\right|\leq\frac{1}{1-\rho}\cdot t_{i_0}^{coin}\right\}$
- (cheating) $Y_2\stackrel{def}{=}\left\{k\in Q_2\mid \left|S^{-1}(y_k)\right|>(1+\rho)r_{y_k}\right\}$

By lemma 19 and lemma 16, with probability at least $1-\frac{\delta}{50}$, $|Y_1|\leq\frac{|Q_2|}{100qt_{coin}}$. By lemma 21, with probability at least $1-\frac{\delta}{100}$, $|Y_2|\leq\frac{|Q_2|}{10qt_{coin}}$. Hence, with probability at least $1-\frac{3\delta}{100}$,

$$\begin{aligned}\Pr_{k\in_R Q_2}\left[\left|S^{-1}(y_k)\right|\geq t_{i_0}^{coin}\right] &\geq \Pr_{k\in_R Q_2}\left[r_{y_k}\geq t_{i_0}^{coin}\right]-\frac{|Y_1|}{|Q_2|}-\frac{|Y_2|}{|Q_2|}\\ &\geq \Pr_{k\in_R Q_2}\left[r_{y_k}\geq t_{i_0}^{coin}\right]-\frac{1}{10qt_{coin}}\end{aligned}$$

Summing up,

$$\Pr_{k\in_R[m]}\left[\left|S^{-1}(y_j^k)\right|\geq t_{i_0}^{coin}\right] \geq \Pr_{k\in_R Q_2}\left[\left|S^{-1}(y_k)\right|\geq t_{i_0}^{coin}\right]+\frac{1}{qt_{coin}}$$

By lemma 22, this happens with probability at most $\frac{\delta}{20}$. Therefore, the probability the event occurs is at most $\frac{\delta}{10}$, and the lemma follows. ■

Next we argue that, by comparison, the verifier gets accurate classification of ok queries.

Lemma 24 (ok mean). *For every $1\leq j\leq q$, if we denote:*

- $e_j\stackrel{def}{=}\Pr_{k\in_R[m]}\left[\hat{w}(y_j^k)\geq t^{ok}\right]$
- $s\stackrel{def}{=}\Pr_{k\in_R Q_2}\left[\hat{w}(y_k)\geq t^{ok}\right]$

then $|e_j-s|\leq\frac{1}{10q}$, with probability at least $1-\frac{\delta}{10q}$.

Proof. Fix some $1\leq j\leq q$. As the members of $\{y_1,\ldots,y_m\}$ and $\left\{y_j^1,\ldots,y_j^k\right\}$ are independently distributed according to $S(U)$, lemma 16 implies that, with probability at least $1-\frac{\delta}{10q}$, if $p\stackrel{def}{=}\Pr_{y\sim S(U)}\left[\hat{w}(y)\geq t^{ok}\right]$, then

$$\begin{aligned}|e_{i,j}-s_i| &= |(e_{i,j}-p_i)+(p_i-s_i)|\\ &\leq |e_{i,j}-p_i|+|p_i-s_i|\\ &\leq 2\cdot\sqrt{\frac{9\ln\frac{1}{\delta}}{m}}\\ &\leq \frac{\delta}{10q}\end{aligned}$$

The lemma follows. ■

Lemma 25 (classification of ok queries). *With probability at least* $1-\frac{\delta}{5}$, *at least one of the following two events occur:*

- *the verifier rejects during the verification phase.*
- *for every* $1 \leq j \leq q$,

$$\Pr_{k\in_R[m]}\left[\frac{2^n}{2^{n'}}\cdot\frac{r_{y_j^k}}{n_{y_j^k}} \leq t^{ok} \leftrightarrow \hat{w}(y_j^k) \geq t^{ok}\right] \geq 1-\frac{1}{4q}$$

Proof. Fix $1 \leq j \leq q$.

