arXiv:1312.2490v2 [cs.CC] 24 Mar 2014

A New View on Worst-Case to Average-Case Reductions for NP Problems

Thomas Holenstein* Robin Künzler†

September 17, 2018

Abstract

We study the result by Bogdanov and Trevisan (FOCS, 2003), who show that under reasonable assumptions, there is no non-adaptive reduction that bases the average-case hardness of an NP-problem on the worst-case complexity of an NP-complete problem. We replace the hiding and the heavy samples protocol in [BT03] by employing the histogram verification protocol of Haitner, Mahmoody and Xiao (CCC, 2010), which proves to be very useful in this context. Once the histogram is verified, our hiding protocol is directly public-coin, whereas the intuition behind the original protocol inherently relies on private coins.

*ETH Zurich, Department of Computer Science, 8092 Zurich, Switzerland. E-mail: thomas.holenstein@inf.ethz.ch

†ETH Zurich, Department of Computer Science, 8092 Zurich, Switzerland. E-mail: robink@inf.ethz.ch

Contents

1 Introduction

One-way functions are functions that are easy to compute on any instance, and hard to invert on average. Assuming their existence allows the construction of a wide variety of secure cryptographic schemes. Unfortunately, it seems we are far from proving that one-way functions indeed exist, as this would imply $\mathsf{BPP} \neq \mathsf{NP}$. Thus, the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$, which states that there exists a worst-case hard problem in NP, is weaker. The following question is natural:

> **Question 1:** Does $\mathsf{NP} \not\subseteq \mathsf{BPP}$ imply that one-way functions (or other cryptographic primitives) exist?

A positive answer to this question implies that the security of the aforementioned cryptographic schemes can be based solely on the worst-case assumption $\mathsf{NP} \not\subseteq \mathsf{BPP}$.

Given a one-way function f and an image y, the problem of finding a preimage $x \in f^{-1}(y)$ is an NP-problem: provided a candidate solution x, one can efficiently verify it by checking if $f(x) = y$. In this sense, a one-way function provides an NP problem that is hard to solve on average, and Question 1 asks whether it can be based on worst-case hardness. Thus, the question is closely related to the study of *average-case complexity*, and in particular to the set distNP of distributional problems $(L, \mathcal{D})$, where $L \in \mathsf{NP}$, and $\mathcal{D}$ is an ensemble of efficiently samplable distributions over problem instances. We say that a distNP problem $(L, \mathcal{D})$ is hard if there is no efficient algorithm that solves the problem (with high probability) on instances sampled from $\mathcal{D}$. In this setting, analogously to Question 1, we ask:

> **Question 2:** Does $\mathsf{NP} \not\subseteq \mathsf{BPP}$ imply that there exists a hard problem in distNP?

A natural approach to answer Question 2 affirmatively is to give a so-called *worst-case to average-case reduction* from some NP-complete L to $(L', \mathcal{D}) \in \mathsf{distNP}$: such a reduction R^O is a polynomial time algorithm with black-box access to an oracle O that solves $(L', \mathcal{D})$ on average, such that $\Pr_R[R^O(x) = L(x)] \geq 2/3$. We say a reduction is *non-adaptive* if the algorithm R fixes all its queries to O in the beginning (see Section 2.6 for a formal definition). Bogdanov and Trevisan [BT06b] (building on work by Feigenbaum and Fortnow [FF93]) show that it is unlikely that a non-adaptive worst-to-average-case reduction exists:

> **Main result of [BT06b] (informal):** If there exists a non-adaptive worst-case to average-case reduction from an NP-complete problem to a problem in distNP, then $\mathsf{NP} \subseteq \mathsf{coNP}/\mathrm{poly}$.

The consequence $\mathsf{NP} \subseteq \mathsf{coNP}/\mathrm{poly}$ implies a collapse of the polynomial hierarchy to the third level [Yap83], which is believed to be unlikely.

The work of Impagliazzo and Levin [IL90] and Ben-David et al. [BDCGL92] shows that an algorithm that solves a problem in distNP can be turned (via a non-adaptive reduction) into an algorithm that solves the search version of the same problem. Thus, as inverting a one-way function well on average corresponds to solving a distNP search problem well on average, the result of [BT06b] also implies that Question 1 cannot be answered positively by employing non-adaptive reductions, unless the polynomial hierarchy collapses.

1.1 Contributions of this Paper

The proof of the main result in [BT06b] proceeds as follows. Assuming that there exists a non-adaptive worst-case to average-case reduction R from an NP-complete language L to $(L', \mathcal{D}) \in \mathsf{distNP}$, it is shown that L and its complement both have a constant-round interactive proof with advice (i.e. L and $\overline{L}$ are in AM/poly according to Definition 2.3). As AM/poly = NP/poly, this gives $\mathsf{coNP} \subseteq \mathsf{NP}$/poly. The final AM/poly protocol consists of three sub-protocols: the heavy samples protocol, the hiding protocol, and the simulation protocol. Using the protocol to verify the histogram of a probability distribution by Haitner et al. [HMX10], we replace the heavy samples protocol and the hiding protocol. Our protocols have several advantages. The heavy samples protocol becomes quite simple, as one only needs to read a probability from the verified histogram. Furthermore, once the histogram is verified, our hiding protocol is directly public-coin, whereas the intuition behind the original hiding protocol crucially uses that the verifier can hide its randomness from the prover. Our protocol is based on a new and different intuition and achieves the same goal. Clearly, one can obtain a public-coin version of the original hiding protocol by applying the Goldwasser-Sipser transformation [GS86], but this might not provide a different intuition. Finally, our protocols show that the histogram verification protocol of [HMX10] is a very useful primitive to approximate probabilities using AM-protocols.

1.2 Related Work

Recall that our Question 2 above asked if average-case hardness can be based on the worst-case hardness of an NP-complete problem. The question if cryptographic primitives can be based on NP-hardness was stated as Question 1.

Average-case complexity. We use the definition of distNP and average-case hardness from [BT06b]. The hardness definition is essentially equivalent to Impagliazzo's notion of heuristic polynomial-time algorithms [Imp95]. We refer to the surveys of Impagliazzo [Imp95], Goldreich [Gol97], and Bogdanov and Trevisan [BT06a] on average-case complexity.

Negative results on Question 2. Feigenbaum and Fortnow [FF93] study a special case of worst-case to average-case reductions, called random self-reductions. Such a reduction is non-adaptive, and reduces L to itself, such that the queries are distributed uniformly at random (but not necessarily independently). They showed that the existence of a random self-reduction for an NP-complete problem is unlikely, as it implies $\mathsf{coNP} \subseteq \mathsf{NP}$/poly and the polynomial hierarchy collapses to the third level. This result generalizes to the case of non-adaptive reductions from $L \in \mathsf{NP}$ to $L' \in \mathsf{distNP}$ where the queries are distributed according to a distribution P that does not depend on the input x to the reduction, but only on the length of x.

The study of random self-reductions was motivated by their use to design interactive proof systems and (program-) checkers[1]. Checkers are introduced by Blum and Blum and Kannan [Blu88, BK95]. Rubinfeld [Rub90] shows that problems that have a random self-reduction and are downward self-reducible (i.e. they can be reduced to solving the same problem on smaller instances) have a program checker. Random self-reductions can be used to prove the worst-case to average-case equivalence of certain PSPACE-complete and EXP-complete problems [STV01]. A long-standing open question is whether SAT is checkable. In this context, Mahmoody and Xiao [MX10] show that if one-way functions can be based on NP-hardness via a randomized, possibly adaptive reduction, then SAT is checkable.

In the context of program checking, Blum et al. [BLR93] introduce the notion of self-correctors. A self-corrector is simply a worst-case to average-case reduction from L to $(L', \mathcal{D})$, where $L = L'$. Clearly, a random self-reduction is also a self-corrector.

[1] Checkers allow to ensure the correctness of a given program on an input-by-input basis. Formally, a checker is an efficient algorithm C that, given oracle access to a program P which is supposed to decide a language L, has the following properties for any instance x. Correctness: If P is always correct, then $C^P(x) = L(x)$ with high probability. Soundness: $C^P(x) \in \{L(x), \perp\}$ with high probability.

As discussed earlier, based on [FF93], Bogdanov and Trevisan [BT06b] show that the average-case hardness of a problem in distNP cannot be based on the worst-case hardness of an NP-complete problem via non-adaptive reductions (unless the polynomial hierarchy collapses). In particular, this implies that SAT does not have a non-adaptive self-corrector (unless the polynomial hierarchy collapses). It is an important open question if the same or a similar result can be proved for adaptive reductions.

Watson [Wat12] shows that there exists an oracle O such that there is no worst-case to average-case reduction for NP relative to O. Impagliazzo [Imp11] then gives the following more general result: any proof that gives a positive answer to Question 2 must use non-relativizing techniques. More precisely, it is shown that there exists an oracle O such that $\mathsf{NP}^O \not\subseteq \mathsf{BPP}^O$, and there is no hard problem in distNP^O. Note that this does not rule out the existence of a worst-case to average-case reduction, as such reductions do not necessarily relativize. In particular, the result of Bogdanov and Trevisan [BT06b] also applies to reductions that are non-adaptive and do *not* relativize: there is no such reduction, unless the polynomial hierarchy collapses.

Negative results on Question 1. This question goes back to the work of Diffie and Hellman [DH76]. Even and Yacobi [EY80] give a cryptosystem that is NP-hard to break. However, their notion of security requires that the adversary can break the system in the worst-case (i.e. for every key). Their cryptosystem can in fact be broken on most keys, as shown by Lempel [Lem79]. It is now understood that breaking a cryptosystem should be hard on average, which is, for example, reflected in the definition of one-way functions.

Brassard [Bra83] shows that public-key encryption cannot be based on NP-hardness in the following sense: under certain assumptions on the scheme, if breaking the encryption can be reduced to deciding L, then $L \in \mathsf{NP} \cap \mathsf{coNP}$. In particular, if L is NP-hard this implies that $\mathsf{NP} = \mathsf{coNP}$. Goldreich and Goldwasser [GG98] show the same result under relaxed assumptions.

To give a positive answer to Question 1, one can aim for a reduction from an NP-complete problem to inverting a one-way function well on average (see for example [AGGM06] for a formal definition). As discussed earlier, the work of Impagliazzo and Levin [IL90] and Ben-David et al. [BDCGL92] allows to translate the results of [FF93] and [BT06b] to this setting. That is, there is no non-adaptive reduction from an NP-complete problem L to inverting a one-way function, unless the polynomial hierarchy collapses to the third level. Akavia et al. [AGGM06] directly use the additional structure of the one-way function to prove that the same assumption allows the stronger conclusion $\mathsf{coNP} \subseteq \mathsf{AM}$, which implies a collapse of the polynomial hierarchy to the second level.

Haitner et al. [HMX10] show that if constant-round statistically hiding commitment can be based on an NP-complete problem via $O(1)$-adaptive reductions (i.e. the reduction makes a constant number of query rounds), then $\mathsf{coNP} \subseteq \mathsf{AM}$, and the polynomial hierarchy collapses to the second level. In fact, they obtain the same conclusion for any cryptographic primitive that can be broken by a constant-depth collision finding oracle (such as variants of collision resistant hash functions and oblivious transfer). They also obtain non-trivial, but weaker consequences for $\text{poly}(n)$-adaptive reductions.

Bogdanov and Lee [BL13] explore the plausibility of basing homomorphic encryption on NP-hardness. They show that if there is a (randomized, adaptive) reduction from some L to breaking a homomorphic bit encryption scheme (that supports the evaluation of any sufficiently "sensitive" collection of functions), then $L \in \mathsf{AM} \cap \mathsf{coAM}$. In particular, if L is NP-complete this implies a collapse of the polynomial hierarchy to the second level.

Positive results. We only know few problems in distNP that have worst-case to average-case reductions where the worst-case problem is believed to be hard. Most such problems are based on lattices, and the most important two are the short integer solution problem (SIS), and the learning with errors problem (LWE).

The SIS problem goes back to the breakthrough work of Ajtai [Ajt96]. He gives a reduction from an approximate worst-case version of the shortest vector problem to an average-case version of the same problem, and his results were subsequently improved [Mic04, MR07]. Many cryptographic primitives, such as one-way functions, collision-resistant hash functions, identification schemes, and digital signatures have been based on the SIS problem, and we refer to [BLP+13] for an overview. He gives a reduction from an approximate worst-case version of the shortest vector problem to an average-case version of the same problem, and his results were subsequently improved [Mic04, MR07]. Many cryptographic primitives, such as one-way functions, collision-resistant hash functions, identification schemes, and digital signatures have been based on the SIS problem, and we refer to [BLP+13] for an overview. He gives a reduction from an approximate worst-case version of the shortest vector problem to an average-case version of the same problem, and his results were subsequently improved [Mic04, MR07]. Many cryptographic primitives, such as one-way functions, collision-resistant hash functions, identification schemes, and digital signatures have been based on the SIS problem, and we refer to [BLP+13] for an overview.

Regev [Reg09] gives a worst- to average-case reduction for the LWE problem in the quantum setting. That is, an algorithm for solving LWE implies the existence of a quantum algorithm to solve the lattice problem. The work of Peikert [Pei09] and Lyubashevsky and Micciancio [LM09] makes progress towards getting a reduction that yields a classical worst-case algorithm. The first classical hardness reduction for LWE (with polynomial modulus) is then given by Brakerski et al. [BLP+13]. A large number of cryptographic schemes are based on LWE, and we refer to Regev's survey [Reg10], and to [BLP+13] for an overview.

Unfortunately, for all lattice-based worst-case to average-case reductions, the worst-case problem one reduces to is contained in $\mathsf{NP} \cap \mathsf{coNP}$, and thus unlikely to be NP-hard. We note that several of these reductions (such as the ones of [Ajt96, Mic04, MR07]) are adaptive.

Gutfreund et al. [GSTS07] make progress towards a positive answer to Question 2: they give a worst-case to average-case reduction for NP, but sampling an input from the distribution they give requires quasi-polynomial time. Furthermore, for any fixed BPP algorithm that tries to decide SAT, they give a distribution that is hard for that specific algorithm. Note that this latter statement does not give a polynomial time samplable distribution that is hard for *any* algorithm. Unlike in [FF93, BT06b], where the reductions under consideration get black-box access to the average-case oracle, the reduction given by [GSTS07] is not black-box, i.e. it requires access to the code of an efficient average-case algorithm. Such reductions (even non-adaptive ones) are not ruled out by the results of [FF93, BT06b]. Gutfreund and Ta-Shma [GTS07] show that even though the techniques of [GSTS07] do not yield an average-case hard problem in distNP, they bypass the negative results of [BT06b]. Furthermore, under a certain derandomization assumption for BPP, they give a worst-case to average-case reduction from NP to an average-case hard problem in $\mathsf{NTIME}(n^{O(\log n)})$.

2 Preliminaries

2.1 Notation

We denote sets using calligraphic letters $\mathcal{A}, \mathcal{B}, \ldots$, and we write capital letters $A, B, \ldots$ to denote random variables. For a set $\mathcal{S}$, we use $x \leftarrow \mathcal{S}$ to denote that x is chosen uniformly from $\mathcal{S}$. We denote probability distributions on bitstrings by P, and write $x \leftarrow \mathsf{P}$ if x is chosen from P. Also, we let $\mathsf{P}(x) := \Pr_{y \leftarrow \mathsf{P}}[x = y]$. For $n \in \mathbb{N}$ we let $(n) := \{0, 1, \ldots, n\}$ and $[n] := \{1, 2, \ldots, n\}$.

2.2 Concentration Bounds

We use several concentration bounds and first state the well-known Chernoff bound.

Lemma 2.1 (Chernoff bound)**.** *Let $X_1, \ldots, X_k$ be independent random variables where for all i we have $X_i \in \{0,1\}$ and $\Pr[X_i = 1] = p$ for some $p \in (0,1)$. Define $\widetilde{X} := \frac{1}{k}\sum_{i\in[k]} X_i$. Then for any $\varepsilon > 0$ it holds that*

$$\Pr_{X_1,\ldots,X_k}\left[\widetilde{X} \geq p + \varepsilon\right] < \exp\left(-\frac{\varepsilon^2 k}{2}\right), \quad \Pr_{X_1,\ldots,X_k}\left[\widetilde{X} \leq p - \varepsilon\right] < \exp\left(-\frac{\varepsilon^2 k}{2}\right).$$

Hoeffding's bound [Hoe63] states that for k independent random variables $X_1, \ldots, X_k$ that take values in some appropriate range, with high probability their sum is close to its expectation.

Lemma 2.2 (Hoeffding's inequality)**.** *Let $X_1, \ldots, X_k$ be independent random variables with $X_i \in [a,b]$, define $\widetilde{X} := \frac{1}{k}\sum_{i\in[k]} X_i$ and let $p = \mathrm{E}_{X_1,\ldots,X_k}[\widetilde{X}]$. Then for any $\varepsilon > 0$ we have*

$$\Pr_{X_1,\ldots,X_k}\left[\widetilde{X} \geq p + \varepsilon\right] \leq \exp\left(-\frac{2\varepsilon^2 k}{(b-a)^2}\right),$$
$$\Pr_{X_1,\ldots,X_k}\left[\widetilde{X} \leq p - \varepsilon\right] \leq \exp\left(-\frac{2\varepsilon^2 k}{(b-a)^2}\right).$$

2.3 Interactive Proofs

In an interactive proof, an all-powerful prover tries to convince a computationally bounded verifier that her claim is true. The notion of an interactive protocol formalizes the interaction between the prover and the verifier, and is defined as follows.

Definition 2.3 (Interactive Protocol)**.** Let $n \in \mathbb{N}$, $V : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \cup \{\text{accept}, \text{reject}\}$, $P : \{0,1\}^* \to \{0,1\}^*$, and $k, \ell, m : \mathbb{N} \to \mathbb{N}$. A k-round interactive protocol (V, P) with message length m and ℓ random coins between V and P on input $x \in \{0,1\}^n$ is defined as follows:

1. Uniformly choose random coins $r \in \{0,1\}^{\ell(n)}$ for V.
2. Let $k := k(n)$ and repeat the following for $i = 0, 1, \ldots, k-1$:
   a) $m_i := V(x, i, r, a_0, \ldots, a_{i-1}), m_i \in \{0,1\}^{m(n)}$
   b) $a_i := P(x, i, m_0, \ldots, m_i), a_i \in \{0,1\}^{m(n)}$

   Finally, we have $V(x, k, r, m_0, a_0, \ldots, m_{k-1}, a_{k-1}) \in \{\text{accept}, \text{reject}\}$.

