Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 52 (2026)

A Sharp Characterization of Pessiland

Shuichi Hirahara* Mikito Nanashima†

Abstract

It is a long-standing open question whether the average-case hardness of NP implies the existence of a one-way function. The hypothetical world in which this does not hold is called *Pessiland*, which is the most pessimistic among Impagliazzo's five possible worlds. In this paper, we present the first "sharp" characterization of Pessiland:

- NP is hard on average if and only if the minimum description length of programs in agnostic learning is hard to approximate on average with an approximation factor $\ell^{1-o(1)}$, where ℓ is a new parameter of a distribution called *advice complexity of sampling*.
- A one-way function does not exist if and only if the minimum description length of programs in agnostic learning is easy to approximate on average, with respect to every polynomial-time samplable distribution over instances, with an approximation factor $O(\ell)$.

In particular, Pessiland is ruled out if and only if the small quantitative gap in approximation factors $\ell^{1-o(1)}$ and $O(\ell)$ is closed.

Our characterization is based on an optimal NP-hardness result for the *Collective Minimum Monotone Satisfying Assignment (CMMSA) Problem*, whose task is, given as input a collection of monotone formulas with at most ℓ literals, to compute the minimum weight of an assignment that satisfies as many monotone formulas as possible. We prove the NP-hardness of approximating the minimum weight within a factor of $\ell^{1-o(1)}$, improving the previous inapproximability factor of $\ell^{\Omega(1)}$ by Hirahara (FOCS 2022). This inapproximability factor is optimal up to the $o(1)$ loss in the exponent, unless $\mathsf{NP} \subseteq \mathsf{coAM}$, because CMMSA admits a coAM protocol with approximation factor $O(\ell)$.

*National Institute of Informatics, Japan. s_hirahara@nii.ac.jp
†Institute of Science Tokyo, Japan. nanashima@comp.isct.ac.jp

ISSN 1433-8092

Contents

1 Introduction

A one-way function is a function that is easy to compute but hard to invert on average. This is one of the most fundamental cryptographic primitives because the existence of a one-way function is sufficient for constructing various "Minicrypt" primitives, such as private-key encryption schemes [GM84], commitment schemes [Nao91], pseudorandom function generators [GGM86; HILL99], zero knowledge protocols for all NP [GMW91; NOV06], and is also necessary for these cryptographic primitives [IL89; OW93; HN24]. It is thus of central importance to investigate what is a minimal hypothesis sufficient for the existence of a one-way function. The existence of a one-way function clearly implies the average-case hardness of NP (with respect to some polynomial-time samplable distribution). Whether the converse holds or not is the central open question known as the exclusion of Pessiland — a hypothetical world in which NP is hard on average, yet no one-way function exists — from Impagliazzo's five possible worlds [Imp95]. Pessiland is the most pessimistic in that neither cryptography nor heuristic algorithms for NP exist. Recently, Hirahara and Nanashima [HN23] put forward an alternative name of Pessiland — "Learnabilica", in which there exist efficient average-case algorithms for various learning tasks, such as PAC learning [BFKL93], learning adaptively changing distributions [NR06], distributional learning and agnostic learning [IL90; HN23].

In this paper, we continue to study the complexity of agnostic learning of minimum programs. Given random labeled examples $(x_1, b_1), \cdots, (x_m, b_m)$ drawn from an unknown distribution $\mathcal{D}$, the task of agnostic learning is to approximately calculate the minimum length of programs M such that the probability that $M(x) = b$ for $(x, b) \sim \mathcal{D}$ is close to 1. We introduce some parameter ℓ of distributions $\mathcal{D}$, which we call *advice complexity of sampling*, and present the following new characterization of Pessiland.

Main Theorem (informal)**.** For the advice-complexity parameter ℓ, the following hold as ℓ ranges over sufficiently large integers.

- NP *is hard on average* $\iff$ *agnostic learning with factor* $\ell^{1-o(1)}$ *is hard on average.*
- *A one-way function exists* $\iff$ *agnostic learning with factor* $C\ell$ *is hard on average for a sufficiently large constant* C*.*

Corollary (informal)**.** *Pessiland exists if and only if average-case agnostic learning is hard with an approximation factor* $\ell^{1-o(1)}$ *for some samplable distribution and is easy with an approximation factor* $O(\ell)$ *for every samplable distribution.*

This is the first "sharp" characterization of Pessiland. Although there has been a flurry of recent characterizations of the existence of one-way functions [LP20; RS21; LP21; IRS22; ACMTV21; LP22; LP23a; LP23b; HILNO23; Hir23; IL90; HN23; HLN24], none of them is quantitatively close to the average-case hardness of NP. For example, Liu and Pass [LP22] presented the equivalence between the existence of one-way functions and average-case hardness of *polynomial*-time-bounded conditional Kolmogorov complexity with respect to the *uniform distribution*, as well as the equivalence between average-case hardness of NP and average-case hardness of the *sublinear*-time-bounded conditional Kolmogorov complexity with respect to *some polynomial-time samplable distribution*. Their characterizations leave large quantitative (polynomial vs. sublinear-time bound) and qualitative (uniform vs. some distribution) gaps. Recently, Lu and Santhanam [LS24] extended this characterization to arbitrary polynomial-time samplable distributions (by considering probabilistic Kolmogorov complexity), which closes the qualitative gap but still leaves the large quantitative gap in the time bounds. By contrast, our characterizations leave only the small quantitative gap between approximation factors $\ell^{1-o(1)}$ and $O(\ell)$, which intuitively suggests that Pessiland is very unlikely to exist.

Ruling out Pessiland is equivalent to closing the small quantitative gap between these approximation factors. Whether or not this means we are "close" to actually ruling out Pessiland remains to be seen, just as in the case of similar sharp threshold results for circuit lower bounds [CJW20]. In fact, our characterization is given by an optimal NP-hardness result of the Collective Minimum Monotone Satisfying Assignment (CMMSA) problem, which cannot be improved further unless $\mathsf{NP} \subseteq \mathsf{coAM}$. This suggests that new ideas are necessary, despite the quantitative closeness to ruling out Pessiland.

We proceed to describe the details of our results. In Section 1.1, we introduce the notion of advice complexity of sampling and the definition of agnostic learning of programs. In Section 1.2, we present the optimal NP-hardness result for CMMSA.

1.1 Agnostic Learning of Minimum Programs

We introduce the notion of advice complexity of sampling.