By lemma 23 and proposition 20, the following event occurs with probability at least $1-\frac{\delta}{5}$: either the verifier rejects during the verification phase, or for every query $1 \leq j \leq q$,

$$\Pr_{k\in_R[m]}\left[\frac{2^n}{2^{n'}}\cdot\frac{r_{y_j^k}}{n_{y_j^k}} \geq \frac{1-\rho}{1+\varepsilon}\cdot\frac{\left|S^{-1}(y_j^k)\right|}{\left|f^{-1}(y_j^k)\right|} \geq (1-3\varepsilon)\cdot\hat{w}(y_j^k)\right] \geq 1-\frac{1}{5q}$$

If $2^n r_{y_j^k}/2^{n'} n_{y_j^k} \leq t^{ok}$, then $\hat{w}(y_j^k) \leq t^{ok}$, unless $t^{ok} \leq \hat{w}(y_j^k) \leq \frac{1}{1-3\varepsilon}\cdot t^{ok}$. By lemma 19 and lemma 16,

$$\Pr_{k\in_R[m]}\left[\frac{1}{1+\rho}\cdot t^{ok} \leq \hat{w}(y_{j_0}^k) \leq \frac{1}{1-\rho}\cdot t^{ok}\right] \leq \frac{1}{5q}$$

Hence, with probability at least $1-\frac{3\delta}{100}$, either the verifier rejects during the verification, or

$$\Pr_{k\in_R[m]}\left[\hat{w}(y_j^k) > t^{ok} \ \wedge\ \frac{2^n}{2^{n'}}\cdot\frac{r_{y_j^k}}{n_{y_j^k}} \leq t^{ok}\right] \quad\leq\quad \frac{1}{5q} \tag{3}$$

But also: with probability at least $1-\frac{3\delta}{100}$, either the verifier rejects during the verification, or

$$\Pr_{k\in_R[m]}\left[\hat{w}(y_j^k) \leq t^{ok} \ \wedge\ \frac{2^n}{2^{n'}}\cdot\frac{r_{y_j^k}}{n_{y_j^k}} > t^{ok}\right] \quad\leq\quad \Pr_{k\in_R[m]}\left[\hat{w}(y_j^k) \leq t^{ok}\right] - \Pr_{k\in_R[m]}\left[\frac{2^n}{2^{n'}}\cdot\frac{r_{y_j^k}}{n_{y_j^k}} \leq t^{ok}\right] + \frac{1}{5q}$$

Since the verifier does not reject in step 3(c)ii,

$$\Pr_{k\in_R[m]}\left[\frac{2^n}{2^{n'}}\cdot\frac{r_{y_j^k}}{n_{y_j^k}} \leq t^{ok}\right] \geq \Pr_{k\in_R Q_2}\left[\frac{2^n}{2^{n'}}\cdot\frac{r_{y_k}}{n_{y_k}} \leq t^{ok}\right] - \frac{1}{10q}$$

Let us lower bound $\Pr_{k\in_R Q_2}\left[2^n r_{y_k}/2^{n'} n_{y_k} \leq t^{ok}\right]$. If $\hat{w}(y_k) \leq t^{ok}$, then $2^n r_{y_k}/2^{n'} n_{y_k} \leq t^{ok}$, unless k belongs to one of the following:

- (margins) $Y_1 \stackrel{def}{=} \left\{k \in Q_2 \mid \frac{1}{1+\rho}\cdot t^{ok} \leq \hat{w}(y_k) \leq \frac{1}{1-\rho}\cdot t^{ok}\right\}$.
- (coins cheating) $Y_2 \stackrel{def}{=} \left\{k \in Q_2 \mid \left|S^{-1}(y_k)\right| < (1+\rho) r_{y_k}\right\}$.
- (pre-images cheating) $Y_3 \stackrel{def}{=} \left\{k \in Q_2 \mid \hat{w}(y_k) \leq \frac{1}{1+\rho}\cdot t^{ok} \wedge \left|f^{-1}(y_k)\right| > (1+\rho) n_{y_k}\right\}$.