We denote by $(V(r), P)(x) \in \{\text{accept}, \text{reject}\}$ the output of V on random coins r after an interaction with P. We say that $(x, r, m_0, a_0, \ldots, m_j, a_j)$ is consistent for V if for all $i \in (k-1)$ we have $V(x, i, r, a_0, \ldots, a_{i-1}) = m_i$. Finally, if $(x, r, m_0, a_0, \ldots, m_{k-1}, a_{k-1})$ is not consistent for V, then $V(x, k, r, m_0, a_0, \ldots, m_{k-1}, a_{k-1}) = \text{reject}$. $\diamondsuit$

We now define the classes IP and AM of interactive proofs. The definition of IP was initially given by Goldwasser, Micali, and Rackoff [GMR89], and the definition of AM goes back to Babai [Bab85].

Definition 2.4 (IP, AM, and AM/poly)**.** The set

$$\mathsf{IP}\begin{pmatrix} \mathsf{rounds} & = & k(n) \\ \mathsf{time} & = & t(n) \\ \mathsf{msg\ size} & = & m(n) \\ \mathsf{coins} & = & \ell(n) \\ \mathsf{compl} & \geq & c(n) \\ \mathsf{sound} & \leq & s(n) \end{pmatrix}$$

contains the languages L that admit a k-round interactive protocol (V, P) with message length m and ℓ random coins, and the following properties:

Efficiency: V can be computed by an algorithm such that for any $x \in \{0,1\}^*$ and P^* the total running time of V in $(V, P^*)(x)$ is at most $t(|x|)$.

Completeness:

$$x \in L \implies \Pr_{r \leftarrow \{0,1\}^{\ell(|x|)}}[(V(r), P)(x) = \text{accept}] \geq c(|x|).$$

Soundness: For any P^* we have

$$x \notin L \implies \Pr_{r \leftarrow \{0,1\}^{\ell(|x|)}}[(V(r), P^*)(x) = \text{accept}] \leq s(|x|).$$

The set AM is defined analogously, with the additional restriction that (V, P) is *public-coin*, i.e. for all i, m_i is an independent uniform random string. We sometimes omit the $\mathsf{msg\ size}$ and coins parameters from the notation, in which case they are defined to be at most time. If we omit the time parameter, it is defined to be $\text{poly}(n)$. We then let

$$\mathsf{IP} := \mathsf{IP}\begin{pmatrix} \mathsf{rounds} & = & \text{poly}(n) \\ \mathsf{compl} & \geq & 2/3 \\ \mathsf{sound} & \leq & 1/3 \end{pmatrix}, \quad \mathsf{AM} := \mathsf{AM}\begin{pmatrix} \mathsf{rounds} & = & 1 \\ \mathsf{compl} & \geq & 2/3 \\ \mathsf{sound} & \leq & 1/3 \end{pmatrix}$$

The set AM/poly is defined like AM, but the verifier is additionally allowed to use $\text{poly}(n)$ bits of non-uniform advice.[2] $\Diamond$

Instead of writing (for example) $L \in \mathsf{AM}(\mathsf{rounds} = k, \mathsf{time} = t, \mathsf{compl} \geq c, \mathsf{sound} \leq s)$, we sometimes say that L has a k-round public-coin interactive proof with completeness c and soundness s, where the verifier runs in time t.

Babai and Moran [BM88] showed that in the definition of AM above, setting $\mathsf{rounds} = k$ for any constant $k \geq 1$ yields the same class. The same is true for AM/poly, and it thus suffices to give a k-round protocol with advice for some constant k to place a language in AM/poly.

Assuming deterministic provers. For proving the soundness condition of an interactive proof, without loss of generalty we may assume that the prover is determinsitic: we consider the deterministic prover that always sends the answer which maximizes the verifier's acceptance probability. No probabilistic prover can achieve better acceptance probability.

[2] For a definition of turing machines with advice we refer, for example, to Arora and Barak's book [AB09], Chapter 6.3.

Interactive proofs for promise problems. *Promise problems* generalize the notion of languages, and are defined as follows: a promise problem $\Pi = (\Pi_Y, \Pi_N)$ is a pair of sets $\Pi_Y, \Pi_N \subseteq \{0,1\}^*$ such that $\Pi_Y \cap \Pi_N = \emptyset$. Given a problem Π, we are interested in algorithms (or protocols) that accept instances in Π_Y and reject instances in Π_N. In particular, we don't care about the algorithm's behavior on instances that are not in $\Pi_Y \cup \Pi_N$.

Promise versions of the classes IP, AM, and $\mathsf{AM/poly}$ are defined in the obvious way by restricting the completeness condition to $x \in \Pi_Y$ and the soundness condition to $x \in \Pi_N$.

2.4 Histograms and the First Wasserstein Distance

We give the definitions of histograms and Wasserstein distance as given in [HMX10]. The histogram of a probability distribution P is a function $h : [0,1] \to [0,1]$ such that $h(p) = \Pr_{x \leftarrow \mathsf{P}}[\mathsf{P}(x) = p]$. The following definition describes a discretized version of this concept.

Definition 2.5 ((ε, t)-histogram)**.** Let P be a probability distribution on $\{0,1\}^n$, fix $t \in \mathbb{N}$, and let $\varepsilon > 0$. For $i \in (t)$ we define the i'th interval $\mathcal{A}_i$ and the i'th bucket $\mathcal{B}_i$ as

$$\mathcal{A}_i := \left(2^{-(i+1)\varepsilon}, 2^{-i\varepsilon}\right], \qquad \mathcal{B}_i := \{x : \mathsf{P}(x) \in \mathcal{A}_i\}.$$

We then let $h := (h_0, \ldots, h_t)$ where $h_i := \Pr_{x \leftarrow \mathsf{P}}[x \in \mathcal{B}_i] = \sum_{x \in \mathcal{B}_i} \mathsf{P}(x)$. The tuple h is called the (ε, t)-histogram of P. ◊

If for all x we have either $\mathsf{P}(x) = 0$ or $\mathsf{P}(x) \geq 2^{-n}$, and we consider the (ε, t)-histogram of P for $t = \lceil n/\varepsilon \rceil$, then $\bigcup_{i \in (t)} \mathcal{B}_i = \{0,1\}^n$ and $\sum_{i \in (t)} h_i = 1$. If smaller probabilities occur (e.g. $\mathsf{P}(x) = 2^{-2n}$ for some x), this sum is smaller than 1.

The following observation follows directly from the above definition:

Claim 2.6. *For all $i \in (t)$ we have $h_i 2^{i\varepsilon} \leq |\mathcal{B}_i| \leq h_i 2^{(i+1)\varepsilon}$.*

Proof. Recall that $h_i = \sum_{x \in \mathcal{B}_i} \mathsf{P}(x)$ and by definition of $\mathcal{B}_i$ we have

$$|\mathcal{B}_i| 2^{-(i+1)\varepsilon} = \sum_{x \in \mathcal{B}_i} 2^{-(i+1)\varepsilon} \leq \sum_{x \in \mathcal{B}_i} \mathsf{P}(x) \leq \sum_{x \in \mathcal{B}_i} 2^{-i\varepsilon} = |\mathcal{B}_i| 2^{-i\varepsilon}. \quad \square$$

We next introduce the Wasserstein distance between histograms. Intuitively, it measures how much work it takes to turn one histogram into another one, where the work is defined as the mass that is moved times the distance over which it is moved. We will only apply the Wasserstein distance to histograms h where $\sum_{i \in (t)} h_i = 1$, and we call such tuples *distribution vectors.*

Definition 2.7 (1st Wasserstein distance over arrays)**.** Given two distribution vectors x and y over (t) we let $a_i = \sum_{j \in (i)} x_j$ and $b_i = \sum_{j \in (i)} y_j$. We let

$$\overrightarrow{\mathrm{W1}}(x, y) := \frac{1}{t} \sum_{i \in (t): a_i > b_i} (a_i - b_i), \qquad \overleftarrow{\mathrm{W1}}(x, y) := \frac{1}{t} \sum_{i \in (t): b_i > a_i} (b_i - a_i),$$

and $\mathrm{W1}(x, y) := \overrightarrow{\mathrm{W1}}(x, y) + \overleftarrow{\mathrm{W1}}(x, y)$. $\mathrm{W1}(x, y)$ is called the 1st Wasserstein distance between x and y. $\overrightarrow{\mathrm{W1}}(x, y)$ and $\overleftarrow{\mathrm{W1}}(x, y)$ are called the right and left Wasserstein distance, respectively. ◊

In more general settings, this distance is also called Kantorovich distance, or Earth Mover's distance. For a more detailed discussion of this concept and the associated intuition we refer to the nice exposition in [HMX10].

2.5 The Parallel Lower Bound and Histogram Verification Protocols

To formalize the guarantees of the two protocols, we use the notion of promise problems as introduced in Section 2.3.

The lower bound protocol. We describe the promise problem that is solved by the parallel lower bound protocol as stated in [BT06b] (Corollary 7), which is based on the lower bound protocol of [GS86]. The goal is to prove approximate lower bounds on the size of a set $\mathcal{S}$ which is specified using a circuit C as $\mathcal{S} := C^{-1}(1) = \{x : C(x) = 1\}$. More generally, the following lemma states that there is a protocol that allows to prove lower bounds in parallel for several sets $\mathcal{S}_i = C^{-1}(y_i) = \{x : C(x) = y_i\}$ for some given bit strings y_i. In the following, for a circuit C we denote its size by $\mathrm{Size}(C)$.

Lemma 2.8 (Parallel Lower Bound Protocol, [BT06b]). *For circuits $C : \{0,1\}^n \to \{0,1\}^m$, $\varepsilon \in (0,1)$ we define the promise problem Π^{LB} as*

$$\Pi_Y^{LB} := \big\{(C, \varepsilon, y_1, s_1, \dots, y_k, s_k) : \forall i \in [k] : |C^{-1}(y_i)| \geq s_i\big\}$$
$$\Pi_N^{LB} := \big\{(C, \varepsilon, y_1, s_1, \dots, y_k, s_k) : \exists i \in [k] : |C^{-1}(y_i)| \leq (1-\varepsilon)s_i\big\}$$

There exists a constant-round public-coin interactive proof for Π^{LB} with completeness $1-\varepsilon$ and soundness ε, where the verifier runs in time $\mathrm{poly}(\frac{\mathrm{Size}(C)k}{\varepsilon})$.

We briefly sketch how such lower bounds can be proved, but refer to [BT06b] for a detailed exposition and a proof of the above lemma. Consider the case $k = 1$, suppose the input (C, ε, s) is given, and we would like to give a protocol such that the verifier accepts with high probability if $|C^{-1}(1)| \geq s$ and rejects with high probability if $|C^{-1}(1)| \leq (1-\varepsilon)s$. The protocol can be based on the hash mixing lemma:

Lemma 2.9 (Hash Mixing Lemma, [Nis92]). *Let $\mathcal{B} \subseteq \{0,1\}^n$, $x \in \{0,1\}^n$. If $\mathcal{H}(n,m)$ is a family of 2-wise independent hash functions mapping n bits to m bits, then the following holds. For all $\gamma > 0$ we have*

$$\Pr_{h \leftarrow \mathcal{H}(n,m)}\left[|\{y \in \mathcal{B} : h(y) = 0^m\}| \notin (1 \pm \gamma)\frac{|\mathcal{B}|}{2^m}\right] \leq \begin{cases} \frac{2^m}{\gamma^2 |\mathcal{B}|} & \text{if } |\mathcal{B}| > 0 \\ 0 & \text{if } |\mathcal{B}| = 0. \end{cases}$$

The idea is to let the verifier choose a pairwise independent hash function with an appropriate range, such that for the set $\mathcal{B} := C^{-1}(1)$ in case $|\mathcal{B}| \geq s$, with high probability the set $\mathcal{M} := \{x : x \in \mathcal{B} \wedge f(x) = 0\}$ has size at least some fixed polynomial $p(n)$. One chooses the parameters such that in case $|\mathcal{B}| \leq (1-\varepsilon)s$, we have $|\mathcal{M}| < p(n)$ with high probability. Then, the prover is supposed to send $p(n)$ many elements $x_1, \dots, x_p$ to the verifier that satisfy $C(x_i) = 1$ and $f(x_i) = 0$. Finally, the verifier checks that the prover sent $p(n)$ many elements with these properties. It is not hard to see completeness and soundness, and Lemma 2.8 can be proved for the parallel repetition of this protocol.

Verifying histograms. We consider circuits $C : \{0,1\}^n \to \{0,1\}^m$, and the distribution P^C defined by $\mathsf{P}^C(y) = \Pr_{r \leftarrow \{0,1\}^n}[C(r) = y]$. The VerifyHist protocol of [HMX10] (Lemma 4.4) allows to verify that some given histogram h is close to the histogram of P^C in terms of the Wasserstein distance.

Lemma 2.10 (VerifyHist Protocol, [HMX10])**.** *For a circuit $C : \{0,1\}^n \to \{0,1\}^m$, $\varepsilon \in (0,1)$, and $t = \lceil n/\varepsilon \rceil$, we denote by $h^C \in [0,1]^{t+1}$ the (ε, t)-histogram of P^C. We define the promise problem $\Pi^{\mathit{VerifyHist}}$ as*

$$\Pi_Y^{\mathit{VerifyHist}} := \left\{(C, \varepsilon, h) : h = h^C\right\}$$
$$\Pi_N^{\mathit{VerifyHist}} := \left\{(C, \varepsilon, h) : \mathit{W1}(h^C, h) > 20/t\right\}$$

There exists a constant-round public-coin interactive proof for $\Pi^{\mathit{VerifyHist}}$ with completeness $1 - 2^{-n}$ and soundness 2^{-n}, where the verifier runs in time $\mathrm{poly}(\frac{\mathrm{Size}(C)}{\varepsilon})$.

We remark that $\mathrm{W1}(h^C, h)$ is well-defined: for all y we have $\mathsf{P}^C(y) = 0$ or $\mathsf{P}^C(y) \geq 2^{-n}$, and thus by choice of t, h^C is a distribution vector.

We give a brief and intuitive description of the VerifyHist protocol, but refer to [HMX10] for a formal treatment and a proof of the above lemma. For a circuit C and a claimed histogram h the protocol proceeds as follows.

The first part of the protocol is called *preimage test*: the verifier samples elements $y_1, \ldots, y_k$ (for some appropriate k) from the distribution P^C and sends them to the prover. The honest prover sends back the probabilities $\mathsf{P}^C(y_i)$, and proves a lower bound on them using the parallel lower bound protocol of Lemma 2.8. Finally, the verifier considers the histogram h' induced by the values $\mathsf{P}^C(y_i)$ and accepts if and only if $\mathrm{W1}(h, h')$ is small.

In the second part, a so-called *image test* is performed: let $\mathcal{W}_i := \{y : \mathsf{P}^C(y) \geq 2^{-i\varepsilon}\}$, and let w_i^h be the estimates of $\mathcal{W}_i$ given by the claimed histogram h. Using the parallel lower bound protocol, the prover proves that indeed $|\mathcal{W}_i| \geq w_i^h$ for all i.

Intuitively, the preimage test prevents the prover from claiming that many probabilities are larger than they actually are, and it can be shown that the image test rejects in case the probabilities are larger than claimed. Haitner et al. [HMX10] prove that indeed, if both tests accept, then h is close to h^C in the first Wasserstein distance.

2.6 Worst-Case to Average-Case Reductions

We give a definition of non-adaptive worst-case to average-case reductions. Informally, such a reduction is a polynomial time algorithm that, given an oracle which solves some given problem on average, solves some other problem in the worst case. The reduction is called *non-adaptive* if it generates all its oracle queries before calling the oracle. The definition we give is from [BT06b].

A distributional problem is a pair $(L, \mathcal{D})$, where L is a language and $\mathcal{D}$ is a set $\mathcal{D} = \{\mathsf{P}_n\}_{n \in \mathbb{N}}$, and for each n, P_n is a distribution over $\{0,1\}^n$.

Definition 2.11. A non-adaptive δ-worst-to-average reduction from L to a distributional problem $(L', \mathcal{D})$ is a family of polynomial size circuits $\{R_n\}_{n \in \mathbb{N}}$ such that for any n the following holds:

- R_n takes as input some $x \in \{0,1\}^n$ and randomness r, and outputs $(y_1, \ldots, y_k)$ (called queries), and a circuit C.
- For any $x \in \{0,1\}^n$, and any oracle O for which $\Pr_{x \leftarrow \mathsf{P}_n, O}[O(x) \neq L'(x)] \leq \delta(n)$ we have

$$\Pr_{r, (y_1, \ldots, y_k, C) := R_n(x, r)}[C(O(y_1), \ldots, O(y_k)) = L(x)] \geq 2/3.$$

$\Diamond$

We may assume that the queries $y_1, \ldots, y_k$ are identically (but not necessarily independently) distributed. If this is not the case for the original reduction R, we can easily obtain a reduction R' that satisfies this property: R' obtains the queries of R and outputs a random permutation of them (the circuit C is also modified accordingly).

Furthermore, the constant $2/3$ can be replaced by $1/2 + 1/n^c$ for some constant c: by the usual repetition argument, executing the reduction a polynomial number of times and outputting the majority answer still yields an exponentially small error probability.

3 Technical Overview

For a formal definition of non-adaptive worst-case to average-case reductions, we refer to Section 2.6 in the preliminaries. In the introduction we stated an informal version of the result of [BT06b]. We now state their main theorem formally. Let $\mathcal{U}$ be the set $\{\mathsf{P}_n\}_{n\in\mathbb{N}}$ where P_n is the uniform distribution on $\{0,1\}^n$.