Definition 1.1 (Advice complexity of sampling; informal; see Definition 3.11 for the formal definition)**.** *For a family $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ of distributions, where $\mathsf{Support}(\mathcal{D}_n) \subseteq \{0,1\}^n$, we say that $\mathcal{D}$ is* samplable with advice complexity $\ell \in \mathbb{N}$ *if there exist a (not necessarily efficient) algorithm S and a (not necessarily computable) family $\alpha = \left\{\alpha_n : \{0,1\}^* \to \{0,1\}^{\leq \ell}\right\}_{n\in\mathbb{N}}$ of functions such that $S(n, r, \alpha_n(r))$ outputs $\perp$ with probability at most $1/4$ and, conditioned on not outputting $\perp$, its output distribution has statistical distance at most 2^{-2n} from $\mathcal{D}_n$.*

This is analogous to the notion of Trevisan–Vadhan advice [TV07], which is an advice string that may depend on the internal randomness of a randomized algorithm. For simplicity, the informal definition is given only for a *family* of distributions; in fact, we may define the advice complexity of $\mathcal{D}_n$ for every distribution $\mathcal{D}_n$ over $\{0,1\}^n$ based on a universal Turing machine (see Definition 3.11). The new notion of advice complexity of sampling is fairly robust, and is shown to be equivalent to other notions, such as the ∞-Rényi divergence to the universal distribution (the property of domination by the universal distribution) and a coding theorem of Kolmogorov complexity (up to a constant factor); see Appendix B for details.

Many natural distributions have small advice complexity. For example, consider a uniformly random function $f\colon \{0,1\}^n \to \{0,1\}$, and consider the distribution $\mathcal{E}$ of a labeled example $(x, f(x))$ for a uniformly random $x \sim \{0,1\}^n$. The advice complexity of $\mathcal{E}$ is 1, despite that the description length (the Kolmogorov complexity) of the distribution itself is as large as $\Omega(2^n)$.

Now we introduce the problem of agnostic learning minimum programs. Ko [Ko91] introduced the problem of learning minimum programs, denoted by MINLT. Informally, MINLT is the problem of finding a minimum program M such that $M(x_i) = b_i$ for all $i \in [m]$ for given labeled examples $(x_1, b_1), \cdots, (x_m, b_m) \in \{0,1\}^n \times \{0,1\}$. Here, we consider an agnostic variant of MINLT: Instead of finding a minimum program M that is consistent with *all* labeled examples $(x_1, b_1), \cdots, (x_m, b_m)$, we aim to find M that is consistent with *most* labeled examples. For a technical reason,[1] we formulate it based on the formulation of Valiant [Val84], where a learner is given oracle access to an example oracle $\mathcal{E}$, which returns a labeled example $(x, h(x))$ for an unknown function h, and is asked to compute a function that approximates h.

Definition 1.2 (GapLearn)**.** *For $\sigma, \ell \in \mathbb{N}$ and $\varepsilon, \gamma \in [0,1]$, the problem $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ is defined as follows. Given parameters 1^n and 1^s and access to an example oracle $\mathcal{E}$, where the oracle $\mathcal{E}$ returns*

[1] To obtain an equivalence between the average-case hardness of this problem and the existence of a one-way function in Theorem 1.5, it is crucial that the description of the example distribution is unknown to a learner. On the other hand, Theorem 1.3 and Corollary 1.4 holds even if the explicit description of an example distribution $\mathcal{E}$ is given to a learner.

random and independent samples drawn from a distribution over $\{0,1\}^n \times \{0,1\}$ *that is promised to be samplable with advice complexity* ℓ*, distinguish the following two cases:*

(Yes Cases) *There exists a linear-time program* h *of length* s *such that*

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] \geq 1 - \varepsilon.$$

(No Cases) *For all programs* h *of length at most* $\sigma \cdot s$*,*

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] \leq \frac{1}{2} + \frac{\gamma}{2}.$$

This is an *agnostic learning* variant of MINLT in that in the Yes case, the polynomial-time program h is allowed to err on a ε-fraction of inputs.

Hirahara [Hir22] proved NP-hardness of MINLT under randomized polynomial-time reductions. Building on this reduction, we prove the following NP-hardness.

Theorem 1.3. *There exist functions* $\varepsilon(\ell), \gamma(\ell) = o(1)$*, and* $\sigma(\ell) = \ell^{1-o(1)}$ *such that the problem* $\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{\sigma(\ell)}\mathrm{Learn}[\ell]$ *is* NP*-hard under randomized many-one reductions for every sufficiently large* $\ell \in \mathbb{N}$*.*

Since this NP-hardness is proved by nonadaptive reductions, we obtain some polynomial-time samplable distribution $\mathcal{D}$ (naturally induced by the reduction) with respect to which the average-case analogue of $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ is DistNP-complete (i.e., complete for an average-case analogue of NP), where $\mathcal{D}$ is a distribution over instances of $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ (i.e., it chooses parameters n, s and the description of an example oracle $\mathcal{E}$, which can be represented as a circuit.)

Corollary 1.4. *There exist functions* $\varepsilon(\ell), \gamma(\ell) = o(1)$*, and* $\sigma(\ell) = \ell^{1-o(1)}$ *such that for every sufficiently large* $\ell \in \mathbb{N}$*, the following are equivalent.*

- DistNP $\not\subseteq$ HeurBPP *(i.e., there is a problem in* NP *that is hard on average with respect to some polynomial-time samplable distribution).*
- $(\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{\sigma(\ell)}\mathrm{Learn}[\ell], \mathcal{D}) \notin$ HeurBPP *for some polynomial-time samplable distribution* $\mathcal{D}$*. (i.e.,* $\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{\sigma(\ell)}\mathrm{Learn}[\ell]$ *is hard on average with respect to* $\mathcal{D}$*).*

Here, HeurBPP is the class of distributional problems solvable by randomized error-prone heuristic schemes, i.e., algorithms that are allowed to err on a small fraction of inputs. See the survey of Bogdanov and Trevisan [BT06a] for background on average-case complexity.

We complement this by showing that the average-case hardness of $\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{\sigma(\ell)}\mathrm{Learn}[\ell]$ characterizes the existence of one-way functions when $\sigma(\ell) = O(\ell)$, based on the average-case learning algorithms in Pessiland [IL90; HN23].