By lemma 19 and lemma 16, with probability at least $1-\frac{\delta}{50}$, $|Y_1| \le \frac{|Q_2|}{100q}$. By lemma 20, with probability at least $1-\frac{\delta}{100}$, either the verifier rejects during the verification, or $|Y_2| \le \frac{|Q_2|}{100q}$. By lemma 21, with probability at least $1-\frac{\delta}{100}$, either the verifier rejects during the verification, or $|Y_2| \le \frac{|Q_2|}{100q}$. Hence, with probability at least $1-\frac{3\delta}{100}$,

$$\Pr_{k\in_R Q_2}\left[\hat{w}(y_k) \le t^{ok}\right] \quad \le \quad \Pr_{k\in_R Q_2}\left[\frac{2^n}{2^{n'}}\cdot\frac{r_{y_k}}{n_{y_k}} \le t^{ok}\right] - \frac{1}{10q}$$

By lemma 24, with probability at least $1-\frac{\delta}{100}$, either the verifier rejects during verification, or

$$\Pr_{k\in_R[m]}\left[\hat{w}(y_j^k) \le t^{ok}\right] - \Pr_{k\in_R Q_2}\left[\hat{w}(y_k) \le t^{ok}\right] \le \frac{1}{10q}$$

Thus, with probability at least $1-\frac{\delta}{50}$, either the verifier rejects during verification, or

$$\Pr_{k\in_R[m]}\left[\hat{w}(y_j^k) \le t^{ok} \ \wedge \ \frac{2^n}{2^{n'}}\cdot\frac{r_{y_j^k}}{n_{y_j^k}} > t^{ok}\right] \quad \le \quad \frac{1}{4q} \tag{4}$$

Inequalities 3-4 yield the statement. ∎

Lemma 26 (pre-images mean). *For every $1 \le j \le q$, for every $1 \le i \le t_{pre}$, if we denote:*

- $e_{i,j} \stackrel{def}{=} \Pr_{k\in_R[m]}\left[\left|f^{-1}(y_j^k)\right| \ge t_i^{pre} \ \wedge \ \hat{w}(y_j^k) \le t^{ok}\right]$
- $s_i \stackrel{def}{=} \Pr_{k\in_R Q_2}\left[\left|f^{-1}(y_k)\right| \ge t_i^{pre} \ \wedge \ \hat{w}(y_k) \le t^{ok}\right]$

then $|e_{i,j}-s_i| \le \frac{1}{100qt_{pre}}$, with probability at least $1-\frac{\delta}{100qt_{pre}}$.

Proof. Fix some $1 \le j \le q$ and $1 \le i \le t_{pre}$. As the members of $\{y_1,\ldots,y_m\}$ and $\left\{y_j^1,\ldots,y_j^k\right\}$ are independently distributed according to $S(U)$, lemma 16 implies that, with probability at least $1-\frac{\delta}{100qt_{pre}}$, if $p_i \stackrel{def}{=} \Pr_{y\sim S(U)}\left[\left|f^{-1}(y)\right| \ge t_i^{pre} \ \wedge \ \hat{w}(y) \le t^{ok}\right]$, then

$$\begin{aligned}
|e_{i,j}-s_i| &= |(e_{i,j}-p_i)+(p_i-s_i)| \\
&\le |e_{i,j}-p_i|+|p_i-s_i| \\
&\le 2\cdot\sqrt{\frac{9\ln\frac{1}{\delta}}{m}} \\
&\le \frac{\delta}{100qt_{pre}}
\end{aligned}$$

∎

Lemma 27 (estimate on number of pre-images of ok queries). *With probability at least $1-\frac{\delta}{10}$, at least one of the following two events occur:*

- *the verifier rejects during the verification phase.*
- *for every query index $1 \le j \le q$, for every threshold index $1 \le i_0 \le t^{pre}$,*

$$\Pr_{k\in_R[m]}\left[\left|f^{-1}(y_j^k)\right| \ge t_{i_0}^{pre} \wedge \hat{w}(y_j^k) \le t^{ok} \wedge n_{y_j^k} < t_{i_0}^{pre}\right] \quad \le \quad \frac{1}{4qt_{pre}}$$