Theorem 3.1 (Main Theorem of [BT06b])**.** *For any L and L' and every constant c the following holds. If L is NP-hard, $L' \in \mathsf{NP}$, and there exists a non-adaptive $1/n^c$-worst-to-average reduction from L to $(L', \mathcal{U})$, then $\mathsf{coNP} \subseteq \mathsf{NP}/\mathrm{poly}$.*

As discussed earlier, the conclusion implies a collapse of the polynomial hierarchy to the third level.

The theorem is stated for the set of uniform distributions $\mathcal{U}$. Using the results of Ben-David et al. [BDCGL92] and Impagliazzo and Levin [IL90], the theorem can be shown to hold for any polynomial time samplable set of distributions $\mathcal{D}$. This is nicely explained in [BT06b] (Section 5).

We first give an overview of the original proof, and then describe how our new protocols fit in.

3.1 The proof of Bogdanov and Trevisan

Suppose R reduces the NP-complete language L to $(L', \mathcal{U}) \in \mathsf{distNP}$. The goal is to give a (constant-round) $\mathsf{AM}/\mathrm{poly}$ protocol for L and its complement. As $\mathsf{NP}/\mathrm{poly} = \mathsf{AM}/\mathrm{poly}$, this will give the result. The idea is to simulate an execution of the reduction R on input x with the help of the prover. The verifier then uses the output of R as its guess for $L(x)$. R takes as input the instance x, randomness $r \in \{0,1\}^n$, and produces (non-adaptively) queries $y_1, \ldots, y_k \in \{0,1\}^m$ for the average-case oracle. The reduction is guaranteed to guess $L(x)$ correctly with high probability, provided the oracle answers are correct with high probability. As mentioned in Section 2.6, we may assume that the queries $y_1, \ldots, y_k$ are identically (but not necessarily independently) distributed. We denote the resulting distribution of individual queries by $\mathsf{P}^{R,x}$, i.e. $\mathsf{P}^{R,x}(y) = \Pr_r[R(x,r) = y]$ (where $R(x,r)$ simply outputs the first query of the reduction on randomness r).

Handling uniform queries: the Feigenbaum-Fortnow protocol. The proof of [BT06b] relies on the following protocol by Feigenbaum and Fortnow [FF93]. The protocol assumes that the queries are uniformly distributed, i.e. $\mathsf{P}^{R,x}(y) = 2^{-m}$ for all y. The advice for the $\mathsf{AM}/\mathrm{poly}$ protocol is $g_{\mathrm{UY}} = \Pr_{y \leftarrow \{0,1\}^m}[y \in L']$, i.e. the probability of a uniform sample being a yes-instance. The protocol proceeds as follows. First, the verifier chooses random strings $r_1, \ldots, r_\ell$ and sends them to the prover. The honest prover defines $(y_{i1}, \ldots, y_{ik}) := R(x, r_i)$ for all i, and indicates to the verifier which y_{ij} are in L' (we call them yes-instances), and provides the corresponding NP-witnesses. The verifier checks the witnesses, expects to see approximately a g_{UY} fraction of yes-answers, and rejects if this is not the case. The verifier then chooses a random i and outputs $R(x, r_i, y_{i1}, \ldots, y_{ik})$ as its guess for $L(x)$.

To see completeness, one uses a concentration bound to show that the fraction of yes-answers sent by the prover is approximately correct with high probability (one must be careful at this point, because the outputs of the reduction for a fixed r_i are not independent). Finally, the reduction decides $L(x)$ correctly with high probability.

To argue that the protocol is sound, we note that the prover cannot increase the number of yes-answers at all, as it must provide correct witnesses. Furthermore, the prover cannot decrease the number of yes-answers too much, as the verifier wants to see approximately a g_{UY} fraction. This gives that most answers provided by the prover are correct, and thus with high probability the reduction gets good oracle answers, in which case it outputs 0 with high probability.

We note that the Feigenbaum-Fortnow simulation protocol is public-coin.

The case of smooth distributions: the Hiding Protocol. Bogdanov and Trevisan [BT06b] generalize the above protocol so that it works for distributions that are α-smooth, i.e. where $\mathsf{P}^{R,x}(y) \leq \alpha 2^{-m}$ for all y and some threshold parameter $\alpha = \mathrm{poly}(n)$ (we say all samples are α-light). If the verifier knew the probability $g_{\mathrm{Y}} := \Pr_{y \leftarrow \mathsf{P}^{R,x}}[y \in L']$, it is easy to see that the Feigenbaum-Fortnow protocol (using g_{Y} instead of g_{UY} as above) can be used to simulate the reduction. Unfortunately, g_{Y} cannot be handed to the verifier as advice, as it may depend on the instance x. Thus, [BT06b] give a protocol, named the *Hiding Protocol*, that allows the verifier to obtain an approximation of g_{Y}, given g_{UY} as advice.

The idea of the protocol is as follows: the verifier hides a $1/\alpha$-fraction of samples from $\mathsf{P}^{R,x}$ among uniform random samples (i.e. it permutes all samples randomly). The honest prover again indicates the yes-instances and provides witnesses for them. The verifier checks the witnesses and that the fraction of yes-answers among the uniform samples is approximately g_{UY}. If this is true, it uses the fraction of yes-answers among the samples from $\mathsf{P}^{R,x}$ as an approximation of g_{Y}.

Completeness follows easily. The intuition to see soundness is that since the distribution is α-smooth, and as the verifier hides only a $1/\alpha$ fraction of $\mathsf{P}^{R,x}$ samples among the uniform ones, the prover cannot distinguish them.

We note that the intuition behind this protocol crucially relies on the fact that the verifier can keep some of its random coins private: the prover is not allowed to know where the distribution samples are hidden.

General distributions and the Heavy Samples Protocol. Finally, [BT06b] remove the restriction that $\mathsf{P}^{R,x}$ is α-smooth as follows. We say y is α-heavy if $\mathsf{P}^{R,x}(y) \geq \alpha 2^{-m}$, and let $g_{\mathrm{H}} := \Pr_{y \leftarrow \mathsf{P}^{R,x}}[\mathsf{P}^{R,x}(y) \geq \alpha 2^{-m}]$ be the probability of a distribution sample being heavy, and $g_{\mathrm{YL}} := \Pr_{y \leftarrow \mathsf{P}^{R,x}}[y \in L' \wedge \mathsf{P}^{R,x}(y) < \alpha 2^{-m}]$ the probability of a distribution sample being a yes-instance and light.

We first note that if the verifier knows (an approximation of) both g_{H} and g_{YL}, it can use the Feigenbaum-Fortnow approach to simulate the reduction: the verifier simply uses g_{YL} instead of g_{UY} in the protocol, and ignores the heavy samples. It can do this by having the prover indicate the α-heavy instances, and checking that their fraction is close to g_{H}. Using the lower bound protocol of Goldwasser and Sipser [GS86] (see Section 2.5), the prover must prove that these samples are indeed heavy. Finally, for the heavy samples the verifier can simply set the oracle answers to 0: this changes the oracle answers on at most a polynomially small (i.e. a $1/\alpha$) fraction of the inputs, as by definition at most a $1/\alpha$ fraction of the y's can be α-heavy. Completeness is not hard to see, and soundness follows because a cheating prover cannot claim light samples to be heavy (by the soundness of the lower bound protocol), and thus, by the verifier's check, cannot lie much about which samples are heavy.

If the verifier knows (an approximation of) g_{H}, then it can use the hiding protocol to approximate g_{YL}: the verifier simply ignores the heavy samples. This is again done by having the prover additionally tell which samples are α-heavy (and prove this fact using the lower bound protocol). The verifier additionally checks that the fraction of heavy samples among the distribution samples is approximately g_{H}, and finally uses the fraction of light distribution samples as approximation for g_{YL}.

It only remains to approximate g_{H}. This is done using the *Heavy Samples Protocol* as follows: the verifier samples $y_1, \ldots, y_k$ from $\mathsf{P}^{R,x}$ by choosing random $r_1, \ldots, r_k$ and letting $y_i := R(x, r_i)$. It sends the y_i to the prover. The honest prover indicates which of them are heavy, and proves to the verifier using the lower bound protocol of [GS86] that the heavy samples are indeed heavy and using the upper bound protocol of Aiello and Håstad [AH91] that the light samples are indeed light. The verifier then uses the fraction of heavy samples as its approximation for g_{H}. It is intuitive that this protocol is complete and sound. The upper bound protocol requires that the verifier knows a uniform random element (which is unknown to the prover) in the set on which the upper bound is proved. In our case, the verifier indeed knows the value r_i, which satisfies this condition.

We note that this protocol relies on private-coins, as the verifier must keep the r_i secret for the upper bound proofs.

3.2 Our Proof

We give two new protocols to approximate the probabilities g_{H} and g_{YL}, as defined in the previous section. These protocols can be used to replace the Hiding Protocol and the Heavy Samples Protocol of [BT06b], respectively. Together with the Feigenbaum-Fortnow based simulation protocol of [BT06b], this then yields a different proof of $\mathsf{coNP} \subseteq \mathsf{AM}/\mathrm{poly}$ under the given assumptions.

Verifying histograms. We are going to employ the VerifyHist protocol by Haitner et al. [HMX10] to verify the histogram of a probability distribution. Recall that the (ε, t)-histogram $h = (h_0, \ldots, h_t)$ of a distribution P is defined by letting $h_i := \Pr_{y \leftarrow \mathsf{P}}[y \in \mathcal{B}_i]$, where $\mathcal{B}_i := \left\{x : \mathsf{P}(x) \in (2^{-(i+1)\varepsilon}, 2^{-i\varepsilon}]\right\}$ (See Definition 2.5). We will use the VerifyHist protocol for the distribution $\mathsf{P}^{R,x}$, as defined by the reduction $R(x, \cdot)$ under consideration, i.e. $\mathsf{P}^{R,x}(y) = \Pr_r[R(x, r) = y]$. Intuitively, this protocol allows to prove that some given histogram h is close to the true histogram of $\mathsf{P}^{R,x}$ in terms of the 1st Wasserstein distance (also known as Earth Mover's distance). This distance between h and h' measures the minimal amount of work that is needed to push the configuration of earth given by h to get the configuration given by h': moving earth over a large distance is more expensive than moving it over a short distance. For formal definitions of histograms and the 1st Wasserstein distance we refer to Section 2.4.

Lemma 3.2 (VerifyHist protocol of [HMX10], informal)**.** *There is a constant-round public-coin protocol VerifyHist where the prover and the verifier get as input the circuit $R(x, \cdot)$ and a histogram h, and we have:*

Completeness: *If h is the histogram of $\mathsf{P}^{R,x}$, then the verifier accepts with high probability.*

Soundness: *If h is far from the histogram of $\mathsf{P}^{R,x}$ in the 1st Wasserstein distance, then the verifier rejects with high probability.*

The formal statement can be found in Section 2.5.

The new Heavy Samples Protocol. The idea to approximate the probability g_{H} is very simple. The honest prover sends the histogram of $\mathsf{P}^{R,x}$, and the verifier uses the VerifyHist protocol to verify it. Finally, the verifier simply reads the probability g_{H} from the histogram.

There is a technical issue that comes with this approach. For example, it may be that all y's with nonzero probability have the property that $\mathsf{P}^{R,x}(y)$ is very close, but just below $\alpha 2^{-m}$. In this case, a cheating prover can send a histogram claiming that these y's have probability slightly above this threshold. This histogram has small Wasserstein distance from the true histogram, as the probability mass is moved only over a short distance. Clearly, the verifier's guess for g_{H} is very far from the true value in this case.

We note that the same issue appears in the proof of [BT06b], and we deal with it in exactly the same way as they do: we choose the threshold α randomly, such that with high probability $\Pr_{y \leftarrow \mathsf{P}^{R,x}}[\mathsf{P}^{R,x}(y)$ is close to $\alpha 2^{-m}]$ is small (see Section 4.1 for the formal statement).

A public-coin Hiding Protocol for smooth distributions. We would like the verifier to only send uniform random samples to the prover (as opposed to the original hiding protocol, where a few samples from the distribution are hidden among uniform samples). We first describe the main idea in the special and simpler case where $\mathsf{P}^{R,x}$ is α-smooth. In this case, we can give the following protocol, which uses g_{UY} as advice:

The verifier sends uniform random samples $y_1, \ldots, y_k$. The prover indicates for each sample whether it is a yes-instance, and provides witnesses. Furthermore, the prover tells $\mathsf{P}^{R,x}(y_i)$ to the verifier, and proves a lower bound on this probability. The verifier checks the witnesses and if the fraction of yes-instances is approximately g_{UY}, and considers the histogram h induced by the probabilities $\mathsf{P}^{R,x}(y_i)$, and in particular checks if the probability mass of h is 1. Finally, the verifier considers the histogram h_Y induced by only considering the yes-instances, and uses the total mass in h_Y as its approximation of g_{YL}.

To see completeness, the crucial point is that the smoothness assumption implies that the verifier can get a good approximation of the true histogram.

Soundness follows because the prover cannot claim the probabilities to be too large (as otherwise the lower bound protocol rejects), and it cannot claim many probabilities to be too small, as otherwise the mass of h gets significantly smaller than 1. As it cannot lie much about yes-instances, this implies a good approximation of g_{YL}.

Dealing with general distributions. The above idea can be applied even to general distributions, assuming that the verifier knows the probability $g_{\mathrm{UH}} := \Pr_{y \leftarrow \{0,1\}^m}[\mathsf{P}^{R,x}(y) \geq \alpha 2^{-m}]$ of a uniform random sample being heavy. The prover still provides the same information. The verifier only considers the part of the induced histogram h below the $\alpha 2^{-m}$ threshold, and checks that the mass of h below the threshold is close to $1 - g_{\mathrm{UH}}$.

As in the heavy samples protocol, we again encounter the technical issue that many y's could have probability close to the threshold, in which case the prover can cheat. But, as discussed earlier, this situation occurs with small probability over the choice of α.

Approximating the probability of a uniform sample being heavy. Thus, it remains to give a protocol to approximate g_{UH}. We do this in exactly the same way as the Heavy Samples protocol approximates g_{H}. That is, given the histogram that was verified using VerifyHist, the verifier simply reads the approximation of g_{UH} from the histogram. The proof that this works is rather technical, as we must show that small Wasserstein distance between the true and the claimed histogram implies a small difference of the probability g_{UH} and its approximation read from the claimed histogram. We note that we include the protocol for approximating g_{UH} directly into our Heavy Samples protocol.

4 The New Protocols

We give protocols to replace the Heavy Samples Protocol and the Hiding Protocol of [BT06b]. A technical overview including proof intuitions can be found in Section 3. In this section, we give the two protocols and state the guarantees they give. The protocol analyses can be found in Sections 5 and 6.

4.1 Choosing a Random Threshold

We let $\mathcal{A}_{\alpha_0,\delta}$ be the uniform distribution on $\{\alpha_0(1+3\delta)^i : 0 \le i \le 1/\delta\}$. This distribution will be used to choose a threshold parameter α. The following claim is from [BT06b].

Claim 4.1 (Choosing a random threshold)**.** *For every $\alpha_0 > 0$ and $0 < \delta < 1/3$, and every distribution P on $\{0,1\}^m$ we have*

$$\mathop{\mathrm{E}}_{\alpha \leftarrow \mathcal{A}_{\alpha_0,\delta}}\left[\Pr_{y\leftarrow \mathsf{P}}[\mathsf{P}(y) \in (1 \pm \delta)\alpha 2^{-m}]\right] \le \delta.$$

We get that with high probability over the choice of α there is only little mass close to the threshold:

Claim 4.2. *For every distribution P on $\{0,1\}^m$ and $\varepsilon \in (0,1)$, with probability at least $1 - 20\sqrt{\varepsilon}$ over the choice of α from $\mathcal{A}_{\alpha_0,4\varepsilon}$, we have $\Pr_{y\leftarrow \mathsf{P}}[\mathsf{P}(y) \in (1 \pm 4\sqrt{\varepsilon})\alpha 2^{-m}] \le \frac{1}{5}\sqrt{\varepsilon}$.*

Proof. This follows from Claim 4.1 by applying Markov's inequality. □

4.2 Preliminaries

Since our protocols can be used to replace part of the proof of [BT06b], we mostly stick to their notation. We give a formal definition of interactive proofs, histograms, and the Wasserstein distance in Section 2. As in [BT06b], we let $\mathcal{A}_{\alpha_0,\delta}$ be the uniform distribution on $\{\alpha_0(1+3\delta)^i : 0 \le i \le 1/\delta\}$. This distribution will be used to choose a threshold parameter α, such that only little probability mass is close to the threshold (see Section 4.1). We consider circuits $C : \{0,1\}^n \to \{0,1\}^m$, and the distribution P^C defined by $\mathsf{P}^C(y) = \Pr_{r\leftarrow\{0,1\}^n}[C(r) = y]$. We use the VerifyHist protocol of [HMX10] which on input (C, ε, h) verifies that h is close to the (t, ε)-histogram of P^C (see Section 2.5). We also use the parallel lower bound protocol as stated in [BT06b], which on input $(C, \varepsilon, y_1, s_1, \dots, y_k, s_k)$ ensures that $\forall i : |C^{-1}(y_i)| \ge (1-\varepsilon)s_i$ (see Section 2.5).

We define the following probabilities. For a given threshold parameter $\alpha > 1$, a circuit $C : \{0,1\}^n \to \{0,1\}^m$ and a nondeterministic circuit[3] $V : \{0,1\}^m \times \{0,1\}^\ell \to \{0,1\}$ we let

$$\begin{aligned} g_{\mathrm{H}}^{C,\alpha} &= \Pr_{y\leftarrow \mathsf{P}^C}[\mathsf{P}^C(y) \ge \alpha 2^{-m}], & g_{\mathrm{UH}}^{C,\alpha} &= \Pr_{y\leftarrow\{0,1\}^m}[\mathsf{P}^C(y) \ge \alpha 2^{-m}], \\ g_{\mathrm{UY}}^{C,V,\alpha} &= \Pr_{y\leftarrow\{0,1\}^m}[y \in V], & g_{\mathrm{YL}}^{C,V,\alpha} &= \Pr_{y\leftarrow \mathsf{P}^C}[\mathsf{P}^C(y) < \alpha 2^{-m} \wedge y \in V]. \end{aligned}$$

In the technical overview as given above (Section 3), we considered the circuit $R(x, r)$ defined by the reduction R, which for a fixed x and randomness r outputs the first reduction query y. The

[3] A *nondeterministic circuit* V is of the form $V : \{0,1\}^m \times \{0,1\}^\ell \to \{0,1\}$, and we say $y \in \{0,1\}^m$ is accepted by V (or also $y \in V$) if there exists $w \in \{0,1\}^\ell$ such that $V(y, w) = 1$.

protocols we give in the following are supposed to get as input the circuit $C(r) := R(x, r)$ for a fixed x. Furthermore, for the new hiding protocol the input circuit V is supposed to provide an NP verifier for the language L' as described in the technical overview.