Theorem 1.5. *There exists a constant* C *such that for all functions* $\varepsilon(\ell), \gamma(\ell) = o(1)$*, the following are equivalent for all sufficiently large* ℓ*:*

- *There exists an infinitely-often one-way function.*
- $(\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{\sigma(\ell)}\mathrm{Learn}[\ell], \mathcal{D}) \notin$ HeurBPP *for* $\sigma(\ell) \coloneqq C \cdot \ell$ *and for some polynomial-time samplable distribution* $\mathcal{D}$*.*

- $(\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{\sigma(n)}\mathrm{Learn}[\ell], \mathcal{D}) \notin \mathsf{HeurBPP}$ *for some* $\sigma(n)$ *satisfying* $C \cdot \ell \leq \sigma(n) \leq n^{O(1)}$ *and for some polynomial-time samplable distribution* $\mathcal{D}$.

Corollary 1.4 and Theorem 1.5 immediately provide a sharp characterization of Pessiland.

Corollary 1.6 (A Sharp Characterization of Pessiland)**.** *There exist functions* $\varepsilon(\ell), \gamma(\ell) = o(1)$, $\sigma_1(\ell) = \ell^{1-o(1)}$ *and* $\sigma_2(\ell) = O(\ell)$ *such that Pessiland exists if and only if, for every sufficiently large* $\ell \in \mathbb{N}$*, the following two conditions hold:*

- $(\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma_1(\ell)}\mathrm{Learn}[\ell], \mathcal{D}_1) \notin \mathsf{HeurBPP}$ *for some polynomial-time samplable distribution* $\mathcal{D}_1$*; and*
- $(\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma_2(\ell)}\mathrm{Learn}[\ell], \mathcal{D}_2) \in \mathsf{HeurBPP}$ *for every polynomial-time samplable distribution* $\mathcal{D}_2$.

For comparison, if one instead starts from the ℓ^α-inapproximability result for CMMSA due to Hirahara [Hir22], the same argument yields a weaker analogue of Corollary 1.6, namely a characterization of Pessiland via GapLearn with approximation gap ℓ^α versus $O(\ell)$ for some small constant $\alpha > 0$. Corollary 1.6 sharpens the hardness factor from ℓ^α to $\ell^{1-o(1)}$, leaving only a nearly tight gap to the $O(\ell)$-approximation regime.

Optimistically, this characterization can be seen as an approach towards eliminating Pessiland. Pessiland is ruled out *if and only if* the small quantitative gap between the inapproximability factors $\sigma_1(\ell) = \ell^{1-o(1)}$ and $\sigma_2(\ell) = O(\ell)$ is closed. However, new ideas are certainly necessary because our NP-hardness is based on an *optimal* NP-hardness result of CMMSA, which we explain next.

1.2 Minimum Monotone Satisfying Assignment Problem

The Minimum Monotone Satisfying Assignment (MMSA) problem, introduced by Goldwasser and Motwani [GM97] and Alekhnovich, Buss, Moran, and Pitassi [ABMP01], asks for the minimum weight of a satisfying assignment for a given monotone formula. This is a generalization of several optimization problems, such as the Set Cover problem (i.e., the case of monotone DNFs) and the Red-Blue Set Cover problem (i.e., the case of depth-3 formulas with a top AND gate) [CDKM00; CNW16]. A sequence of works [DS04; DFKRS11; DHK15] established the NP-hardness of approximating MMSA within a factor of $n^{1/(\log\log n)^{O(1)}}$, where n is the number of variables (see [Hir22]). The hardness of approximating MMSA has been instrumental in establishing hardness results in proof complexity [ABMP01; AR08] and learning theory [ABFKP08; Hir22].

The Collective Minimum Monotone Satisfying Assignment (CMMSA) problem, introduced by Hirahara [Hir22], is a variant of MMSA in which the topmost gate of a given formula is an approximate threshold gate. Informally, this problem asks for the minimum weight of an assignment that satisfies as many formulas as possible given a collection of monotone formulas as input. In order to present the formal description of an approximation version, let $[n]$ denote $\{1, \ldots, n\}$ for each $n \in \mathbb{N}$. A *weight function* $w \colon [n] \to [0,1]$ is a function such that $\sum_{i\in[n]} w(i) = 1$. We define the *w-weight of an assignment* $\alpha \in \{0,1\}^n$ as $w(\alpha) = \sum_{i\in[n]\colon \alpha_i = 1} w(i)$.

Definition 1.7 (GapCMMSA for a class $\mathfrak{C}$)**.** *Let* $\mathfrak{C} = \{\mathfrak{C}_n\}_{n\in\mathbb{N}}$ *be a class of monotone functions, where* $\varphi \colon \{0,1\}^n \to \{0,1\}$ *for each* $\varphi \in \mathfrak{C}_n$*. For* $\varepsilon, \gamma \in [0,1]$ *and* $\sigma \in \mathbb{N}$*, the problem* $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathfrak{C}$-CMMSA *is defined as follows: Given as input a collection* $\Phi = \{\varphi_1, \ldots, \varphi_m\} \subseteq \mathfrak{C}_n$*, a weight function* $w \colon [n] \to [0,1]$*, and a size parameter* $s \in [0,1]$*, the task is to distinguish between the following two cases:*

(Yes Cases) *There exists an assignment* $\alpha \in \{0,1\}^n$ *of* w*-weight at most* s *such that*

$$\Pr_{\varphi\sim\Phi}[\varphi(\alpha) = 1] \geq 1 - \varepsilon,$$

where $\varphi \sim \Phi$ *means that* φ *is randomly and uniformly chosen from* Φ.

(No Cases) *There exists no assignment $\alpha \in \{0,1\}^n$ of w-weight at most $\sigma \cdot s$ such that*

$$\Pr_{\varphi \sim \Phi}[\varphi(\alpha) = 1] \geq \gamma.$$

Throughout this paper, we assume that the weight function and the size parameter are lower bounded by the inverse of a polynomial in the number of variables. We also assume that each φ_i is an $O(\log n)$-junta (i.e., it depends only on $O(\log n)$ variables among n input variables), and thus each φ_i can be represented by its truth table without blowing up the input size too much.

When $\sigma \equiv 1$, this problem is studied under the name of *biased CSP* [GL22], which includes natural problems such as the Densest s-Subgraph problem (the case where $\mathfrak{C}$ is the class of AND functions over 2 bits).