Proof. By proposition 20, with probability at least $1 - \frac{\delta}{100}$, either the verifier rejects in step 1b, or for every trial $1 \le k \le m$, for every query $1 \le j \le q$,

$$\left| f^{-1}(y_j^k) \right| \ge (1-\rho) n_{y_j^k}$$

Let $1 \le i_0 \le t_{pre}$ be some threshold index. Let $1 \le j_0 \le q$ be some query index. If $n_{y_j^k} \ge t_{i_0}^{pre}$, then $\left| f^{-1}(y_j^k) \right| \ge t_{i_0}^{pre}$, unless $\frac{1}{1+\rho} \cdot t_{i_0}^{pre} \le \left| f^{-1}(y_j^k) \right| \le \frac{1}{1-\rho} \cdot t_{i_0}^{pre}$. By lemma 19 and 16, this happens with probability at most $\frac{1}{5qt_{pre}}$. Thus, with probability at least $1 - \frac{3\delta}{100}$,

$$\begin{aligned} \Pr_{k \in_R [m]} \left[\left| f^{-1}(y_j^k) \right| \ge t_{i_0}^{pre} \wedge \hat{w}(y_j^k) \le t^{ok} \right] \quad &\ge \quad \Pr_{k \in_R [m]} \left[n_{y_j^k} \ge t_{i_0}^{pre} \wedge \hat{w}(y_j^k) \le t^{ok} \right] - \frac{1}{5qt_{pre}} \\ &+ \quad \Pr_{k \in_R [m]} \left[\left| f^{-1}(y_j^k) \right| \ge t_{i_0}^{pre} \wedge \hat{w}(y_j^k) \le t^{ok} \wedge n_{y_j^k} < t_{i_0}^{pre} \right] \end{aligned}$$

Since the verifier does not reject in step 3(c)iii, it must be that

$$\Pr_{k \in_R [m]} \left[r_{y_j^k} \ge t_{i_0}^{pre} \wedge \hat{w}(y_j^k) \le t^{ok} \right] \quad \ge \quad \Pr_{k \in_R Q_2} \left[n_{y_k} \ge t_{i_0}^{pre} \wedge \hat{w}(y_k) \le t^{ok} \right] - \frac{1}{10qt_{pre}}$$

Let us lower bound the latter probability. If $\left| f^{-1}(y_k) \right| \ge t_{i_0}^{pre}$, as well as $\hat{w}(y_k) \le t^{ok}$, then $n_{y_k} \ge t_{i_0}^{pre}$, unless k belongs to one of the following:

- (margins) $Y_1 \stackrel{def}{=} \left\{ k \in Q_2 \mid \frac{1}{1+\rho} \cdot t_{i_0}^{pre} \le \left| f^{-1}(y_k) \right| \le \frac{1}{1-\rho} \cdot t_{i_0}^{pre} \right\}$.
- (cheating) $Y_2 \stackrel{def}{=} \left\{ k \in Q_2 \mid \left| f^{-1}(y_k) \right| > (1+\rho) n_{y_k} \wedge \hat{w}(y_k) \le t^{ok} \right\}$.

By lemma 19 and 16, with probability at least $1 - \frac{\delta}{50}$, $|Y_1| \le \frac{|Q_2|}{100qt_{pre}}$. By lemma 21, with probability at least $1 - \frac{\delta}{100}$, $|Y_2| \le \frac{|Q_2|}{100qt_{pre}}$. Hence, with probability at least $1 - \frac{3\delta}{100}$,

$$\Pr_{k \in_R Q_2} \left[\left| f^{-1}(y_k) \right| \ge t_{i_0}^{pre} \wedge \hat{w}(y_k) \le t^{ok} \right] \quad \ge \quad \Pr_{k \in_R Q_2} \left[n_{y_k} \ge t_{i_0}^{pre} \wedge \hat{w}(y_k) \le t^{ok} \right] - \frac{1}{10qt_{pre}}$$