4.3 The new Heavy Samples Protocol

Given a circuit C and $\alpha > 0$, the goal of the heavy samples protocol is to estimate the probability of heavy elements. The first probability, which we denote by $g_{\mathrm{H}}^{C,\alpha}$ is the probability that an element chosen from P^C is α-heavy, i.e. satisfies $\mathsf{P}^C(y) \geq \alpha 2^{-m}$. The second probability, denoted $g_{\mathrm{UH}}^{C,\alpha}$ is the probability that a uniform random element satisfies this property.

We give a protocol for the family of promise problems $\{\Pi^{\mathrm{PubHeavy},\alpha}\}$, which is defined as follows.

$$\Pi_Y^{\mathrm{PubHeavy},\alpha} := \left\{(C, p_{\mathrm{H}}, p_{\mathrm{UH}}, \varepsilon) : p_{\mathrm{H}} = g_{\mathrm{H}}^{C,\alpha} \wedge p_{\mathrm{UH}} = g_{\mathrm{UH}}^{C,\alpha}\right\}$$
$$\Pi_N^{\mathrm{PubHeavy},\alpha} := \left\{(C, p_{\mathrm{H}}, p_{\mathrm{UH}}, \varepsilon) : p_{\mathrm{H}} \notin [g_{\mathrm{H}}^{C,\alpha} \pm \frac{4}{5}\sqrt{\varepsilon}] \vee p_{\mathrm{UH}} \notin [g_{\mathrm{UH}}^{C,\alpha} \pm 10\sqrt{\varepsilon}]\right\}$$

We assume that the input $(C, p_{\mathrm{H}}, p_{\mathrm{UH}}, \varepsilon)$ is such that $C : \{0,1\}^n \to \{0,1\}^m$ is a circuit, $p_{\mathrm{H}}, p_{\mathrm{UH}} \in [0,1]$, and $\varepsilon \in (0,1)$. The proof of the following theorem can be found in Section 5, and the protocol is stated below.

Theorem 4.3. *For every integer α_0, with probability at least $1 - 20\sqrt{\varepsilon}$ over the choice of α from $\mathcal{A}_{\alpha_0,4\varepsilon}$, the heavy samples protocol is a constant-round interactive proof for $\Pi^{PubHeavy,\alpha}$ with completeness $1 - 2^{-n}$ and soundness $1 - 2^{-n}$, where the verifier runs in time* $\mathrm{poly}(\frac{\mathrm{Size}(C)}{\varepsilon})$.

The heavy samples protocol. On input $(C, p_{\mathrm{H}}, p_{\mathrm{UH}}, \varepsilon)$:

> **Prover:** Let $t := \lfloor \frac{n}{\tilde{\varepsilon}} \rfloor$ and $\tilde{\varepsilon} := (\frac{4}{100})^2 \varepsilon^2$, and send an $(\tilde{\varepsilon}, t)$-histogram $h \in [0,1]^{t+1}$ to the verifier.
>
> If the prover is honest, it sends the $(\tilde{\varepsilon}, t)$-histogram of P^C, denoted by h^C.
>
> **Prover and Verifier:** Run the VerifyHist protocol (Lemma 2.10) on input $(C, \tilde{\varepsilon}, h)$. The verifier rejects in case that protocol rejects.
>
> **Verifier:** Let $j^* := \max\{j : 2^{-(j+1)\tilde{\varepsilon}} > \alpha 2^{-m}\}$. Accept if and only if all of the following conditions hold:
>
> (a) $$\sum_{j \in \{j^* \pm \lceil 25/\sqrt{\tilde{\varepsilon}} \rceil\}} h_j \leq \tilde{\varepsilon}^{1/4}$$ (b) $$\sum_{j \leq j^*} h_j \in [p_{\mathrm{H}} \pm \tilde{\varepsilon}^{1/4}]$$
>
> (c) $$\frac{1}{2^m} \sum_{j \leq j^*} h_j \cdot 2^{j\tilde{\varepsilon}} \in [p_{\mathrm{UH}} \pm 4\tilde{\varepsilon}^{1/4}]$$

4.4 The new Hiding Protocol

Given a circuit C, a nondeterministic circuit V, and $\alpha > 0$, the goal of the hiding protocol is as follows. Given advice $g_{\mathrm{UY}}^{C,V,\alpha}$ and approximations of the probabilities $g_{\mathrm{H}}^{C,\alpha}$ and $g_{\mathrm{UH}}^{C,\alpha}$, the protocol approximates the probability $g_{\mathrm{YL}}^{C,V,\alpha}$ that an element is a yes-instance and α-light.

We give a protocol for the family of promise problems $\{\Pi^{\text{Hide},\alpha}\}$, which is defined as follows.

$$\Pi_Y^{\text{Hide},\alpha} := \Big\{(C, V, p_{\text{H}}, p_{\text{UH}}, p_{\text{YL}}, \varepsilon) : p_{\text{H}} = g_{\text{H}}^{C,\alpha} \wedge p_{\text{UH}} = g_{\text{UH}}^{C,\alpha} \wedge p_{\text{YL}} = g_{\text{YL}}^{C,V,\alpha}\Big\}$$

$$\Pi_N^{\text{Hide},\alpha} := \Big\{(C, V, p_{\text{H}}, p_{\text{UH}}, p_{\text{YL}}, \varepsilon) : p_{\text{H}} \in [g_{\text{H}}^{C,\alpha} \pm \frac{4}{5}\sqrt{\varepsilon}] \wedge p_{\text{UH}} \in [g_{\text{UH}}^{C,\alpha} \pm 10\sqrt{\varepsilon}] \wedge p_{\text{YL}} \notin [g_{\text{YL}}^{C,V,\alpha} \pm 117\sqrt{\varepsilon}\alpha]\Big\}$$

We assume that the input $(C, V, p_{\text{H}}, p_{\text{UH}}, p_{\text{YL}}, \varepsilon)$ is such that $C : \{0,1\}^n \to \{0,1\}^m$ is a circuit, $V : \{0,1\}^m \times \{0,1\}^\ell \to \{0,1\}$ is a nondeterministic circuit, $p_{\text{H}}, p_{\text{UH}}, p_{\text{YL}} \in [0,1]$, and $\varepsilon \in (0,1)$. The proof of the following theorem can be found in Section 6, and the protocol is given below.

Theorem 4.4. *For every integer α_0, with probability at least $1 - 20\sqrt{\varepsilon}$ over the choice of α from $\mathcal{A}_{\alpha_0,4\varepsilon}$, the hiding protocol with advice $g_{UY}^{C,V,\alpha}$ is a constant-round interactive proof for $\Pi^{Hide,\alpha}$ with completeness $1 - 5\varepsilon$ and soundness 6ε, where the verifier runs in time* $\text{poly}(\frac{\text{Size}(C)+\text{Size}(V)}{\varepsilon})$.

The hiding protocol. On input $(C, V, p_{\text{H}}, p_{\text{UH}}, p_{\text{YL}}, \varepsilon)$ and advice $g_{\text{UY}}^{C,V,\alpha}$:

> **Verifier:** Let $t := \lceil \frac{n}{\varepsilon} \rceil$ and let $\mathcal{B}_i$ for $i \in (t)$ be defined as in Definition 2.5. Let $k := \ln(\frac{2}{\varepsilon})\alpha^2 \frac{9}{2\varepsilon^2}$. Choose $y_1, \ldots, y_k \leftarrow \{0,1\}^m$, and send $y_1, \ldots, y_k$ to the prover.
>
> **Prover:** Send a labeling u, a set $\mathcal{Y} \subseteq [k]$ and witnesses $(w_i)_{i \in \mathcal{Y}}$ to the verifier.
>
> If the prover is honest, it sends $\mathcal{Y} := \{i : y_i \in V\}$, and witnesses $(w_i)_{i\in\mathcal{Y}}$ such that $V(y_i, w_i) = 1$, and[4] u such that for $i \in [k]$ we have $u(i) = \begin{cases} j & \text{if } \exists j : y_i \in \mathcal{B}_j \\ \infty & \text{otherwise.} \end{cases}$
>
> **Verifier:** Let $\mathcal{L} := \left\{i : 2^{-(u(i)+1)\varepsilon} < \alpha 2^{-m}\right\}$, $\mathcal{H} := [k] \setminus \mathcal{L}$, and reject if one of the following conditions does not hold:
>
> $$\text{(a)} \quad \frac{|\mathcal{Y}|}{k} \in [g_{\text{UY}}^{C,V,\alpha} \pm \varepsilon], \qquad \text{(b)} \quad \forall i \in \mathcal{Y} : V(y_i, w_i) = 1,$$
>
> $$\text{(c)} \quad \frac{|\mathcal{H}|}{k} \in [p_{\text{UH}} \pm 3\sqrt{\varepsilon}], \qquad \text{(d)} \quad \frac{1}{k}\sum_{i\in\mathcal{L}} 2^m \cdot 2^{-u(i)\varepsilon} \in [1 - p_{\text{H}} \pm 5\sqrt{\varepsilon}],$$
>
> $$\text{(e)} \quad \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{Y}} 2^m \cdot 2^{-u(i)\varepsilon} \in [p_{\text{YL}} \pm 5\sqrt{\varepsilon}].$$
>
> **Prover and Verifier:** Run the parallel lower bound protocol (see Lemma 2.8) on input $(C, \varepsilon/2, y_1, s_1, \ldots, y_k, s_k)$, using the values $s_i = 2^m \cdot 2^{-(u(i)+1)\varepsilon}$.

5 Analysis of the New Heavy Samples Protocol

Throughout the proof, we will use $\tilde{\varepsilon} := (\frac{4}{100})^2\varepsilon^2$. Note that then

$$\Pi_N^{\text{PubHeavy},\alpha} = \Big\{(C, p_{\text{H}}, p_{\text{UH}}, \varepsilon) : p_{\text{H}} \notin [g_{\text{H}}^{C,\alpha} \pm 4\tilde{\varepsilon}^{1/4}] \vee p_{\text{UH}} \notin [g_{\text{UH}}^{C,\alpha} \pm 50\tilde{\varepsilon}^{1/4}]\Big\},$$

Also, Claim 4.2 states the following when substituting $\tilde{\varepsilon}$ for ε:

Claim 5.1. *For every distribution P on $\{0,1\}^m$ and $\varepsilon \in (0,1)$, with probability at least $1 - 100\tilde{\varepsilon}^{1/4}$ over the choice of α from $\mathcal{A}_{\alpha_0,4\varepsilon}$, we have* $\Pr_{y\leftarrow\mathsf{P}}[\mathsf{P}(y) \in (1 \pm 100\sqrt{\tilde{\varepsilon}})\alpha 2^{-m}] \le \tilde{\varepsilon}^{1/4}$.

[4]For the special symbol ∞, we use the conventions $2^{-\infty\varepsilon} = 0$, $\forall i \in \mathbb{N} : i < \infty$, and $\forall i \in \mathbb{N} : \infty + i = \infty$.

5.1 Proof of Completeness: Overview

We use the following lemma, which states that if there is only little mass around the threshold $\alpha 2^{-m}$, then the verifier's checks are indeed satisfied for the honest prover who sends h^C.

Lemma 5.2. *Suppose*

$$\Pr_{y \leftarrow \mathsf{P}^C}[\mathsf{P}^C(y) \in (1 \pm 100\sqrt{\tilde{\varepsilon}})\alpha 2^{-m}] \leq \tilde{\varepsilon}^{1/4}. \tag{1}$$

Then we have

$$\begin{aligned}
&\textit{(i)} && 2^{-(j^*-\lceil 25/\sqrt{\tilde{\varepsilon}}\rceil)\tilde{\varepsilon}} \leq 2^{28\sqrt{\tilde{\varepsilon}}}\alpha 2^{-m} \\
& && 2^{-(j^*+\lceil 25/\sqrt{\tilde{\varepsilon}}\rceil)\tilde{\varepsilon}} \geq 2^{-28\sqrt{\tilde{\varepsilon}}}\alpha 2^{-m} \\
&\textit{(ii)} && \sum_{j \in \{j^* \pm \lceil 25/\sqrt{\tilde{\varepsilon}}\rceil\}} h_j^C \leq \tilde{\varepsilon}^{1/4} \\
&\textit{(iii)} && \sum_{j \leq j^*} h_j^C \in [g_H^{C,\alpha} \pm \tilde{\varepsilon}^{1/4}] \\
&\textit{(iv)} && \frac{1}{2^m}\sum_{j \leq j^*} h_j^C \cdot 2^{j\tilde{\varepsilon}} \in [g_{UH}^{C,\alpha} \pm 4\tilde{\varepsilon}^{1/4}]
\end{aligned}$$

With this, it is straightforward to prove completeness:

Proof of completeness. With high probability there is indeed little mass around the threshold: by Claim 5.1, (1) holds with probability at least $1 - 100\tilde{\varepsilon}^{1/4} = 1 - 20\sqrt{\varepsilon}$ over the choice of α. Furthermore, by the completeness of VerifyHist, that protocol accepts the true histogram h^C with probability at least $1 - 2^{-n}$. Finally, the above lemma gives that (1) implies (a)-(c). □

It remains to prove Lemma 5.2. We focus on the interesting parts of the proof, and defer the technical details to Section 5.3.

Proof of Lemma 5.2. We defer the proofs of (i) and (ii) to Section 5.3. Part (i) is a straightforward calculation, which follows by the definition of j^*. Part (ii) then follows from part (i), since the probabilities we sum over are close to the threshold $\alpha 2^{-m}$ and we can apply (1).

Now part (iii) is easy to prove: by definition of j^* and h_j^C we have

$$g_{\mathrm{H}}^{C,\alpha} \in \Big[\sum_{j \leq j^*} h_j^C, \sum_{j \leq j^*+1} h_j^C\Big].$$

Now (ii) gives $h_{j^*+1}^C \leq \tilde{\varepsilon}^{1/4}$, which gives the claim.

The proof of (iv) again follows using (ii) (i.e. as there is only little mass close to the threshold), and we give the proof in Section 5.3. □

5.2 Proof of Soundness: Overview

We use the following lemma, which states that if there is only little mass around the threshold, and the guarantee on the Wasserstein distance (which holds with high probability by the soundness of VerifyHist) indeed holds, then the values the verifier computes are close to the true values $g_{\mathrm{H}}^{C,\alpha}$ and $g_{\mathrm{UH}}^{C,\alpha}$.

Lemma 5.3. *Suppose that the verifier's check (a) holds,*

$$\Pr_{y \leftarrow \mathsf{P}^C}[\mathsf{P}^C(y) \in (1 \pm 100\sqrt{\tilde{\varepsilon}})\alpha 2^{-m}] \leq \tilde{\varepsilon}^{1/4}, \tag{2}$$

and W1$(h^C, h) \leq \frac{20}{t}$*. Then we have*

$$\begin{aligned} &\textit{(i)} && \sum_{j \leq j^*} h_j \in \big[\sum_{j \leq j^*} h_j^C \pm 2\tilde{\varepsilon}^{1/4}\big], \\ &\textit{(ii)} && \sum_{j \leq j^*} h_j \in \big[g_H^{C,\alpha} \pm 3\tilde{\varepsilon}^{1/4}\big], \\ &\textit{(iii)} && \sum_{j \leq j^*} h_j 2^{j\tilde{\varepsilon}} \ \in \big[g_{UH}^{C,\alpha} \pm 46\tilde{\varepsilon}^{1/4}\big]. \end{aligned}$$

With this lemma, it is straightforward to prove soundness:

Proof of soundness. By Claim 5.1, (2) holds with probability at least $1 - 100\tilde{\varepsilon}^{1/4} = 1 - 20\sqrt{\varepsilon}$ over the choice of α. Now, by the soundness of VerifyHist, we get that $\mathrm{W1}(h^C, h) \leq \frac{20}{t}$ with probability at least $1 - 2^{-n}$ (or the verifier rejects). Clearly if (a) does not hold, the verifier rejects. If (a) holds, then by the above lemma we have (ii) and (iii), which as we are considering a no-instance of $\Pi^{\mathrm{PubHeavy},\alpha}$ gives that one of the following holds:

$$\sum_{j \leq j^*} h_j \notin [p_{\mathrm{H}} \pm \tilde{\varepsilon}^{1/4}], \qquad \sum_{j \leq j^*} h_j 2^{j\tilde{\varepsilon}} \notin [p_{\mathrm{UH}} \pm 4\tilde{\varepsilon}^{1/4}].$$

Thus the verifier rejects in (b) or (c). ◻

It remains to prove Lemma 5.3. We focus on the interesting parts of the proof, and defer the details to Section 5.4.

Proof of Lemma 5.3. We defer the proof of (i). It is not hard to see that if $\sum_{j \leq j^*} h^j$ is not in the desired interval, then $\mathrm{W1}(h^C, h)$ is big: by (a) and Lemma 5.2 (ii), only little mass can be around the threshold for both h and h^C, and thus a lot of mass must be moved from below to above the threshold, or vice versa.

Part (ii) can then be proved easily: by Lemma 5.2 (iii), the interval in (i) is contained in $[g_{\mathrm{H}}^{C,\alpha} \pm 3\tilde{\varepsilon}^{1/4}]$.