For $\ell \in \mathbb{N}$, let $\mathsf{F}[\ell] = \{\mathsf{F}[\ell]_n\}_{n\in\mathbb{N}}$ denote the class of functions computable by monotone formulas on n variables with ℓ literals. Hirahara [Hir22] proved NP-hardness of $\mathrm{Gap}^{0,\gamma(\ell)}_{\sigma(\ell)}\mathsf{F}[\ell]$-CMMSA for $\sigma(\ell) = \ell^{\alpha}$ and $\gamma(\ell) = 1/\sigma(\ell)$ for some constant $\alpha > 0$, and left as an open question whether the inapproximability factor σ can be improved to $\ell^{1-o(1)}$. Our main technical contribution is to answer this question affirmatively.

Theorem 1.8. *For every sufficiently large $\ell \in \mathbb{N}$, $\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{\sigma(\ell)}\mathsf{F}[\ell]$-CMMSA is NP-hard under randomized many-one reductions for some functions $\gamma(\ell) = o(1)$, $\varepsilon(\ell) = o(1)$, and $\sigma(\ell) = \ell^{1-o(1)}$.*

We also present an upper bound of coAM when $\sigma(\ell) = O(\ell)$, and thus the inapproximability factor $\sigma(\ell)$ of Theorem 1.8 is *optimal* up to the $o(1)$ loss in the exponent unless $\mathsf{NP} \subseteq \mathsf{coAM}$.

Theorem 1.9. *For every constant $\gamma \in [0, 1/5)$, there exists a constant C such that for every $\varepsilon(\ell) = o(1)$ and every large enough $\ell \in \mathbb{N}$, the following hold.*

- $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{\sigma(\ell)}\mathsf{F}[\ell]$-CMMSA $\in \mathsf{coAM}$, *where* $\sigma(\ell) \coloneqq C \cdot \ell$.
- $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{\sigma(\ell)}\mathsf{Junta}[\ell]$-CMMSA $\in \mathsf{coAM}$, *where* $\sigma(\ell) \coloneqq 2^{0.585\ell}$ *and* $\mathsf{Junta}[\ell] = \{\mathsf{Junta}[\ell]\}_{n\in\mathbb{N}}$ *denotes the class of all the ℓ-junta monotone functions* $\varphi\colon \{0,1\}^n \to \{0,1\}$.

In particular, for every $\gamma(\ell) = o(1)$, $\varepsilon(\ell) = o(1)$ and every large enough $\ell \in \mathbb{N}$, we have $\mathrm{Gap}^{\varepsilon,\gamma}_{C\cdot\ell}\mathsf{F}[\ell]$-CMMSA $\in \mathsf{coAM}$.

More generally, for any class $\mathfrak{C} = \{\mathfrak{C}_n\}_{n\in\mathbb{N}}$ of monotone functions, we prove $\mathrm{Gap}^{\varepsilon,\gamma}_{O(q)}\mathfrak{C}$-CMMSA $\in \mathsf{coAM}$, where q is an upper bound of the total share size of a secret sharing scheme for $\mathfrak{C}_n$. The first item of Theorem 1.9 follows from the fact that there exists a secret sharing scheme of total share size ℓ for every access structure represented by a monotone formula with ℓ literals [BL88]. The second item follows from the work of [AN21], which shows the existence of a secret sharing scheme of total share size $1.5^{\ell+o(\ell)} < 2^{0.585\ell}$ for every monotone function over ℓ variables.

Although it is unlikely that $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma=O(\ell)}\mathsf{F}[\ell]$-CMMSA is NP-hard, we prove that its worst-case hardness implies the errorless average-case hardness of NP, and that its errorless average-case hardness implies the existence of a one-way function.

Theorem 1.10. *For every $\sigma \in \mathbb{N}$, let $\mathfrak{C}[\sigma] = \{\mathfrak{C}[\sigma]_n\}_{n\in\mathbb{N}}$ be the class of monotone functions for which there exists a secret sharing scheme with total share size σ. For every constant $\gamma \in [0, 1/5)$, there exists a constant C such that for every $\varepsilon(\sigma) = o(1)$ and every sufficiently large $\sigma \in \mathbb{N}$, the following hold.*

- *If* $\mathrm{Gap}^{\varepsilon,\gamma}_{C\cdot\sigma}\mathfrak{C}[\sigma]$-CMMSA $\notin \mathsf{BPP}$, *then* $\mathsf{DistNP} \nsubseteq \mathsf{AvgBPP}$ (*i.e., there exists a problem in NP that is hard on errorless average with respect to some polynomial-time samplable distribution*).

- *If* $(\mathrm{Gap}^{\varepsilon,\gamma}_{C\cdot\sigma}\mathfrak{C}[\sigma]\text{-CMMSA}, \mathcal{D}) \notin \mathsf{AvgBPP}$ *for some polynomial-time samplable distribution* $\mathcal{D}$*, then an infinitely-often one-way function exists.*[2]

Here, *errorless average-case* means that an algorithm must not output incorrect answers, and is allowed to output a special symbol "$\perp$", which indicates the failure of an algorithm for a small fraction of inputs. This notion is equivalent to average-case polynomial-time (i.e., the expected running time is "polynomially bounded") [BT06a].

Previously, Hirahara [Hir22] reduced $\mathrm{Gap}^{0,\gamma}_{\sigma}\mathsf{F}[\ell]$-CMMSA to MINLT, and using the learning algorithm of Hirahara and Nanashima [HN21], observed that its worst-case hardness implies the average-case hardness of NP if $\sigma(\ell) = 1.01\ell$ and the soundness error γ is sufficiently smaller than $1/(sn)$. The first item of Theorem 1.10 improves this result in that γ can be as large as a constant.[3]

Consequently, Theorems 1.8 to 1.10 reveal a sharp threshold at $\sigma(\ell) \approx \ell$ in the complexity of $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathsf{F}[\ell]$-CMMSA: It is NP-hard if $\sigma(\ell) \leq \ell^{1-o(1)}$; It is in coAM if $\sigma(\ell) \geq O(\ell)$, and its worst-case (resp. average-case) hardness implies the average-case hardness of NP (resp. the existence of one-way functions). This can be compared with the complexity of the shortest vector problem, which is the foundational problem for lattice-based cryptography [Pei16]. The complexity of the approximate version GapSVP of the shortest vector problem greatly depends on an approximate factor γ (see [Ben23] and references therein). It is NP-hard if $\gamma = O(1)$; It is in coAM if $\gamma = O(\sqrt{n/\log n})$; its worst-case hardness implies the average-case hardness of NP if $\gamma = O(n)$; It is in P if $\gamma = 2^{O(n \log\log n/\log n)}$. To compare this with the complexity of $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathsf{F}[\ell]$-CMMSA, our results provide a sharper threshold than GapSVP using polynomial-time reductions. (A sharper complexity landscape of GapSVP was recently revealed by considering exponential time complexities [ABBGKLPSV23].)