By lemma 26, with probability at least $1 - \frac{\delta}{100}$,

$$\Pr_{k \in_R [m]} \left[\left| f^{-1}(y_j^k) \right| \ge t_{i_0}^{pre} \wedge \hat{w}(y_j^k) \le t^{ok} \right] - \Pr_{k \in_R Q_2} \left[\left| f^{-1}(y_k) \right| \ge t_{i_0}^{pre} \wedge \hat{w}(y_k) \le t^{ok} \right] \quad \le \quad \frac{1}{10qt_{pre}}$$

Hence, with probability at least $1 - \frac{\delta}{10}$,

$$\Pr_{k \in_R [m]} \left[\left| f^{-1}(y_j^k) \right| \ge t_{i_0}^{pre} \wedge \hat{w}(y_j^k) \le t^{ok} \wedge n_{y_j^k} < t_{i_0}^{pre} \right] \quad \le \quad \frac{1}{4qt_{pre}}$$

■

We tie the loose ends in the following theorem establishing the validity of the assumption in the emulation phase.

Corollary 28 (emulation assumption resolved). *With probability at least* $1 - \frac{\delta}{2}$, *at least one of the following two events occur:*

- *the verifier rejects during the verification phase.*

- *there exists a trial* $1 \le k \le m$*, such that for every* $1 \le j \le q$*,*

  1. $\frac{r_{y_j^k}}{n_{y_j^k}} \le t^{ok}$ *if and only if* $\hat{w}(y_j^k) \le t^{ok}$*.*

  2. *if* $\hat{w}(y_j^k) \le t^{ok}$*, then* $n_{y_j^k}(1-\varepsilon) \le \left|f^{-1}(y_j^k)\right| \le n_{y_j^k}(1+\varepsilon)$*,*

Proof. Let us focus on the event that the verifier does not reject.

Applying a union-bound over the j's in lemma 25, with probability at least $1 - \frac{\delta}{10}$, for at least $\frac{3}{4}$ of the $k \in [m]$,

$$\forall 1 \le j \le q \quad \frac{2^n}{2^{n'}} \cdot \frac{r_{y_j^k}}{n_{y_j^k}} \le t^{ok} \leftrightarrow \hat{w}(y_j^k) \le t^{ok}$$

Let us consider the estimate of the number of pre-images. By proposition 20, with probability at least $1 - \frac{\delta}{100}$,

$$\forall 1 \le k \le m \; \forall 1 \le j \le q \quad \left|f^{-1}(y_j^k)\right| \ge (1-\varepsilon) n_{y_j^k}$$

Let us consider the estimate in the other direction. Let $k \in [m]$ be a trial, for which there exists ok query $1 \le j_0 \le q$, such that $\left|f^{-1}(y_{j_0}^k)\right| > n_{y_{j_0}^k}(1+\varepsilon)$. Then there necessarily exists a threshold $1 \le i_0 \le t^{pre}$, for which $\left|f^{-1}(y_{j_0}^k)\right| \ge t_{i_0}^{pre}$, but $n_{y_{j_0}^k} < t_{i_0}^{pre}$. Applying a union bound on lemma 27, with probability at least $1 - \frac{\delta}{10}$, for at most $\frac{1}{4}$ of the $k \in [m]$ there exist such i_0, j_0. The corollary follows. ■

The correctness of the entire protocol, *i.e.*, the emulation phase and the verification phase concatenated together, immediately follows:

Theorem 29 (correctness of protocol). *The following hold:*

- (Completeness) *There exists a prover strategy (being honest) for which the verifier does not reject throughout the protocol, with probability at least* $1 - \delta$*.*

- (Soundness) *The probability the verifier does not reject in the protocol, however* $x \in L$*, is at most* δ*.*

Proof. Completeness follows by a union-bound from theorem 17, as well as lemmata 22, 24 and 26. Soundness follows by a union-bound from corollary 28 and theorem 17. ■