It remains to prove (iii). For $j \in (t)$ we consider the differences $d_j := h_j - h_j^C$. Then our assumption $\mathrm{W1}(h^C, h) \leq \frac{20}{t}$ gives

$$\frac{20}{t} \geq \mathrm{W1}(h^C, h) = \frac{1}{t} \sum_{i \in (t)} \Big| \sum_{j \leq i} h_j - \sum_{j \leq i} h_j^C \Big| = \frac{1}{t} \sum_{i \in (t)} \Big| \sum_{j \leq i} d_j \Big| = \mathop{\mathrm{E}}_{i \leftarrow (t)} \left[\Big| \sum_{j \leq i} d_j \Big| \right]. \tag{3}$$

Furthermore, part (i) gives

$$\sum_{j \leq j^*} d_j = \sum_{j \leq j^*} h_j - \sum_{j \leq j^*} h_j^C \overset{\text{(i)}}{\in} [\pm 2\tilde{\varepsilon}^{1/4}]. \tag{4}$$

At this point, we will use Lemma 5.4 as stated below which contains the core of the argument. As (3) and (4) hold, we may apply this lemma to the d_j as defined above, and obtain

$$\frac{1}{2^m} \sum_{j \leq j^*} d_j 2^{j\tilde{\varepsilon}} \in \left[\pm 42\tilde{\varepsilon}^{1/4} \right].$$

Plugging in the definition of the d_j, we get

$$\frac{1}{2^m}\sum_{j\le j^*} h_j 2^{j\tilde\varepsilon} \in \Big[\frac{1}{2^m}\sum_{j\le j^*} h_j^C 2^{j\tilde\varepsilon} \pm 42\tilde\varepsilon^{1/4}\Big] \subseteq \big[g_{\mathrm{UH}}^{C,\alpha} \pm 46\tilde\varepsilon^{1/4}\big],$$

where we used Lemma 5.2 (iv) for the above set inclusion. ☐

Lemma 5.4. *Let t, j^* and $\tilde\varepsilon$ be as above, and fix any $d = (d_0, \dots, d_t) \in \mathbb{R}^{t+1}$. Suppose that*

$$(i) \qquad \sum_{j\le j^*} d_j \in [-\delta, \delta], \qquad\qquad (ii) \qquad \mathop{\mathrm{E}}_{i\leftarrow(t)}\big[\big|\sum_{j\le i} d_j\big|\big] \le \frac{20}{t}.$$

Then we have

$$\frac{1}{2^m}\sum_{j\le j^*} d_j 2^{j\tilde\varepsilon} \in \Big[\pm(\delta + 40\tilde\varepsilon)\Big].$$

Again, we defer a few details of the proof to Section 5.4.

Proof. We only prove the the inequality $\sum_{j\le j^*} d_j 2^{j\tilde\varepsilon} \le (\delta + 40\tilde\varepsilon)2^m$. The proof of $\sum_{j\le j^*} d_j 2^{j\tilde\varepsilon} \ge -(\delta + 40\tilde\varepsilon)2^m$ is analogous.

We first define a vector d' such that for all $i < j^*$ it holds that $\sum_{j\le i} d'_i = \min\{\sum_{j\le i} d_i, 0\}$, and $\sum_{j\le j^*} d'_i = \sum_{j\le j^*} d_i$. Note that this defines d' uniquely.

Claim 5.5. *We have*

$$\mathop{\mathrm{E}}_{i\leftarrow(t)}\big[\big|\sum_{j\le i} d'_j\big|\big] \le \mathop{\mathrm{E}}_{i\leftarrow(t)}\big[\big|\sum_{j\le i} d_j\big|\big], \tag{5}$$

$$\sum_{j\le j^*} d'_j 2^{j\tilde\varepsilon} \ge \sum_{j\le i} d_j 2^{j\tilde\varepsilon}. \tag{6}$$

As the proof is not difficult, we defer it to Section 5.4 and just give some intuition here. The first part follows by definition. To prove the second part, we show that d' can be obtained from d by moving mass from coordinate i to coordinate $i+1$ for each i individually. This then implies the claim, as moving mass to larger coordinates only increases the sum.

Now define d'' as follows: $d''_j := d'_j$ for $j < j^*$, $d''_{j^*} := d'_{j^*} - \sum_{j\le j^*} d'_j$, and $d''_j := 0$ for $j > j^*$.

Claim 5.6. *We have*

$$\sum_{j\le j^*} d''_j = 0, \tag{7}$$

$$\forall i \in (t) : \sum_{j\le i} d''_j \le 0, \tag{8}$$

$$\mathop{\mathrm{E}}_{i\leftarrow(t)}\big[\big|\sum_{j\le i} d''_j\big|\big] \le \mathop{\mathrm{E}}_{i\leftarrow(t)}\big[\big|\sum_{j\le i} d'_j\big|\big], \tag{9}$$

$$\sum_{j\le j^*} d''_j 2^{j\tilde\varepsilon} \ge \sum_{j\le j^*} d'_j 2^{j\tilde\varepsilon} - \delta 2^{j^*\tilde\varepsilon}. \tag{10}$$

The proof is straightforward, and we defer it to Section 5.4.

Claim 5.7. *There exists $t \in \mathbb{N}$ and vectors $v^{(1)}, \dots, v^{(t)} \in \mathbb{R}^{t+1}$ such that the following holds:*

(i) $d'' = \sum_{a\in[t]} v^{(a)}$,

(ii) For every $a \in [t]$, $v^{(a)}$ has exactly two nonzero entries, whose index positions we denote by $i(a)$ and $i'(a)$ where $i(a) < i'(a) \leq j^*$. Furthermore, $v^{(a)}_{i(a)} = -w_a$ and $v^{(a)}_{i'(a)} = w_a$ for some $w_a \in \mathbb{R}$, $w_a > 0$.

We prove the claim in Section 5.4. There we show that the vectors $v^{(a)}$ can be defined iteratively by greedily picking the smallest nonzero index position i (which must have negative d_i), and matching it with the smallest index position i' with $d_{i'} > 0$.

Now note that

$$\begin{aligned}\mathop{\mathrm{E}}_{i \leftarrow (t)}\Big[\sum_{j \leq i} d''_j\Big] &= \mathop{\mathrm{E}}_{i \leftarrow (t)}\Big[\sum_{j \leq i}\sum_a v^{(a)}_j\Big] = \sum_a \mathop{\mathrm{E}}_{i \leftarrow (t)}\Big[\sum_{j \leq i} v^{(a)}_j\Big] \\ &= -\sum_a \frac{1}{t} w_a (i'(a) - i(a)). \end{aligned} \tag{11}$$

Then we find

$$\begin{aligned}\sum_{j \leq j^*} d''_j 2^{j\tilde{\varepsilon}} &= \sum_{j \leq j^*}\sum_a v^{(a)}_j 2^{j\tilde{\varepsilon}} = \sum_a \sum_{j \leq j^*} v^{(a)}_j 2^{j\tilde{\varepsilon}} = \sum_a w_a (2^{i'(a)\tilde{\varepsilon}} - 2^{i(a)\tilde{\varepsilon}}) \\ &= \sum_a w_a 2^{i'(a)\tilde{\varepsilon}} (1 - 2^{(i(a) - i'(a))\tilde{\varepsilon}}) \\ &\leq \sum_a w_a 2^{i'(a)\tilde{\varepsilon}} (i'(a) - i(a))(1 - 2^{-\tilde{\varepsilon}}) \\ &= (1 - 2^{-\tilde{\varepsilon}}) \sum_a w_a 2^{i'(a)\tilde{\varepsilon}} (i'(a) - i(a)) \\ &\leq (1 - 2^{-\tilde{\varepsilon}}) \sum_a w_a 2^{j^*\tilde{\varepsilon}} (i'(a) - i(a)) \\ &\overset{(11)}{=} (1 - 2^{-\tilde{\varepsilon}}) 2^{j^*\tilde{\varepsilon}} \left(-t \cdot \mathop{\mathrm{E}}_{i \leftarrow (t)}\Big[\sum_{j \leq i} d''_j\Big] \right) \\ &\overset{(8)}{=} (1 - 2^{-\tilde{\varepsilon}}) 2^{j^*\tilde{\varepsilon}} \cdot t \cdot \mathop{\mathrm{E}}_{i \leftarrow (t)}\Big[\Big|\sum_{j \leq i} d''_j\Big|\Big] \\ &\overset{\text{(ii)}}{\leq} 2\tilde{\varepsilon} \cdot 2^{j^*\tilde{\varepsilon}} \cdot t \cdot \frac{20}{t} = 40\tilde{\varepsilon} 2^{j^*\tilde{\varepsilon}}. \end{aligned} \tag{12}$$

The first inequality above follows by Bernoulli's inequality[5] when setting $n = i'(a) - i(a)$ and $x = 2^{-\tilde{\varepsilon}} - 1$. We now conclude the argument by calculating

$$\begin{aligned}\sum_{j \leq j^*} d_j 2^{j\tilde{\varepsilon}} &\overset{(6)}{\leq} \sum_{j \leq j^*} d'_j 2^{j\tilde{\varepsilon}} \overset{(10)}{\leq} \sum_{j \leq j^*} d''_j 2^{j\tilde{\varepsilon}} + \delta 2^{j^*\tilde{\varepsilon}} \overset{(12)}{\leq} (40\tilde{\varepsilon} + \delta) 2^{j^*\tilde{\varepsilon}} \\ &\leq (40\tilde{\varepsilon} + \delta) 2^m. \end{aligned}$$

□

5.3 Proof of Completeness: the Details

In the following, we give the parts of the proof of Lemma 5.2 that we omitted in Section 5.1.

[5]Bernoulli's inequality states that for any $n \in \mathbb{N}$, $n \geq 0$ and any $x \in \mathbb{R}$, $x \geq -1$ we have $(1 + x)^n \geq 1 + nx$.

Proof of (i). From the definition of j^* we get that $2^{-(j^*+1)\tilde{\varepsilon}} \in (\alpha 2^{-m}, 2^{\tilde{\varepsilon}}\alpha 2^{-m}]$ (otherwise, j^* would not be maximal). Thus, we find

$$\begin{aligned}
2^{-(j^*-\lceil 25/\sqrt{\tilde{\varepsilon}}\rceil)\tilde{\varepsilon}} &\leq 2^{-(j^*-26/\sqrt{\tilde{\varepsilon}})\tilde{\varepsilon}} = 2^{-(j^*+1)\tilde{\varepsilon}} 2^{(26/\sqrt{\tilde{\varepsilon}}+1)\tilde{\varepsilon}} \leq 2^{\tilde{\varepsilon}}\alpha 2^{-m} 2^{27\sqrt{\tilde{\varepsilon}}} \\
&\leq \alpha 2^{-m} 2^{28\sqrt{\tilde{\varepsilon}}}, \\
2^{-(j^*+\lceil 25/\sqrt{\tilde{\varepsilon}}\rceil)\tilde{\varepsilon}} &\geq 2^{-(j^*+26/\sqrt{\tilde{\varepsilon}})\tilde{\varepsilon}} = 2^{-(j^*+1)\tilde{\varepsilon}} 2^{-(26/\sqrt{\tilde{\varepsilon}}-1)\tilde{\varepsilon}} \geq \alpha 2^{-m} 2^{-27\sqrt{\tilde{\varepsilon}}}.
\end{aligned}$$

□

Proof of (ii). Note that $h_j^C = \sum_{y:\mathsf{P}^C(y)\in(2^{-(j+1)\tilde{\varepsilon}},2^{-j\tilde{\varepsilon}}]} \mathsf{P}^C(y)$. Since $j \leq j^* + \lceil 25/\sqrt{\tilde{\varepsilon}}\rceil$, we only sum over y such that

$$\mathsf{P}^C(y) \geq 2^{-(j^*+\lceil 25/\sqrt{\tilde{\varepsilon}}\rceil+1)\tilde{\varepsilon}} = 2^{-\tilde{\varepsilon}} 2^{-(j^*+\lceil 25/\sqrt{\tilde{\varepsilon}}\rceil)\tilde{\varepsilon}} \overset{\text{(i)}}{\geq} \alpha 2^{-m} 2^{-29\sqrt{\tilde{\varepsilon}}}.$$

On the other hand, because $j \geq j^* - \lceil 25/\sqrt{\tilde{\varepsilon}}\rceil$, for all y we sum over, we have

$$\mathsf{P}^C(y) \leq 2^{-(j^*-\lceil 25/\sqrt{\tilde{\varepsilon}}\rceil)\tilde{\varepsilon}} \overset{\text{(i)}}{\leq} \alpha 2^{-m} 2^{28\sqrt{\tilde{\varepsilon}}}.$$

Thus, we conclude that

$$\sum_{j\in\{j^*\pm\lceil 25/\sqrt{\tilde{\varepsilon}}\rceil\}} h_j^C \leq \sum_{y:\alpha 2^{-m}2^{-29\sqrt{\tilde{\varepsilon}}}\leq\mathsf{P}^C(y)\leq\alpha 2^{-m}2^{28\sqrt{\tilde{\varepsilon}}}} \mathsf{P}^C(y) \leq \tilde{\varepsilon}^{1/4},$$

where the last inequality holds because $[2^{-29\sqrt{\tilde{\varepsilon}}}, 2^{28\sqrt{\tilde{\varepsilon}}}] \subseteq (1 \pm 100\sqrt{\tilde{\varepsilon}})$, and thus (1) can be applied. □

Proof of (iv). By definition of h_j^C, we have that for each j

$$\Pr_{y\leftarrow\{0,1\}^m}\big[\mathsf{P}^C(y) \in (2^{-(j+1)\tilde{\varepsilon}}, 2^{-j\tilde{\varepsilon}}]\big] \in \big[\frac{1}{2^m} h_j^C 2^{j\tilde{\varepsilon}}, \frac{1}{2^m} h_j^C 2^{(j+1)\tilde{\varepsilon}}\big].$$

Thus the definition of j^* gives

$$g_{\mathrm{UH}}^{C,\alpha} \in \big[\frac{1}{2^m}\sum_{j\leq j^*} h_j^C 2^{j\tilde{\varepsilon}}, \frac{1}{2^m}\sum_{j\leq j^*+1} h_j^C 2^{(j+1)\tilde{\varepsilon}}\big]. \tag{13}$$

Now we find

$$\begin{aligned}
\sum_{j\leq j^*+1} h_j^C 2^{(j+1)\tilde{\varepsilon}} &= \sum_{j\leq j^*} h_j^C 2^{(j+1)\tilde{\varepsilon}} + h_{j^*+1}^C 2^{(j^*+2)\tilde{\varepsilon}} \\
&\overset{\text{(ii)}}{\leq} 2^{\tilde{\varepsilon}} \sum_{j\leq j^*} h_j^C 2^{j\tilde{\varepsilon}} + \tilde{\varepsilon}^{1/4} 2^{\tilde{\varepsilon}} 2^{(j^*+1)\tilde{\varepsilon}} \\
&< (1+2\tilde{\varepsilon}) \sum_{j\leq j^*} h_j^C 2^{j\tilde{\varepsilon}} + \frac{\tilde{\varepsilon}^{1/4}2^{\tilde{\varepsilon}}}{\alpha} 2^m \\
&\leq \sum_{j\leq j^*} h_j^C 2^{j\tilde{\varepsilon}} + \big(2\tilde{\varepsilon} + \frac{\tilde{\varepsilon}^{1/4}2^{\tilde{\varepsilon}}}{\alpha}\big) 2^m \\
&\leq \sum_{j\leq j^*} h_j^C 2^{j\tilde{\varepsilon}} + 4\tilde{\varepsilon}^{1/4} 2^m,
\end{aligned}$$

where the second inequality follows by definition of j^*, and the third inequality holds since $2^{j\tilde{\varepsilon}} \leq 2^m$ for any $j \leq j^*$ and $\sum_{j\leq j^*} h_j^C \leq 1$. Plugging this into (13) gives the claim. □

5.4 Proof of Soundness: the Details

Proof of Lemma 5.3 (i). First suppose

$$\sum_{j \le j^*} h_j < \sum_{j \le j^*} h_j^C - 2\tilde{\varepsilon}^{1/4}. \tag{14}$$

We show that this implies $\mathrm{W1}(h^C, h) > \frac{20}{t}$, contradicting our assumption. For any $i \in \{j^* - \lceil 25/\sqrt{\tilde{\varepsilon}} \rceil, \ldots, j^*\}$ we have

$$\sum_{j \le i} h_j \le \sum_{j \le j^*} h_j \overset{(14)}{<} \sum_{j \le j^*} h_j^C - 2\tilde{\varepsilon}^{1/4} \le \sum_{j \le i} h_j^C - \tilde{\varepsilon}^{1/4}, \tag{15}$$

where the last inequality holds by Lemma 5.2 (ii). This gives $\overleftarrow{\mathrm{W1}}\,(h^C, h) \ge \frac{1}{t} \cdot \frac{25}{\sqrt{\tilde{\varepsilon}}} \cdot \tilde{\varepsilon}^{1/4} \ge \frac{25}{t}$.

Now assume that

$$\sum_{j \le j^*} h_j > \sum_{j \le j^*} h_j^C + 2\tilde{\varepsilon}^{1/4}. \tag{16}$$

Again, we show that this implies $\mathrm{W1}(h^C, h) > \frac{20}{t}$. Similar to above, for any $i \in \{j^* - \lceil 25/\sqrt{\tilde{\varepsilon}} \rceil, \ldots, j^*\}$ we have

$$\sum_{j \le i} h_j \ge \sum_{j \le j^*} h_j - \tilde{\varepsilon}^{1/4} \overset{(16)}{>} \sum_{j \le j^*} h_j^C + \tilde{\varepsilon}^{1/4} \ge \sum_{j \le i} h_j^C + \tilde{\varepsilon}^{1/4}, \tag{17}$$

where the first inequality holds by the verifier's check (a). Thus $\overrightarrow{\mathrm{W1}}\,(h^C, h) \ge \frac{1}{t} \cdot \frac{25}{\sqrt{\tilde{\varepsilon}}} \cdot \tilde{\varepsilon}^{1/4} \ge \frac{25}{t}$. □

Proof of Claim 5.5. Inequality (5) holds because for each i, $\left|\sum_{j \le i} d_i'\right| \le \left|\sum_{j \le i} d_i\right|$ by definition.