Just as in the case of the shortest vector problem, it remains unclear whether the upper bound of coAM in Theorem 1.9 can be improved to P. In Appendix A, we present a simple approximation algorithm based on linear programming that solves $\mathrm{Gap}^{0,1}_{\sigma}\mathsf{DNF}[\ell]$-CMMSA for $\sigma(\ell) = \ell$, where $\mathsf{DNF}[\ell]$ denotes the class of monotone DNF formulas over n variables with at most ℓ terms. (In particular, it solves $\mathrm{Gap}^{0,1}_{\sigma}\mathsf{Junta}[\ell]$-CMMSA for $\sigma(\ell) = 2^{\ell}$.) We pose the following algorithmic challenge of improving this approximation algorithm.

Open Question 1.11. Design a randomized algorithm M such that for every $\ell \in \mathbb{N}$, every $\varepsilon = \varepsilon(\ell) = o(1)$, and every polynomial-time samplable distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$, it holds that

- M decides every instance of $\mathrm{Gap}^{\varepsilon,1/8}_{\sigma}\mathsf{Junta}[\ell]$-CMMSA correctly with high probability over the internal randomness of M, where $\sigma = \sigma(\ell) \coloneqq 2^{0.6\ell}$.
- $\mathbb{E}_{x\sim\mathcal{D}_n}[t_M(x)^{\alpha}] \leq O(n)$ for some constant $\alpha > 0$, where $t_M(x)$ denotes the running time of M on input x.

Currently, it seems consistent with our knowledge that the answer to Open Question 1.11 is negative. A negative answer to Open Question 1.11 implies the existence of a one-way function by Theorem 1.10. To the best of our knowledge, this is the first construction of a one-way function based on the errorless average-case hardness of some CSP with respect to *arbitrary* polynomial-time samplable distributions. Related results were obtained in an exciting line of research [DLS14; DS16; Dan16; Vad17; DV21], where the average-case hardness of random CSPs with respect to *specific*

[2] An *infinitely-often* one-way function refers to one whose security holds for infinitely many security parameters. We remark that this result also applies to standard one-way functions (whose security holds for any sufficiently large security parameter) by assuming the average-case hardness for all sufficiently large instance sizes.

[3] Our result strictly improves the previous result because the constant C in Theorem 1.10 can be chosen to be 1.01 if $\gamma = o(1)$.

distributions was shown to imply the hardness of PAC learning. We also mention that one-way functions can be constructed from error-prone average-case hardness of Random SAT with respect to high-entropy distributions [IRS22; LP23a].

1.3 Perspective: Can We Rule Out Pessiland?

Since our results are quantitatively "close" to ruling out Pessiland, it is natural to ask how close we are to *actually* ruling out Pessiland. There has been a vast literature on barriers for ruling out Heuristica [FF93; BT06b; Wat12; Imp11; HN21; Vio05] and Pessiland [Wee06; Liv10; AGGM06; BB15; ABX08]. Below, we discuss the two barriers most relevant to our work.

The first barrier is the relativization barrier. Wee [Wee06] presented an oracle under which Pessiland exists, and thus non-relativizing proofs are necessary for ruling out Pessiland. In fact, Theorem 1.3 is non-relativizing (see [Ko91; Hir22]). Thus, it may be possible that the inapproximability factor $\sigma(\ell) = O(\ell)$ of Theorem 1.5 can be improved to $\sigma(\ell) = \ell^{1-o(1)}$ by relativizing proofs, which is sufficient for ruling out Pessiland. Whether there is a relativizing barrier for improving Theorem 1.5 or not is left as an important open question.

The second barrier is the nonadaptive (black-box) reduction barrier of Akavia, Goldreich, Goldwasser, and Moshkovitz [AGGM06]. They showed that any problem reducible to the task of inverting one-way functions via randomized nonadaptive reductions is in coAM, and in particular, no NP-complete problems can be reducible to the task of inverting one-way functions unless $\mathsf{NP} \subseteq \mathsf{coAM}$. In fact, our coAM upper bound for GapCMMSA (Theorem 1.9) is proved by combining their barrier with a reduction from GapCMMSA to the task of inverting one-way functions. Similarly, Theorem 1.5 is unlikely to be improved by *nonadaptive* reductions. More generally, we have the following barrier, based on [AGGM06].[4]

Theorem 1.12. *Let γ, C, ℓ, and ε be as in Theorem 1.5. Suppose there is a randomized polynomial-time parametric-honest nonadaptive reduction from a paddable language L to $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ for $\sigma(\ell) \coloneqq C \cdot \ell$. Then $L \in \mathsf{coAM}$.*

Here, a *parametric-honest* reduction is a reduction that produces a size parameter at least $n^{\Omega(1)}$ on inputs of length n. Almost all NP-hardness reductions in the literature of meta-complexity are parametric-honest and nonadaptive (see, e.g., [Hir23; Ila23] and references therein), and thus the barrier seems formidable. However, in the computational learning theory literature, *adaptive* reductions have been successfully employed in the context of boosting algorithms (e.g., [Sch90; Fel10; KK09; FS12]), and such techniques have been utilized in the literature of meta-complexity [HN21; GK23]. This suggests that adaptive reductions could offer a viable path toward overcoming this barrier.

2 Technical Overview

For technical convenience, we work with a slight generalization of CMMSA, called the *distributional* minimum monotone satisfying assignment (DMMSA). DMMSA is the same problem as CMMSA, except that the collection of monotone formulas is given as a *distribution* of monotone formulas, where the distribution is specified by a sampling circuit (see Definition 4.1 for the formal definition). In other words, CMMSA is a special case of DMMSA in which the distribution is uniform over a polynomially bounded set of formulas. Nevertheless, it is straightforward to observe that GapDMMSA can be reduced to GapCMMSA with a small loss in error parameters ε and γ via a

[4] We thank an anonymous reviewer for suggesting this result.

BPP-reduction that samples polynomially many formulas from the distribution in DMMSA to form a collection for CMMSA (see Proposition 4.3). Therefore, we will identify DMMSA with CMMSA below.

Our main theorems are established based on the following three reductions:

1. An improved reduction from NP to GapDMMSA;
2. Hirahara's reduction from GapDMMSA to GapLearn [Hir22];
3. A reduction from GapLearn to the task of inverting an auxiliary-input one-way function.