To see (6), for each $k < j^*$ we define $e^{(k)} = (e_0^{(k)}, \ldots, e_t^{(k)})$ as follows. If $\sum_{j \le k} d_j > 0$, we let

$$e_i^{(k)} := \begin{cases} -\sum_{j \le k} d_j & \text{if } i = k, \\ \sum_{j \le k} d_j & \text{if } i = k+1, \\ 0 & \text{otherwise.} \end{cases}$$

and thus

$$\sum_{j \le i} e_i^{(k)} = \begin{cases} -\sum_{j \le k} d_j & \text{if } i = k, \\ 0 & \text{otherwise.} \end{cases}$$

If $\sum_{j \le k} d_j \le 0$, we let $e_i^{(k)} = 0$ for all i. Now we find for any k and i that

$$\sum_{j \le i} (d_j + e_j^{(k)}) = \sum_{j \le i} d_j + \sum_{j \le i} e_j^{(k)} = \begin{cases} \sum_{j \le i} d_j & \text{if } i \ne k, \\ \min\{\sum_{j \le i} d_j, 0\} & \text{if } i = k. \end{cases} \tag{18}$$

This implies that $d + \sum_{k < j^*} e^{(k)} = d'$. Since by definition it holds that $\sum_{j < j^*} e_j^{(k)} 2^{j\tilde{\varepsilon}} \ge 0$ for any k, we find

$$\begin{aligned} \sum_{j \le j^*} d_j' 2^{j\tilde{\varepsilon}} &= \sum_{j \le j^*} (d_j + \sum_{k < j^*} e_j^{(k)}) 2^{j\tilde{\varepsilon}} = \sum_{j \le j^*} d_j 2^{j\tilde{\varepsilon}} + \sum_{j \le j^*} \sum_{k < j^*} e_j^{(k)} 2^{j\tilde{\varepsilon}} \\ &= \sum_{j \le j^*} d_j 2^{j\tilde{\varepsilon}} + \sum_{k < j^*} \sum_{j \le j^*} e_j^{(k)} 2^{j\tilde{\varepsilon}} \ge \sum_{j \le j^*} d_j 2^{j\tilde{\varepsilon}}. \end{aligned}$$

□

Proof of Claim 5.6. Equality (7) holds because $\sum_{j\le j^*} d''_j = \sum_{j<j^*} d'_j + d'_{j^*} - \sum_{j\le j^*} d'_j = 0$, and (8) follows by definition. Inequality (9) follows because for $i < j^*$ we have $\left|\sum_{j\le i} d''_j\right| = \left|\sum_{j\le i} d'_j\right|$, and for $i \ge j^*$ we have $0 = \left|\sum_{j\le i} d''_j\right| \le \left|\sum_{j\le i} d'_j\right|$.

To see (10), we note that

$$\begin{aligned}\sum_{j\le j^*} d''_j 2^{j\tilde{\varepsilon}} &= \sum_{j<j^*} d'_j 2^{j\tilde{\varepsilon}} + d'_{j^*}2^{j^*\tilde{\varepsilon}} - (\sum_{j\le j^*} d'_j)2^{j^*\tilde{\varepsilon}} = \sum_{j\le j^*} d'_j 2^{j\tilde{\varepsilon}} - (\sum_{j\le j^*} d'_j)2^{j^*\tilde{\varepsilon}} \\ &\overset{\text{(i)}}{\ge} \sum_{j\le j^*} d'_j 2^{j\tilde{\varepsilon}} - \delta 2^{j^*\tilde{\varepsilon}}.\end{aligned}$$

□

Proof of Claim 5.7. We define the vectors $v^{(1)}, \dots, v^{(t)}$ using the following procedure.

$a := 0$
$f := d''$
while $(\exists j : f_j \ne 0)$ **do**
  $a := a + 1$
  $i := \min\{j : f_j \ne 0\}$
  $i' := \min\{j : j > i \wedge f_j > 0\}$
  $w := \min\{|f_i|, |f_{i'}|\}$
  for $j = 0$ **to** t **do**
    if $j = i$ **then** $v^{(a)}_j := -w$
    else if $j = i'$ **then** $v^{(a)}_j := w$
    else $v^{(a)}_j := 0$
  $f := f - v^{(a)}$
$t := a$
return $(v^{(1)}, \dots, v^{(t)})$

We claim that the following invariants always hold for f:

$$\text{Invariant 1:} \quad \sum_{j\le j^*} f_j = 0, \qquad\qquad \text{Invariant 2:} \quad \forall k \in (t) : \sum_{j\le k} f_j \le 0.$$

By (7) and (8), the invariants hold in the beginning where $f = d''$. Now suppose the invariants hold for f in some loop iteration, and we show they hold for $f' = f - v^{(a)}$ as defined in the next iteration, given f still has a nonzero component. As invariant 2 holds for f, we have that $f_i < 0$, invariant 1 for f implies that there exists i' with $f_{i'} > 0$. The definition of $v^{(a)}_j$ directly implies that invariant 1 holds for f'. Invariant 2 clearly holds for f' for any $k < i$, as the sum does not change. For $k = i$ it holds because $w < |f_i|$ and f_i is the first non-zero component. For $i < k < i'$ it holds because

$$\sum_{j\le k} f'_j \le \sum_{j\le i} f'_j \le 0,$$

where the first inequality holds by the minimality of i', and the second inequality is invariant 2 for $k = i$. Finally, the second invariant also holds for $k \ge i'$, as then

$$\sum_{j\le k} f'_j = \sum_{j\le k}(f_j - v^{(a)}_j) = \sum_{j\le k} f_j - \underbrace{\sum_{j\le k} v^{(a)}_j}_{=-w+w=0} \le 0,$$

where we applied invariant 2 for f to obtain the inequality.

Finally, in every iteration some nonzero component of f (either f'_i or $f'_{i'}$) is set to 0. Thus the procedure terminates, and in the end we have $\sum_i f_i = 0$ and $\sum_{a=1}^{t} v^{(a)} = d''$. Clearly, the vectors $v^{(a)}$ satisfy (ii). □

6 Analysis of the New Hiding Protocol

Throughout this section, we let $u', \mathcal{Y}', \mathcal{L}', \mathcal{H}'$ be the values as defined by the honest prover's strategy.

6.1 Proof of Completeness: Overview

We define the labeling u' for all $y \in \{0,1\}^m$ as follows:

$$u'(y) = \begin{cases} j & \text{if } \exists j : y \in \mathcal{B}_j, \\ \infty & \text{otherwise.} \end{cases}$$

Note that the honest prover sends a labeling $u = u'$ such that $u'(y_i) = u'(i)$. By definition, we have

Claim 6.1. $\mathsf{P}^C(y) \in (2^{-(u'(y)+1)\varepsilon}, 2^{-u'(y)\varepsilon}]$.

The following lemma states that if the prover is honest, then the values calculated by the verifier in (a), (c)-(e) are close to the true values as defined by P^C.

Lemma 6.2. *Let $\mathcal{S} := \{y_1, \ldots, y_k\}$, $\mathcal{M} := \{y : \mathsf{P}^C(y) \in (1 \pm 4\varepsilon)\alpha 2^{-m}\}$, and assume*

$$\Pr_{y \leftarrow \mathsf{P}^C}[\mathsf{P}^C(y) \in (1 \pm 4\varepsilon)\alpha 2^{-m}] \leq \sqrt{\varepsilon}, \tag{19}$$

then

$$\text{(i)} \qquad \Pr_{y_1,\ldots,y_k}\left[\frac{|\mathcal{Y}'|}{k} \notin [g_{UY}^{C,V,\alpha} \pm \varepsilon]\right] \leq \varepsilon$$

$$\text{(ii)} \qquad \Pr_{y_1,\ldots,y_k}\left[\frac{|\mathcal{H}'|}{k} \notin [g_{UH}^{C,\alpha} \pm 3\sqrt{\varepsilon}]\right] \leq \varepsilon$$

$$\text{(iii)} \qquad \Pr_{y_1,\ldots,y_k}\left[\frac{1}{k}\sum_{i \in \mathcal{L}'} 2^m \cdot 2^{-u'(i)\varepsilon} \notin [1 - g_H^{C,\alpha} \pm 5\sqrt{\varepsilon}]\right] \leq \varepsilon$$

$$\text{(iv)} \qquad \Pr_{y_1,\ldots,y_k}\left[\frac{1}{k}\sum_{i \in \mathcal{L}' \cap \mathcal{Y}'} 2^m \cdot 2^{-u'(i)\varepsilon} \notin [g_{YL}^{C,V,\alpha} \pm 5\sqrt{\varepsilon}]\right] \leq \varepsilon$$

$$\text{(v)} \qquad \Pr_{y_1,\ldots,y_k}\left[\frac{|\mathcal{S} \cap \mathcal{M}|}{k} \geq \frac{3\sqrt{\varepsilon}}{\alpha}\right] \leq \varepsilon$$

With this Lemma, it is not hard to prove completeness:

Proof of completeness. Suppose $(C, V, p_{\mathrm{H}}, p_{\mathrm{UH}}, p_{\mathrm{YL}}, \varepsilon) \in \Pi_Y^{\mathrm{Hide},\alpha}$. Fix α such that we have $\Pr_{y \leftarrow \mathsf{P}^C}[\mathsf{P}^C(y) \in (1 \pm 4\varepsilon)\alpha 2^{-m}] \leq \sqrt{\varepsilon}$. By Claim 4.2, this holds with probability at least $1 - 20\sqrt{\varepsilon}$ over the choice of α.

Since the prover is honest, it sends u' and $\mathcal{Y}'$ with correct witnesses. Then Lemma 6.2 implies that with probability at least $1 - 20\varepsilon$ (a), (c), (d), and (e) hold. Note that (b) always holds since the prover is honest. Finally, The lower bound protocol rejects with probability at most $\varepsilon/2$. □

It remains to prove Lemma 6.2. We defer the formal proof to Section 6.3, as the proof simply applies Chernoff and Hoeffding bounds. Still, we give a short proof sketch.

Proof of Lemma 6.2 (Sketch). Part (i) is a straightforward application of the Chernoff bound.

Part (ii) also follows by the Chernoff bound, but here $|\mathcal{H}'|/k$ may deviate from $g_{\mathrm{UH}}^{C,\alpha}$ by $O(\sqrt{\varepsilon})$ since this much mass may be close to the threshold, and be cut off due to the rounding we introduce with the use of the labeling u'.

The proofs of (iii) and (iv) are applications of the Hoeffding bound, and again the $O(\sqrt{\varepsilon})$ deviation comes in due to the rounding issues as described.

Finally, (v) is a straightforward application of the Chernoff bound on (19). □

6.2 Proof of Soundness: Overview

Suppose $(C, V, p_{\mathrm{H}}, p_{\mathrm{UH}}, p_{\mathrm{YL}}, \varepsilon) \in \Pi_N^{\mathrm{Hide},\alpha}$. Then the following lemma states that if there is not too much probability mass around the threshold, the verifier's checks (a)-(d) are true, the guarantees of the lower bound protocol hold, and the high probability estimates for $u', \mathcal{Y}'$ and $\mathcal{L}'$ hold, then the sum in the verifier's check (e) is close to $g_{\mathrm{YL}}^{C,V,\alpha}$.

Lemma 6.3. *Suppose $(C, V, p_H, p_{UH}, p_{YL}, \varepsilon) \in \Pi_N^{Hide,\alpha}$. Define the sets $\mathcal{S} := \{y_1, \ldots, y_k\}$, $\mathcal{M} := \{y : \mathsf{P}^C(y) \in (1 \pm 4\varepsilon)\alpha 2^{-m}\}$, assume that the verifier's conditions (a)-(d) hold, and*

$$\Pr_{y \leftarrow \mathsf{P}^C}[\mathsf{P}^C(y) \in (1 \pm 4\varepsilon)\alpha 2^{-m}] \leq \sqrt{\varepsilon} \tag{20}$$

$$\forall i \in [k] : u'(i) = \infty \implies u(i) = \infty \tag{21}$$

$$\forall i \in [k] : u(i) = \infty \vee |C^{-1}(y_i)| > (1 - \varepsilon/2) \cdot 2^m \cdot 2^{-(u(i)+1)\varepsilon} \tag{22}$$

$$\frac{|\mathcal{S} \cap \mathcal{M}|}{k} < \frac{3\sqrt{\varepsilon}}{\alpha} \tag{23}$$

$$\frac{|\mathcal{Y}'|}{k} \in [g_{UY}^{C,V,\alpha} \pm \varepsilon], \tag{24}$$

$$\frac{|\mathcal{H}'|}{k} \in [g_{UH}^{C,\alpha} \pm 3\sqrt{\varepsilon}], \tag{25}$$

$$\frac{1}{k} \sum_{i \in \mathcal{L}'} 2^m \cdot 2^{-u'(i)\varepsilon} \in [1 - g_H^{C,\alpha} \pm 5\sqrt{\varepsilon}], \tag{26}$$

$$\frac{1}{k} \sum_{i \in \mathcal{L}' \cap \mathcal{Y}'} 2^m \cdot 2^{-u'(i)\varepsilon} \in [g_{YL}^{C,V,\alpha} \pm 5\sqrt{\varepsilon}]. \tag{27}$$

Then we have

$$\frac{1}{k} \sum_{i \in \mathcal{L} \cap \mathcal{Y}} 2^m 2^{-u(i)\varepsilon} \in [g_{YL}^{C,V,\alpha} \pm 112\sqrt{\varepsilon}\alpha]$$

This lemma allows us to prove soundness as follows.

Proof of soundness. Let $(C, V, p_{\mathrm{H}}, p_{\mathrm{UH}}, p_{\mathrm{YL}}, \varepsilon) \in \Pi_N^{\mathrm{Hide},\alpha}$, and fix α such that $\Pr_{y \leftarrow \mathsf{P}}[\mathsf{P}(y) \in (1 \pm 4\varepsilon)\alpha 2^{-m}] \leq \sqrt{\varepsilon}$. By Claim 4.2, this holds with probability at least $1 - 20\sqrt{\varepsilon}$ over the choice of α.

We proceed to show that with probability at least $1-6\varepsilon$, all the assumptions of Lemma 6.3 hold, or the verifier rejects (with high probability). We have that (20) holds by the above assumption. Moreover, (21) holds because $u'(i) = \infty$ implies $|C^{-1}(y_i)| = 0$ and thus the lower bound protocol rejects with probability 1. Furthermore, (22) holds with probability at least $1-\varepsilon/2$, by the soundness of the parallel lower bound protocol. Finally, by Lemma 6.2 and the union bound, we have that with probability at least $1-5\varepsilon$, all of (23), (24), (25), (26), and (27) hold. Finally, either (a)-(d) hold, or the verifier rejects.

If its assumptions hold, Lemma 6.3 gives that $\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} \in [g_{\mathrm{YL}}^{C,V,\alpha} \pm 112\sqrt{\varepsilon}\alpha]$. Together with the soundness assumption $p_{\mathrm{YL}} \notin [g_{\mathrm{YL}}^{C,V,\alpha} \pm 117\sqrt{\varepsilon}\alpha]$, this implies $\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} \notin [p_{\mathrm{YL}} \pm 5\sqrt{\varepsilon}\alpha]$, which gives that the verifier rejects in (e). This shows that the verifier rejects with probability at least $1-6\varepsilon$. ☐

It remains to prove Lemma 6.3. For this, we will use the notion of Loss and Gain, which is defined as follows:

Definition 6.4. Given two mappings u', u from $[k]$ to $(m)\cup\{\infty\}$ and a set $\mathcal{A}\subseteq[k]$, we define

$$\mathsf{Loss}_{\mathcal{A}}(u',u) := \frac{1}{k}\sum_{i\in\mathcal{A}:u'(i)<u(i)} 2^m(2^{-u'(i)\varepsilon} - 2^{-u(i)\varepsilon}),$$

$$\mathsf{Gain}_{\mathcal{A}}(u',u) := \frac{1}{k}\sum_{i\in\mathcal{A}:u'(i)>u(i)} 2^m(2^{-u(i)\varepsilon} - 2^{-u'(i)\varepsilon}).$$

◊

Note that Gain and Loss are always positive. This notion is supposed to capture the change of probability mass when using the labeling u instead of the labeling u', as described by the following claim. Its proof is not hard, and we defer it to Section 6.4.

Claim 6.5. *For any two mappings u', u from $[k]$ to $(m)\cup\{\infty\}$ and any $\mathcal{A}\subseteq[k]$ we have*

$$\frac{1}{k}\sum_{i\in\mathcal{A}} 2^m 2^{-u(i)\varepsilon} = \frac{1}{k}\sum_{i\in\mathcal{A}} 2^m 2^{-u'(i)\varepsilon} + \mathsf{Gain}_{\mathcal{A}}(u',u) - \mathsf{Loss}_{\mathcal{A}}(u',u).$$

We establish the following sequence of intermediate claims which will then allow us to prove Lemma 6.3.

Claim 6.6. *Under the conditions of Lemma 6.3, we have:*

(i) $\forall i\in[k] : u(i) \geq u'(i) - 1$

(ii) $|\mathcal{L}'\setminus\mathcal{L}| \leq 3k\sqrt{\varepsilon}$

(iii) $|\mathcal{L}\setminus\mathcal{L}'| \leq 19k\sqrt{\varepsilon}$

(iv)
$$\frac{1}{k}\sum_{i\in\mathcal{L}'\setminus\mathcal{L}} 2^m 2^{-u'(i)\varepsilon} \leq 6\sqrt{\varepsilon}\alpha$$
$$\frac{1}{k}\sum_{i\in\mathcal{L}\setminus\mathcal{L}'} 2^m 2^{-u(i)\varepsilon} \leq 38\sqrt{\varepsilon}\alpha$$

(v)
$$\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'} 2^m 2^{-u(i)\varepsilon} \in [1-g_H^{C,\alpha} \pm 44\sqrt{\varepsilon}\alpha]$$
$$\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'} 2^m 2^{-u'(i)\varepsilon} \in [1-g_H^{C,\alpha} \pm 11\sqrt{\varepsilon}\alpha]$$

(vi) $\mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'}(u',u) \leq 4\varepsilon$

$\mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'}(u',u) \leq 59\sqrt{\varepsilon}\alpha$

We defer the proof to Section 6.4, and only sketch the proof here.

Proof (Sketch). Part (i) holds because the claimed probabilities as given by u' cannot be too large, as the lower bound guarantees (21) and (22) hold.

Part (ii) holds because in order to claim that some light $y_i, i \in \mathcal{L}'$ is heavy (i.e. $i \notin \mathcal{L}$), its probability must be close to the threshold, which holds only for an $\Theta(\sqrt{\varepsilon})$-fraction of the y_i's. This holds because the lower bounds are accurate up to a factor of even $(1-\varepsilon/2)$.

Then (iii) follows from (ii), as by condition (c) and $p_{\mathrm{UH}} \in [g_{\mathrm{UH}}^{C,\alpha} \pm \Theta(\sqrt{(\varepsilon)})]$ we have that $|\mathcal{L}|$ and $|\mathcal{L}'|$ can differ by at most $k \cdot \Theta(\sqrt{\varepsilon})$.

Part (iv) is then a direct consequence of (ii) and (iii), using the fact that $i \in \mathcal{L}'$ and $i \in \mathcal{L}$, respectively.

To see (v), we note that the sum over $\mathcal{L}'$ using u' is close to $1 - g_{\mathrm{H}}^{C,\alpha}$ by (26), and the sum over $\mathcal{L}$ using u is close to $1 - g_{\mathrm{H}}^{C,\alpha}$ by the guarantee $p_{\mathrm{H}} \in [g_{\mathrm{H}}^{C,\alpha} \pm 4/5\sqrt{\varepsilon}]$ and (d). Applying (ii) and (iii) gives the result.

Finally, to prove (vi) we note that the two sums in (v) are close, which implies they have small difference. By definition, this difference is exactly $\mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'}(u', u) - \mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'}(u', u)$. Since (i) allows to upper bound the Gain, we get the claim. $\square$

Finally, this allows us to prove our goal as follows. We only sketch the proof and defer the details to Section 6.4.

Proof of Lemma 6.3 (Sketch). We will directly refer to (i)-(vi) as given by Claim 6.6. We make the following observations:

(1) We may as well consider the sum over $\mathcal{Y}'$ instead of $\mathcal{Y}$: this only induces an error of order $O(\varepsilon\alpha)$, because the prover must provide witnesses (see (a) and (b)), and the set $|\mathcal{Y}|$ must still be big (24).

(2) We may as well consider the sum over $\mathcal{L} \cap \mathcal{L}'$ instead of $\mathcal{L}$ or $\mathcal{L}'$: this is a direct consequence of (iv), and induces an error of at most $O(\sqrt{\varepsilon}\alpha)$.

(3) By definition, $\mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'}(u', u) \leq \mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'}(u', u), \mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'}(u', u) \leq \mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'}(u', u)$, and thus both are bounded by $\Theta(\sqrt{\varepsilon}\alpha)$ by (vi).

This allows us to conclude (we put the actual constants to be explicit)

$$
\begin{aligned}
&\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} \overset{(2)}{\in} [\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} \pm 38\sqrt{\varepsilon}\alpha] \\
&\overset{(1)}{\subseteq} [\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u(i)\varepsilon} \pm 42\sqrt{\varepsilon}\alpha] \subseteq [\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon} \pm 101\sqrt{\varepsilon}\alpha] \\
&\overset{(2)}{\subseteq} [\frac{1}{k}\sum_{i\in\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon} \pm 107\sqrt{\varepsilon}\alpha] \overset{(27)}{\subseteq} [g_{\mathrm{YL}}^{C,V,\alpha} \pm 112\sqrt{\varepsilon}\alpha],
\end{aligned}
$$

where the third step follows by (3) and the definition of Gain and Loss. $\square$

6.3 Proof of Completeness: the Details

It is not hard to see that only few y's in the support have $\mathsf{P}^C(y)$ close to the threshold:

Claim 6.7. *Suppose*

$$\Pr_{y \leftarrow \mathsf{P}^C}[\mathsf{P}^C(y) \in (1 \pm 4\varepsilon)\alpha 2^{-m}] \leq \sqrt{\varepsilon}. \tag{28}$$

Then

$$\begin{aligned} &\textit{(i)} \qquad |\{y : \mathsf{P}^C(y) \in (1 \pm 4\varepsilon)\alpha 2^{-m}\}| \leq \frac{2\sqrt{\varepsilon} \cdot 2^m}{\alpha} \\ &\textit{(ii)} \qquad |\{y : 2^{-u'(y)\varepsilon} \geq \alpha 2^{-m} > 2^{-(u'(y)+1)\varepsilon}\}| \leq \frac{2\sqrt{\varepsilon} \cdot 2^m}{\alpha} \end{aligned}$$

Proof of Claim 6.7. We first prove (i). Let $\mathcal{M}_1$ be the set in (i). Now (28) gives that $|\mathcal{M}_1| \leq \frac{\sqrt{\varepsilon}}{(1-4\varepsilon)\alpha 2^{-m}} \leq \frac{2\sqrt{\varepsilon}\cdot 2^m}{\alpha}$ (for $\varepsilon \leq 1/8$).

To see (ii), let $\mathcal{M}_2$ be the set in (ii). Using Claim 6.1, we find that for any $y \in \mathcal{M}_2$ we have

$$\mathsf{P}^C(y) \in [2^{-\varepsilon}\alpha 2^{-m}, 2^{\varepsilon}\alpha 2^{-m}] \subseteq (1 \pm 2\varepsilon)\alpha 2^{-m},$$

and thus $\mathcal{M}_2 \subseteq \mathcal{M}_1$, which proves the claim. □

The following claim describes how we apply the Hoeffding bound, we will use it to prove parts (iii) and (iv) of Lemma 6.2.

Claim 6.8. *Consider the set* $\mathcal{A}_{light} := \{y : 2^{-(u'(y)+1)\varepsilon} < \alpha 2^{-m}\}$, *let* $\mathcal{A}$ *be any subset of* $\{0,1\}^m$, *and define*

$$M := \sum_{y \in \mathcal{A}_{light} \cap \mathcal{A}} 2^{-u'(y)\varepsilon}, \qquad X_i := \begin{cases} 2^m \cdot 2^{-u'(y_i)\varepsilon} & \textit{if } y_i \in \mathcal{A}_{light} \cap \mathcal{A}, \\ 0 & \textit{otherwise.} \end{cases}$$

Then we have for any $\delta > 0$ *that*

$$\Pr_{y_1,\ldots,y_k \in \{0,1\}^{km}}\left[\frac{1}{k}\sum_{i\in[k]} X_i \notin [M \pm \delta]\right] \leq 2 \cdot \exp\left(-\frac{2k\delta^2}{(1+2\varepsilon)^2\alpha^2}\right).$$

Proof. For any $i \in [k]$ we find $\mathrm{E}_{y_i \in \{0,1\}^m}[X_i] = M$, and since $2^{-(u'(y_i)+1)\varepsilon} < \alpha 2^{-m}$ implies that $2^m \cdot 2^{-u'(y_i)\varepsilon} = 2^m \cdot 2^{\varepsilon} \cdot 2^{-(u'(y_i)+1)\varepsilon} \leq (1+2\varepsilon)\alpha$, we have $X_i \in [0, (1+2\varepsilon)\alpha]$. As

$$\mathop{\mathrm{E}}_{y_1,\ldots,y_k}[\frac{1}{k}\sum_{i\in[k]} X_i] = \mathop{\mathrm{E}}_{y_1}[X_1] = M,$$

the Hoeffding bound (Lemma 2.2) gives the claim. □

Proof of Lemma 6.2 (i). Let Y_i be the indicator variable for the event $y_i \in V$. Then $\Pr[Y_i = 1] = g_{\mathrm{UY}}^{C,V,\alpha}$, we have $|\mathcal{Y}'| = \sum_{i\in[k]} Y_i$, and so the Chernoff bound (Lemma 2.1) gives

$$\Pr_{y_1,\ldots,y_k}\left[\frac{|\mathcal{Y}'|}{k} \notin [g_{\mathrm{UY}}^{C,V,\alpha} \pm \varepsilon]\right] = \Pr\left[\sum_{i\in[k]} Y_i \notin [g_{\mathrm{UY}}^{C,V,\alpha} \pm \varepsilon]k\right] \leq 2\exp(-\frac{\varepsilon^2 k}{2}) \leq \varepsilon.$$

□

Proof of Proof of Lemma 6.2 (ii). Let X_i be the indicator variable for the event $y_i \in \{y : 2^{-(u(y)+1)\varepsilon} \geq \alpha 2^{-m}\}$. We first show that $p := \Pr[X_i = 1]$ is close to $g_{\mathrm{UH}}^{C,\alpha}$. Using Claim 6.1, we get

$$\begin{aligned} p &= \frac{|\{y : 2^{-(u(y)+1)\varepsilon} \geq \alpha 2^{-m}\}|}{2^m} \leq \frac{|\{y : \mathsf{P}^C(y) \geq \alpha 2^{-m}\}|}{2^m} = g_{\mathrm{UH}}^{C,\alpha}, \\ g_{\mathrm{UH}}^{C,\alpha} &\leq \frac{|\{y : 2^{-u(y)\varepsilon} \geq \alpha 2^{-m}\}|}{2^m} \\ &= \frac{|\{y : 2^{-(u(y)+1)\varepsilon} \geq \alpha 2^{-m}\}|}{2^m} + \frac{|\{y : 2^{-u(y)\varepsilon} \geq \alpha 2^{-m} > 2^{-(u(y)+1)\varepsilon}\}|}{2^m} \\ &\leq p + 2\sqrt{\varepsilon}, \end{aligned}$$

where we applied Claim 6.7 to obtain the last inequality. This shows that

$$p \in [g_{\mathrm{UH}}^{C,\alpha} - 2\sqrt{\varepsilon}, g_{\mathrm{UH}}^{C,\alpha}]. \tag{29}$$

Now we have $|\mathcal{H}'| = \sum_{i\in[k]} X_i$, and so the Chernoff bound (Lemma 2.1) gives

$$\Pr_{y_1,\ldots,y_k}\left[\frac{|\mathcal{H}'|}{k} \notin [p \pm \varepsilon]\right] = \Pr\left[\sum_{i\in[k]} X_i \notin [p \pm \varepsilon]k\right] \leq 2 \cdot \exp(-\frac{\varepsilon^2 k}{2}) \leq \varepsilon. \tag{30}$$

Plugging (29) into the above gives the claim. □

Proof of Proof of Lemma 6.2 (iii). Let y denote a bitstring in $\{0,1\}^m$. First note that

$$1 - g_{\mathrm{H}}^{C,\alpha} = 1 - \sum_{y:\mathsf{P}^C(y)\geq\alpha 2^{-m}} \mathsf{P}^C(y) = \sum_{y:\mathsf{P}^C(y)<\alpha 2^{-m}} \mathsf{P}^C(y) \tag{31}$$

$$= \sum_{y:\mathsf{P}^C(y)<2^{\varepsilon}\alpha 2^{-m}} \mathsf{P}^C(y) - \underbrace{\sum_{y:\mathsf{P}^C(y)\in[\alpha 2^{-m},2^{\varepsilon}\alpha 2^{-m})} \mathsf{P}^C(y)}_{\in[0,\sqrt{\varepsilon}]} \tag{32}$$

The above sum is indeed in $[0, \sqrt{\varepsilon}]$, as $[\alpha 2^{-m}, 2^{\varepsilon}\alpha 2^{-m}) \subseteq (1 \pm 4\varepsilon)\alpha 2^{-m}$, and thus we can use assumption (19). Using Claim 6.1, we find

$$\begin{aligned} \sum_{y:2^{-(u'(y)+1)\varepsilon}<\alpha 2^{-m}} \mathsf{P}^C(y) &\geq \sum_{y:\mathsf{P}^C(y)<\alpha 2^{-m}} \mathsf{P}^C(y) = 1 - g_{\mathrm{H}}^{C,\alpha} \\ &\overset{(32)}{\geq} \sum_{y:\mathsf{P}^C(y)<2^{\varepsilon}\alpha 2^{-m}} \mathsf{P}^C(y) - \sqrt{\varepsilon} \geq \sum_{y:2^{-u'(y)\varepsilon}<2^{\varepsilon}\alpha 2^{-m}} \mathsf{P}^C(y) - \sqrt{\varepsilon} \\ &= \sum_{y:2^{-(u'(y)+1)\varepsilon}<\alpha 2^{-m}} \mathsf{P}^C(y) - \sqrt{\varepsilon}. \end{aligned}$$

This implies that $1 - g_{\mathrm{H}}^{C,\alpha}$ is in the interval

$$\left[\sum_{y:2^{-(u'(y)+1)\varepsilon}<\alpha 2^{-m}} \mathsf{P}^C(y) \pm \sqrt{\varepsilon}\right] \subseteq \left[2^{\pm\varepsilon} \underbrace{\sum_{y:2^{-(u'(y)+1)\varepsilon}<\alpha 2^{-m}} 2^{-u'(y)\varepsilon}}_{=:M} \pm\sqrt{\varepsilon}\right],$$

where the last step above follows by Claim 6.1. From this we get

$$\begin{aligned}M &\le 2^{\varepsilon}(1-g_{\mathrm{H}}^{C,\alpha}) + 2^{\varepsilon}\sqrt{\varepsilon} \le (1+2\varepsilon)(1-g_{\mathrm{H}}^{C,\alpha}) + 2\sqrt{\varepsilon}\\ &\le (1-g_{\mathrm{H}}^{C,\alpha}) + 2\varepsilon + 2\sqrt{\varepsilon} \le (1-g_{\mathrm{H}}^{C,\alpha}) + 4\sqrt{\varepsilon},\\ M &\ge 2^{-\varepsilon}(1-g_{\mathrm{H}}^{C,\alpha}) - 2^{-\varepsilon}\sqrt{\varepsilon} \ge (1-2\varepsilon)(1-g_{\mathrm{H}}^{C,\alpha}) - \sqrt{\varepsilon}\\ &\ge (1-g_{\mathrm{H}}^{C,\alpha}) - 2\varepsilon - \sqrt{\varepsilon} \ge (1-g_{\mathrm{H}}^{C,\alpha}) - 3\sqrt{\varepsilon},\end{aligned}$$

and thus

$$M \in [(1-g_{\mathrm{H}}^{C,\alpha}) \pm 4\sqrt{\varepsilon}]. \tag{33}$$

Applying Claim 6.8 to M as defined above for $\mathcal{A} = \{0,1\}^m$, we get that

$$\Pr_{y_1,\ldots,y_k}\left[\frac{1}{k}\sum_{i\in\mathcal{L}'} 2^m 2^{-u'(i)\varepsilon} \notin [M \pm \varepsilon]\right] \le 2\exp\left(-\frac{2k\varepsilon^2}{(1+2\varepsilon)^2\alpha^2}\right) \le \varepsilon.$$

By (33), we have $[M \pm \varepsilon] \subseteq [1 - g_{\mathrm{H}}^{C,\alpha} \pm 5\sqrt{\varepsilon}]$, which gives the claim. $\square$

Proof of Proof of Lemma 6.2 (iv). The proof is analogous to the proof of (iii): we find

$$g_{\mathrm{YL}}^{C,V,\alpha} = \sum_{y:\mathsf{P}^C(y)<\alpha 2^{-m}\wedge y\in V} \mathsf{P}^C(y) \in \left[2^{\pm\varepsilon} \sum_{y:2^{-(u(y)+1)\varepsilon}<\alpha 2^{-m}\wedge y\in V} 2^{-u(y)\varepsilon} \pm \sqrt{\varepsilon}\right], \tag{34}$$

where the $\sqrt{\varepsilon}$ deviation can be seen as in (iii), since the sum here has only less summands. Thus, applying Claim 6.8 to the sum in (34) for $\mathcal{A} = \{y : y \in V\}$ gives the claim. $\square$

Proof of Lemma 6.2 (v). Claim 6.7 gives that $|\mathcal{M}| \le \frac{2\sqrt{\varepsilon}\cdot 2^m}{\alpha}$. Let X_i be the indicator random variable for the event $y_i \in \mathcal{M}$. Then $|\mathcal{S}\cap\mathcal{M}| = \sum_{i\in[k]} X_i$, and $p := \mathrm{E}_{y_i}[X_i] \le \frac{2\sqrt{\varepsilon}}{\alpha}$. The Chernoff bound (Lemma 2.1) gives

$$\begin{aligned}\Pr_{\mathcal{S}}\left[\frac{|\mathcal{S}\cap\mathcal{M}|}{k} \ge \frac{3\sqrt{\varepsilon}}{\alpha}\right] &\le \Pr_{\mathcal{S}}\left[\frac{|\mathcal{S}\cap\mathcal{M}|}{k} \ge p + \frac{\sqrt{\varepsilon}}{\alpha}\right]\\ &= \Pr_{\mathcal{S}}\left[\sum_{i\in[k]} X_i \ge (p + \frac{\sqrt{\varepsilon}}{\alpha})k\right] \le \exp\left(-\frac{\varepsilon k}{2\alpha^2}\right) \le \varepsilon.\end{aligned}$$ $\square$

6.4 Proof of Soundness: the Details

Proof of Claim 6.5. By definition of Gain and Loss, the right hand side equals

$$\begin{aligned}
&\frac{1}{k}\sum_{i\in\mathcal{A}} 2^m 2^{-u'(i)\varepsilon} + \frac{1}{k}\sum_{i\in\mathcal{A}:u'(i)>u(i)} 2^m(2^{-u(i)\varepsilon} - 2^{-u'(i)\varepsilon}) \\
&\qquad - \frac{1}{k}\sum_{i\in\mathcal{A}:u'(i)<u(i)} 2^m(2^{-u'(i)\varepsilon} - 2^{-u(i)\varepsilon}) \\
&= \frac{2^m}{k}\Big(\sum_{i\in\mathcal{A}} 2^{-u'(i)\varepsilon} + \sum_{i\in\mathcal{A}:u'(i)>u(i)} 2^{-u(i)\varepsilon} - \sum_{i\in\mathcal{A}:u'(i)>u(i)} 2^{-u'(i)\varepsilon} \\
&\qquad - \sum_{i\in\mathcal{A}:u'(i)<u(i)} 2^{-u'(i)\varepsilon} + \sum_{i\in\mathcal{A}:u'(i)<u(i)} 2^{-u(i)\varepsilon}\Big) \\
&= \frac{2^m}{k}\Big(\sum_{i\in\mathcal{A}} 2^{-u'(i)\varepsilon} + \sum_{i\in\mathcal{A}:u'(i)\neq u(i)} 2^{-u(i)\varepsilon} - \sum_{i\in\mathcal{A}:u'(i)\neq u(i)} 2^{-u'(i)\varepsilon}\Big) \\
&= \frac{1}{k}\sum_{i\in\mathcal{A}} 2^m 2^{-u(i)\varepsilon}.
\end{aligned}$$