Here, an *auxiliary-input* function is a collection $f = \{f_z\}_{z\in\{0,1\}^*}$ of functions such that, for every $z, x \in \{0,1\}^*$, the value $f_z(x)$ can be uniformly computed in polynomial time (where z is referred to as the auxiliary input). The inversion task is to find a preimage of $f_z(r)$ for most r with respect to f_z for every $z \in \{0,1\}^*$ using a single inversion algorithm that takes $(z, f_z(r))$ as input.

The first reduction maps each 3-SAT instance to a Gap_σDMMSA instance whose formulas have leaf size at most ℓ and whose approximation factor is $\sigma(\ell) = \ell^{1-o(1)}$. The key input is the near-optimal k-CSP hardness theorem of Minzer and Zheng [MZ26], together with a monotone compilation of the star-projection constraints produced by their PCP. A detailed overview of the reduction, which establishes Theorem 1.8, is presented in Section 2.1.

The second reduction is essentially from the previous work [Hir22], where the relation between the total size of secret sharing schemes and the advice complexity of sampling was not explicitly addressed. We simply observe that the advice complexity of the sample distribution in the resulting learning problem is at most the total size of the secret sharing scheme for the authorized set induced by the formulas in the support of instances of DMMSA. For completeness, we provide a formal proof in Section 5. We derive Theorem 1.3 (and Corollary 1.4) from the first and second reductions.

The third reduction reduces Gap_σLearn to the task of inverting an auxiliary-input function, provided that the approximation factor σ is linear in the advice complexity of the sample distribution. The reduction is based on the theory of Solomonoff's inductive inference [Sol64a] and universal extrapolation developed by Impagliazzo and Levin [IL90] and Hirahara and Nanashima [HN23]. This third reduction establishes Theorem 1.5. Combined with the second reduction, it also yields Theorems 1.9 and 1.10. More details are provided in Section 2.2.

2.1 Improved NP-Hardness of GapDMMSA

We first explain the reduction that improves the hardness factor for GapDMMSA from the ℓ^α factor of Hirahara [Hir22] to $\ell^{1-o(1)}$. The parameter that matters for the later learning application is not the size of the original PCP instance, but the leaf size ℓ of the monotone formulas: in Hirahara's reduction from CMMSA/DMMSA to learning, this leaf size upper bounds the total size of the induced secret sharing scheme, and hence the advice complexity of sampling. Thus, to obtain a nearly sharp learning-theoretic threshold, the PCP-to-DMMSA reduction must lose almost nothing when measured against the leaf size.

The Bottleneck in the Previous PCP-to-Monotone Reductions

The reductions of Dinur and Safra [DS04] and Hirahara [Hir22] start from a PCP, viewed as a CSP, and encode candidate labels by monotone variables. For each CSP variable u and each alphabet symbol $a \in \Sigma_u$, they introduce a Boolean variable $z_{u,a}$; setting $z_{u,a} = 1$ means that the monotone assignment keeps a as a candidate label for u. Thus an arbitrary monotone assignment represents

a list $L_u \subseteq \Sigma_u$ of candidate labels for every CSP variable u, while an honest labeling corresponds to lists of size one.

For a local constraint $e = (u_1, \ldots, u_q)$ with predicate $f_e \colon \Sigma_{u_1} \times \cdots \times \Sigma_{u_q} \to \{0,1\}$, the literal DNF compilation makes one term for each accepting local tuple:

$$F_e^{\mathrm{DNF}} = \bigvee_{(a_1,\ldots,a_q) \in f_e^{-1}(1)} \left(z_{u_1,a_1} \wedge \cdots \wedge z_{u_q,a_q} \right).$$

This preserves completeness, since a satisfying CSP labeling selects one accepting tuple in every local test. The cost is that the leaf size of the local formula is proportional to $q \cdot |f_e^{-1}(1)|$: the compilation pays for accepting tuples, not merely for alphabet symbols.

The soundness side naturally becomes a list-decoding argument. If many monotone formulas are satisfied by lists whose occurrence-weighted average size is B, then choosing one label uniformly from each nonempty list gives a CSP labeling that satisfies a random arity-q constraint with probability roughly at least B^{-q}. Therefore a CSP with soundness ζ can only rule out list size $B \lesssim \zeta^{-1/q}$.

This is where the previous approach loses its sharpness. To make the DMMSA gap large, one wants a PCP whose soundness is extremely small as a function of its alphabet size R. However, the literal DNF compilation above is useful for a K-leaf formula only when $q \cdot |f_e^{-1}(1)| \lesssim K$. As the alphabet or arity is pushed to improve soundness, the number of accepting local tuples can also grow, and the gain from smaller PCP soundness is then eaten up by the larger leaf size. This is the DNF bottleneck behind the previous ℓ^{α}-type hardness factor.

The Minzer–Zheng PCP

To obtain the sharp reduction, we use the near-optimal k-CSP hardness theorem of Minzer and Zheng [MZ26], together with a structural property of their final PCP. In the form used in the proof, for every $m \geq 2$, every completeness error $\delta > 0$, and every soundness-slack parameter $\xi > 0$, sufficiently large alphabet bound R gives NP-hardness of distinguishing an $(m+1)$-CSP of value at least $1 - \delta$ from one of value at most

$$R^{-(1-\xi)m}.$$

Here ξ measures the loss in the soundness exponent; in the final parameter setting we take $\xi = 1/m^2$.

This alphabet-soundness tradeoff is exactly what the list-decoding calculation above asks for. If the local tests could be compiled into monotone formulas of leaf size about R, then the CSP soundness $\zeta = R^{-(1-\xi)m}$ would turn into a DMMSA weight gap

$$\zeta^{-1/(m+1)} = R^{(1-\xi)m/(m+1)}.$$

With $\xi = 1/m^2$, the exponent is $1 - 1/m$, and hence this is $R^{1-o(1)}$ as $m \to \infty$. Thus the alphabet size is the right scale for the desired leaf-size-versus-gap tradeoff.

The remaining issue is to preserve this alphabet-size scale when converting the resulting CSP into monotone formulas. Our reduction avoids the extra loss incurred in the previous monotone-compilation step by exploiting the star-projection representation of the final constraints.

A star-projection constraint has one center variable y, leaf variables $x_1, \ldots, x_m$, and projection maps $\pi_{e,i} \colon \Sigma_{x_i} \to \Sigma_y$. It accepts a local labeling $(b, a_1, \ldots, a_m)$ exactly when

$$\pi_{e,i}(a_i) = b \qquad \text{for every } i \in [m].$$

Minzer and Zheng state their theorem for general $(m+1)$-CSPs. In Section 4.3, we observe that the constraints of the final PCP constructed in their proof have the star-projection form, although this is not stated explicitly in [MZ26]. We use this form to compile each constraint into a monotone formula with at most $(m+1)R$ leaves.

Compiling Star-Projection Constraints into Monotone Formulas

We now give the compilation, using the variables $z_{u,a}$ introduced above. Consider a star-projection constraint $e = (y; x_1, \ldots, x_m)$. A monotone assignment represents lists $L_u = \{a : z_{u,a} = 1\}$, and the local formula for e asks whether these lists contain a center label b and compatible leaf labels projecting to b.

For $b \in \Sigma_y$ and $i \in [m]$, define

$$P_{e,i,b} := \{a \in \Sigma_x : \pi_{e,i}(a) = b\}.$$

Then the formula is

$$F_e := \bigvee_{b \in \Sigma_y} \left(z_{y,b} \wedge \bigwedge_{i \in [m]} \left(\bigvee_{a \in P_{e,i,b}} z_{x_i,a} \right) \right),$$

where an empty inner OR is interpreted as the constant 0.

This formula is monotone because it only checks positive membership in the chosen lists. More importantly, it is small. The center literals contribute at most $|\Sigma_y| \le R$ leaves. Fix a leaf variable x_i. Each label $a \in \Sigma_x$ can belong to $P_{e,i,b}$ for at most one value of b: if $a \in P_{e,i,b} \cap P_{e,i,b'}$, then we have $b = \pi_{e,i}(a) = b'$. Hence all inner ORs associated with this x contribute at most $|\Sigma_x| \le R$ leaves in total. Since the number of leaf variables is at most m, the total leaf size is at most $(m+1)R$.

A Tight Leaf-Size Gap

Completeness is immediate. If a labeling satisfies the original constraint, selecting exactly that one label for each CSP variable gives a DMMSA assignment of weight s, and the branch $b = A(y)$ satisfies F_e.

The soundness proof shows that the factorized compilation loses essentially only the factor forced by list decoding. Suppose a DMMSA assignment has weight $B \cdot s$, where s is the weight of an honest one-label-per-variable assignment. Equivalently, if $s_u = |L_u|$, the occurrence-weighted average list size is B. If this assignment satisfies a large fraction of the formulas F_e, then for many satisfied constraints the local sum

$$s_y + s_{x_1} + \cdots + s_{x_m}$$

is $O((m+1)B)$ by Markov's inequality.

On such a satisfied constraint, the formula F_e gives a center label $b \in L_y$ and, for every leaf variable x_i, a label $a_{x_i} \in L_x$ such that

$$\pi_{e,i}(a_{x_i}) = b \qquad \text{for every } i \in [m].$$

If we decode by choosing a random label from each list, this particular consistent choice is selected with probability at least

$$\frac{1}{s_y \prod_{i=1}^{m} s_{x_i}} \gtrsim B^{-(m+1)},$$

where the last step is AM–GM. Thus, if the original instance has soundness at most ζ, any DMMSA assignment satisfying many formulas must have

$$B \gtrsim \zeta^{-1/(m+1)}.$$

Plugging in the original soundness with $\xi = 1/m^2$ gives

$$B \gtrsim R^{(1-1/m^2)m/(m+1)} = R^{1-1/m} = R^{1-o(1)}.$$

We set $R \approx K/(m+1)$ so that the compiled formula has leaf size at most $(m+1)R \leq K$. Since m grows only subpolynomially in K, the gap $R^{1-o(1)}$ is still $K^{1-o(1)}$. Finally, we take a small AND product of independent formulas to reduce the NO-side satisfaction threshold to $o(1)$. This multiplies the leaf size only by a subpolynomial factor, so after setting the base leaf bound slightly below the final parameter ℓ, the final gap remains

$$\sigma(\ell) = \ell^{1-o(1)}.$$

This is the tight connection we need: the alphabet size controls both the monotone leaf size and, through the exponent in the soundness bound, the DMMSA approximation gap.

2.2 A Reduction from Learning to Inverting Auxiliary-Input Functions

Theorems 1.5, 1.9 and 1.10 are based on a reduction from $\mathrm{Gap}_\sigma\mathrm{Learn}$ with $\sigma = O(\ell)$ to inverting an auxiliary-input function, where ℓ denotes the advice complexity. The reduction builds upon the theory of universal extrapolation [IL90; HN23].

In the reduction, we consider a specific case where a distribution $\mathcal{E}$ of samples for GapLearn is represented by two strings: nonuniform advice z and a randomly selected advice z'. Specifically, $\mathcal{E}$ is represented as $\mathcal{E} := \mathcal{D}_{z,z'}$ for some samplable distribution family $\mathcal{D} = \{\mathcal{D}_{z,z'}\}$. The goal, then, is to reduce GapLearn for $\mathcal{D}_{z,z'}$ to inverting an auxiliary-input function f_z indexed by z on average over z'. Here z and z' serve different roles in the context. For instance, in GapLearn instances derived from Hirahara's reduction from DMMSA to learning, z corresponds to the DMMSA instance, and z' represents randomness used in the reduction.

The advantage of treating z and z' separately is that the description size of the inverter depends only on z and not on z'. This enables us to achieve a good approximation factor for GapLearn that is independent of any randomly selected nonuniform advice z' embedded in the underlying distribution $\mathcal{E}$.

Specifically, Hirahara and Nanashima [HN23] showed that to learn $\mathcal{D}_{z,z'}$ from independent samples, it suffices to extrapolate i.i.d. samples (treated as a prefix string) according to the universal distribution $\mathrm{Q}^t|z$ for a sufficiently large polynomial t. Here, $\mathrm{Q}^t_{|z}$ represents the distribution of the output of a fixed universal Turing machine U executed in t steps on a randomly selected input $\Pi \sim \{0,1\}^t$ given the advice string z. This extrapolation task can be reduced to inverting a polynomial-time computable function f_z indexed by z, resulting in the reduction from learning to inverting an auxiliary-input function $f = \{f_z\}$.

When there exists a hypothesis of size s with $o(1)$ accuracy error, the extrapolation-based learner succeeds using at most $m(s) = O(s)$ samples with high probability over the choice of z', by the chain rule for KL divergence (see Lemmas 6.5 and 6.7). We use this sample-complexity upper bound to distinguish, for a given input z, between the following two cases: (i) $\mathcal{D}_{z,z'}$ is a yes instance for most z', and (ii) $\mathcal{D}_{z,z'}$ is a no instance for a noticeable fraction of z'.