□

Proof of Claim 6.6 (i). In case $u'(i) = \infty$, the claim follows by (21). So suppose $u'(i) < \infty$. Towards a contradiction assume $u(i) \le u'(i) - 2$. Then

$$\begin{aligned}
|C^{-1}(y_i)| &= 2^m \mathsf{P}^C(y_i) \le 2^m \cdot 2^{-u'(i)\varepsilon} \le 2^m \cdot 2^{-(u(i)+2)\varepsilon} \\
&= 2^{-\varepsilon} \cdot 2^m \cdot 2^{-(u(i)+1)\varepsilon} \le (1-\varepsilon/2) \cdot 2^m \cdot 2^{-(u(i)+1)\varepsilon},
\end{aligned}$$

where the first inequality follows by Claim 6.1. As $u(i) < \infty$ by assumption, this contradicts (22). □

Proof of Claim 6.6 (ii). By (23), we have $|(\mathcal{L}' \setminus \mathcal{L}) \cap \mathcal{M}| \le \frac{3k\sqrt{\varepsilon}}{\alpha}$. We show that $(\mathcal{L}' \setminus \mathcal{L}) \cap \overline{\mathcal{M}} = \emptyset$. Towards a contradiction assume there is some $y_i \in (\mathcal{L}' \setminus \mathcal{L}) \cap \overline{\mathcal{M}}$. We have

$$y_i \notin \mathcal{M} \implies \mathsf{P}^C(y) \notin (1 \pm 4\varepsilon)\alpha 2^{-m}, \tag{35}$$

$$y_i \in \mathcal{L}' \implies 2^{-(u'(i)+1)\varepsilon} < \alpha 2^{-m}, \tag{36}$$

$$y_i \notin \mathcal{L} \implies 2^{-(u(i)+1)\varepsilon} \ge \alpha 2^{-m}, \tag{37}$$

$$\mathsf{P}^C(y_i) \overset{\text{Claim 6.1}}{\le} 2^{-u'(i)\varepsilon} = 2^{\varepsilon} 2^{-(u'(i)+1)\varepsilon} \overset{(36)}{\le} (1+2\varepsilon)\alpha 2^{-m}. \tag{38}$$

Now (35) and (38) give that

$$\mathsf{P}^C(y_i) \le (1-4\varepsilon)\alpha 2^{-m}. \tag{39}$$

Then we find

$$\begin{aligned}
2^{-(u'(i)+1)\varepsilon} &\overset{\text{Claim 6.1}}{\le} \mathsf{P}^C(y_i) \overset{(39)}{\le} (1-4\varepsilon)\alpha 2^{-m} \overset{(37)}{\le} (1-4\varepsilon) 2^{-(u(i)+1)\varepsilon} \\
&\le 2^{-2\varepsilon} 2^{-(u(i)+1)\varepsilon} = 2^{-(u(i)+3)\varepsilon},
\end{aligned}$$

which implies $u(i) < u'(i) - 1$, and thus contradicts (i). □

Proof of Claim 6.6 (iii). (25), condition (c), and $p_{\mathrm{UH}} \in [g_{\mathrm{UH}}^{C,\alpha} \pm 10\sqrt{\varepsilon}]$ give that

$$|\mathcal{L}'|/k \in [1 - g_{\mathrm{UH}}^{C,\alpha} \pm 3\sqrt{\varepsilon}], \tag{40}$$

$$|\mathcal{L}|/k \in [1 - p_{\mathrm{UH}} \pm 3\sqrt{\varepsilon}] \subseteq [1 - g_{\mathrm{UH}}^{C,\alpha} \pm 13\sqrt{\varepsilon}]. \tag{41}$$

Now we find

$$|\mathcal{L}' \cap \mathcal{L}| = |\mathcal{L}'| - |\mathcal{L}' \setminus \mathcal{L}| \overset{\text{(ii)}}{\geq} |\mathcal{L}'| - k \cdot 3\sqrt{\varepsilon} \overset{(40)}{\geq} k \cdot (1 - g_{\mathrm{UH}}^{C,\alpha} - 6\sqrt{\varepsilon}), \tag{42}$$

and thus

$$\begin{aligned}|\mathcal{L} \setminus \mathcal{L}'| = |\mathcal{L}| - |\mathcal{L} \cap \mathcal{L}'| &\overset{(42)}{\leq} |\mathcal{L}| - k \cdot (1 - g_{\mathrm{UH}}^{C,\alpha} - 6\sqrt{\varepsilon}) \\ &\overset{(41)}{\leq} k \cdot (1 - g_{\mathrm{UH}}^{C,\alpha} + 13\sqrt{\varepsilon}) - k \cdot (1 - g_{\mathrm{UH}}^{C,\alpha} - 6\sqrt{\varepsilon}) = k \cdot 19\sqrt{\varepsilon}.\end{aligned}$$

□

Proof of Claim 6.6 (iv). By definition of $\mathcal{L}$ and $\mathcal{L}'$ we find

$$\begin{aligned}\frac{1}{k}\sum_{i \in \mathcal{L}' \setminus \mathcal{L}} 2^m 2^{-u'(i)\varepsilon} &\leq \frac{1}{k}|\mathcal{L}' \setminus \mathcal{L}| 2^m 2^\varepsilon \alpha 2^{-m} \overset{\text{(ii)}}{\leq} \frac{1}{k} k \cdot 2^\varepsilon \cdot 3\sqrt{\varepsilon}\alpha \\ &= 2^\varepsilon \cdot 3\sqrt{\varepsilon}\alpha \leq 6\sqrt{\varepsilon}\alpha,\end{aligned} \tag{43}$$

$$\begin{aligned}\frac{1}{k}\sum_{i \in \mathcal{L} \setminus \mathcal{L}'} 2^m 2^{-u(i)\varepsilon} &\leq \frac{1}{k}|\mathcal{L} \setminus \mathcal{L}'| 2^m 2^\varepsilon \alpha 2^{-m} \overset{\text{(iii)}}{\leq} \frac{1}{k} k \cdot 2^\varepsilon \cdot 19\sqrt{\varepsilon}\alpha \\ &= 2^\varepsilon \cdot 19\sqrt{\varepsilon}\alpha \leq 38\sqrt{\varepsilon}\alpha.\end{aligned} \tag{44}$$

□

Proof of Claim 6.6 (v). We find

$$\begin{aligned}\frac{1}{k}\sum_{i \in \mathcal{L}} 2^m 2^{-u(i)\varepsilon} &= \frac{1}{k}\sum_{i \in \mathcal{L} \cap \mathcal{L}'} 2^m 2^{-u(i)\varepsilon} + \frac{1}{k}\sum_{i \in \mathcal{L} \setminus \mathcal{L}'} 2^m 2^{-u(i)\varepsilon} \\ &\overset{(44)}{\in} \frac{1}{k}\sum_{i \in \mathcal{L} \cap \mathcal{L}'} 2^m 2^{-u(i)\varepsilon} + [0, 38\sqrt{\varepsilon}\alpha], \\ \frac{1}{k}\sum_{i \in \mathcal{L}'} 2^m 2^{-u'(i)\varepsilon} &= \frac{1}{k}\sum_{i \in \mathcal{L} \cap \mathcal{L}'} 2^m 2^{-u'(i)\varepsilon} + \frac{1}{k}\sum_{i \in \mathcal{L}' \setminus \mathcal{L}} 2^m 2^{-u'(i)\varepsilon} \\ &\overset{(43)}{\in} \frac{1}{k}\sum_{i \in \mathcal{L} \cap \mathcal{L}'} 2^m 2^{-u'(i)\varepsilon} + [0, 6\sqrt{\varepsilon}\alpha].\end{aligned}$$

Applying assumptions (d) and $p_{\mathrm{H}} \in [g_{\mathrm{H}}^{C,\alpha} \pm \frac{4}{5}\sqrt{\varepsilon}]$ to the first inclusion and assumption (26) to the second inclusion gives the claim. □

Proof of Claim 6.6 (vi). Claim 6.5 gives

$$\frac{1}{k}\sum_{i \in \mathcal{L} \cap \mathcal{L}'} 2^m 2^{-u(i)\varepsilon} = \frac{1}{k}\sum_{i \in \mathcal{L} \cap \mathcal{L}'} 2^m 2^{-u'(i)\varepsilon} + \mathsf{Gain}_{\mathcal{L} \cap \mathcal{L}'}(u', u) - \mathsf{Loss}_{\mathcal{L} \cap \mathcal{L}'}(u', u). \tag{45}$$

Furthermore, (v) gives

$$\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'} 2^m 2^{-u(i)\varepsilon} \in \left[\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'} 2^m 2^{-u'(i)\varepsilon} \pm 55\sqrt{\varepsilon}\alpha\right] \tag{46}$$

Combining (45) and (46) then gives

$$\mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'}(u',u) - \mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'}(u',u) \in [0 \pm 55\sqrt{\varepsilon}\alpha] \tag{47}$$

Now we find

$$\begin{aligned}
\mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'}(u',u) &= \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}':u'(i)>u(i)} 2^m(2^{-u(i)\varepsilon} - 2^{-u'(i)\varepsilon}) \\
&\overset{\text{(i)}}{\leq} \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}':u'(i)>u(i)} 2^m(2^{-(u'(i)-1)\varepsilon} - 2^{-u'(i)\varepsilon}) \\
&= \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}':u'(i)>u(i)} 2^m(2^{\varepsilon}\cdot 2^{-u'(i)\varepsilon} - 2^{-u'(i)\varepsilon}) \\
&= \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}':u'(i)>u(i)} 2^m(2^{\varepsilon} - 1)2^{-u'(i)\varepsilon} \\
&\leq 2\varepsilon\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}':u'(i)>u(i)} 2^m 2^{-u'(i)\varepsilon} \\
&\overset{(26)}{\leq} 2\varepsilon(1 - g_{\mathrm{H}}^{C,\alpha} + 5\sqrt{\varepsilon}) \leq 4\varepsilon.
\end{aligned} \tag{48}$$

Now we get

$$\mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'}(u',u) \overset{(47)}{\leq} \mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'}(u',u) + 55\sqrt{\varepsilon}\alpha \overset{(48)}{\leq} 59\sqrt{\varepsilon}\alpha.$$

□

Proof of Lemma 6.3. Throughout the proof we will write (i)-(vi) to refer to the corresponding items of Claim 6.6. Assumptions (a), (b) and (24), imply that

$$\mathcal{Y} \subseteq \mathcal{Y}' \tag{49}$$

$$|\mathcal{Y}' \setminus \mathcal{Y}| \leq 2\varepsilon k, \tag{50}$$

and so we find

$$\begin{aligned}
\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u(i)\varepsilon} &\overset{(49)}{\geq} \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} \\
&= \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u(i)\varepsilon} - \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap(\mathcal{Y}'\setminus\mathcal{Y})} 2^m 2^{-u(i)\varepsilon} \\
&\overset{(i\in\mathcal{L})}{\geq} \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u(i)\varepsilon} - \frac{1}{k}|\mathcal{Y}'\setminus\mathcal{Y}|2^m 2^{\varepsilon}\alpha 2^{-m} \\
&\overset{(50)}{\geq} \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u(i)\varepsilon} - 4\varepsilon\alpha.
\end{aligned}$$

This gives

$$\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} \in [\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u(i)\varepsilon} \pm 4\varepsilon\alpha]. \tag{51}$$

We find

$$\begin{aligned}\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} &= \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} + \frac{1}{k}\sum_{i\in(\mathcal{L}\setminus\mathcal{L}')\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon}\\ &\overset{\text{(iv)}}{\in} \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} + [0, 38\sqrt{\varepsilon}\alpha], \qquad (52)\\ \frac{1}{k}\sum_{i\in\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon} &= \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon} + \frac{1}{k}\sum_{i\in(\mathcal{L}'\setminus\mathcal{L})\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon}\\ &\overset{\text{(iv)}}{\in} \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon} + [0, 6\sqrt{\varepsilon}\alpha], \qquad (53)\end{aligned}$$

where we could apply (iv) because $(\mathcal{L}\setminus\mathcal{L}')\cap\mathcal{Y}\subseteq\mathcal{L}\setminus\mathcal{L}'$ and $(\mathcal{L}'\setminus\mathcal{L})\cap\mathcal{Y}'\subseteq\mathcal{L}'\setminus\mathcal{L}$. Claim 6.5 gives

$$\begin{aligned}\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u(i)\varepsilon} = \frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon} &+ \mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'}(u', u)\\ &- \mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'}(u', u),\end{aligned} \tag{54}$$

and we get

$$\mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'}(u', u) \le \mathsf{Gain}_{\mathcal{L}\cap\mathcal{L}'}(u', u) \overset{\text{(vi)}}{\le} 4\varepsilon, \tag{55}$$

$$\mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'}(u', u) \le \mathsf{Loss}_{\mathcal{L}\cap\mathcal{L}'}(u', u) \overset{\text{(vi)}}{\le} 59\sqrt{\varepsilon}\alpha. \tag{56}$$

Thus we find

$$\begin{aligned}\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} &\overset{(52)}{\in} [\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}} 2^m 2^{-u(i)\varepsilon} \pm 38\sqrt{\varepsilon}\alpha]\\ &\overset{(51)}{\subseteq} [\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u(i)\varepsilon} \pm 42\sqrt{\varepsilon}\alpha]\\ &\overset{(54),\,(55),\,(56)}{\subseteq} [\frac{1}{k}\sum_{i\in\mathcal{L}\cap\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon} \pm 101\sqrt{\varepsilon}\alpha]\\ &\overset{(53)}{\subseteq} [\frac{1}{k}\sum_{i\in\mathcal{L}'\cap\mathcal{Y}'} 2^m 2^{-u'(i)\varepsilon} \pm 107\sqrt{\varepsilon}\alpha]\\ &\overset{(27)}{\subseteq} [g_{\mathrm{YL}}^{C,V,\alpha} \pm 112\sqrt{\varepsilon}\alpha].\end{aligned}$$

This concludes the proof. □

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach.* Cambridge University Press, 2009.

[AGGM06] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on np-hardness. In Jon M. Kleinberg, editor, *STOC*, pages 701–710. ACM, 2006. See also errata on author's webpage: `http://www.wisdom.weizmann.ac.il/~oded/p_aggm.html`.

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *STOC*, pages 99–108. ACM, 1996.

[Bab85] László Babai. Trading group theory for randomness. In Robert Sedgewick, editor, *STOC*, pages 421–429. ACM, 1985.

[BDCGL92] Shai Ben-David, Benny Chor, Oded Goldreich, and Michael Luby. On the theory of average case complexity. *J. Comput. Syst. Sci.*, 44(2):193–219, 1992.

[BK95] Manuel Blum and Sampath Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, 1995.

[BL13] Andrej Bogdanov and Chin Ho Lee. Limits of provable security for homomorphic encryption. In Ran Canetti and Juan A. Garay, editors, *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*, pages 111–128. Springer, 2013.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC*, pages 575–584. ACM, 2013.

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

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

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

[Bra83] Gilles Brassard. Relativized cryptography. *IEEE Transactions on Information Theory*, 29(6):877–893, 1983.

[BT06a] Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1), 2006.

[BT06b] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006.

[DBL10] *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, June 9-12, 2010.* IEEE Computer Society, 2010.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[EY80] Shimon Even and Yacov Yacobi. Cryptocomplexity and NP-completeness. In J. W. de Bakker and Jan van Leeuwen, editors, *ICALP*, volume 85 of *Lecture Notes in Computer Science*, pages 195–207. Springer, 1980.

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

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

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

[Gol97] Oded Goldreich. Notes on levin's theory of average-case complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(58), 1997.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In Juris Hartmanis, editor, *STOC*, pages 59–68. ACM, 1986.

[GSTS07] Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. If NP languages are hard on the worst-case, then it is easy to find their hard instances. *Computational Complexity*, 16(4):412–441, 2007.

[GTS07] Dan Gutfreund and Amnon Ta-Shma. Worst-case to average-case reductions revisited. In Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim, editors, *APPROX-RANDOM*, volume 4627 of *Lecture Notes in Computer Science*, pages 569–583. Springer, 2007.

[HMX10] Iftach Haitner, Mohammad Mahmoody, and David Xiao. A new sampling protocol and applications to basing cryptographic primitives on the hardness of NP. In *IEEE Conference on Computational Complexity* [DBL10], pages 76–87.

[Hoe63] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[IL90] Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *FOCS*, pages 812–821. IEEE Computer Society, 1990.

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

[Imp11] Russell Impagliazzo. Relativized separations of worst-case and average-case complexities for NP. In *IEEE Conference on Computational Complexity*, pages 104–114. IEEE Computer Society, 2011.

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

[LM09] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2009.

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

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[MX10] Mohammad Mahmoody and David Xiao. On the power of randomized reductions and the checkability of sat. In *IEEE Conference on Computational Complexity* [DBL10], pages 64–75.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *STOC*, pages 333–342. ACM, 2009.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[Reg10] Oded Regev. The learning with errors problem (invited survey). In *IEEE Conference on Computational Complexity* [DBL10], pages 191–204.

[Rub90] Ronitt Rubinfeld. *A mathematical theory of self-checking, self-testing and self-correcting programs*. PhD thesis, UC Berkeley, 1990.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the xor lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[Wat12] Thomas Watson. Relativized worlds without worst-case to average-case reductions for NP. *TOCT*, 4(3):8, 2012.

[Yap83] Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983.