More concretely, given access to an inverter for the auxiliary-input function, we construct a tester that checks whether the extrapolation-based learner succeeds using $m(s)$ samples with high probability over the choice of z' and the samples drawn from $\mathcal{D}_{z,z'}$. This is feasible because the tester can generate samples from $\mathcal{D}_{z,z'}$ on its own, by choosing z' itself and then executing the sampling procedure using the inverter. The tester accepts z if this success test passes. By the sample-complexity guarantee above, if $\mathcal{D}_{z,z'}$ is a yes instance of GapLearn for most z', then the tester accepts z with high probability. This proves completeness.

For soundness, we must show the converse direction: if the tester accepts z with high probability, then case (ii) cannot occur. It is enough to prove that in this situation, for most z', there exists a highly accurate hypothesis h for $\mathcal{D}_{z,z'}$ whose description size is at most $O(\ell) \cdot s$. Indeed, such a

hypothesis certifies that $\mathcal{D}_{z,z'}$ is a yes instance for most z'. Thus, our remaining task is to construct such a hypothesis.

A natural first attempt is the following. Since acceptance of z means that the extrapolation-based learner succeeds with high probability when given $m(s)$ samples, one may try to hardwire into the learner both $m(s)$ independently drawn samples from $\mathcal{D}_{z,z'}$ and the inverter I_z for f_z, thereby obtaining a highly accurate hypothesis $\tilde{h}$. However, this yields only the bound $|\tilde{h}| \leq m(s) \cdot n + |I_z|$, where n is the bit-length of each sample. This is too large for our purposes, since in general the sample length n can be much larger than the advice complexity ℓ needed to generate samples from $\mathcal{D}_{z,z'}$. Therefore, this naive hardwiring argument does not suffice.

To reduce the description size of $\tilde{h}$ to a bound in terms of ℓ, we do not hardwire the entire training sample set itself. Instead, we hardwire a succinct description of the training set that exploits the advice complexity of the sampling procedure. Recall that the $m := m(s)$ samples are generated from m random seeds $\bar{r} = (r^1, \ldots, r^m)$, chosen independently of z and z', together with an advice string $\alpha_{z,z',\bar{r}}$ of length $m \cdot \ell$. By a standard probabilistic argument, there exists a choice of random seeds $\bar{r}$ such that, with high probability over the choice of z', the learner succeeds on the training set for $\mathcal{D}_{z,z'}$ generated using $\bar{r}$.

Now fix such a tuple $\bar{r}$. The property that "with high probability over the choice of z', the learner succeeds on the training set for $\mathcal{D}_{z,z'}$ generated using $\bar{r}$" is computable, albeit inefficiently, from z, $\bar{r}$, and the description of the learner; here the learner is determined by the inverter, whose description depends only on z. Hence, we may define $\bar{r}^*$ to be the lexicographically first tuple of seeds satisfying this property. Thus, the description size of $\bar{r}^*$ is at most $O(\mathrm{K}(z) + \log n)$, where $\mathrm{K}(z)$ denotes the Kolmogorov complexity of z. Moreover, by the choice of $\bar{r}^*$, with high probability over the choice of z', the learner succeeds on the training set for $\mathcal{D}_{z,z'}$ generated using $\bar{r}^*$.

We therefore hardwire $\bar{r}^*$ together with the corresponding advice string $\alpha_{z,z',\bar{r}^*}$, and thereby obtain a hypothesis $\tilde{h}_{z,z'}$ of description size at most

$$O\big(|\alpha_{z,z',\bar{r}^*}| + \mathrm{K}(\bar{r}^*)\big) \leq O(m(s) \cdot \ell + \mathrm{K}(z) + \log n)\,.$$

This bound is sufficient to establish Theorems 1.5, 1.9 and 1.10. Below, we highlight each case.

Theorem 1.10 (Item 1). First, we reduce $\mathrm{Gap}_{O(\ell)}$DMMSA for formulas whose induced authorized set admits secret sharing of total size ℓ to a GapLearn instance with distribution $\mathcal{D}_{z,z'}$ over samples and size parameter s, where z is the GapDMMSA instance and z' is a random seed, using Hirahara's reduction [Hir22] (see also Section 5). The key properties of this reduction are as follows: (i) the advice complexity for sampling from $\mathcal{D}_{z,z'}$ is at most ℓ, (ii) the size parameter s is polynomially larger than $|z|$, and (iii) the approximation factor of GapLearn for $\mathcal{D}_{z,z'}$ corresponds to that of GapDMMSA for z. Consequently, the description size of the hypothesis constructed above is at most $O(m(s) \cdot \ell + \mathrm{K}(z) + \log n) = O(\ell) \cdot s$.

Therefore, $\mathrm{Gap}_{O(\ell)}$Learn on the support of Hirahara's reduction can be reduced to inverting an auxiliary-input function, and $\mathrm{Gap}_{O(\ell)}$DMMSA is also reducible to the same inversion task. This implies that the worst-case hardness of $\mathrm{Gap}_{O(\ell)}$DMMSA yields the hardness of inverting an auxiliary-input function (often called an auxiliary-input one-way function), which is known to imply the errorless average-case hardness of NP [HS17; Nan21].

Theorem 1.10 (Item 2). The argument above shows that $\mathrm{Gap}_{O(\ell)}$DMMSA for an instance z is reducible to inverting a function f_z indexed by z. In particular, if $\mathrm{Gap}_{O(\ell)}$DMMSA is hard on average under a samplable distribution $\mathcal{D}$ over instances z, then f_z is hard to invert on average over $z \sim \mathcal{D}$, which yields a standard one-way function. Moreover, the reduced inversion task for f_z is testable for every z by empirically estimating the probability that an algorithm

outputs a valid inverse of $f_z(r)$ for a random seed r. This allows us to detect any error in the reduction on each input z, as observed in [Nan21; Hir23; HN24]. Consequently, we can base the existence of one-way functions on the *errorless* average-case hardness of $\mathrm{Gap}_{O(\ell)}\mathrm{DMMSA}$ under *any* samplable distribution.

Theorem 1.9. We have shown that solving $\mathrm{Gap}_{O(\ell)}\mathrm{DMMSA}$ for an instance z is reducible to inverting a function f_z indexed by z. This instance-wise reduction is called a fixed-auxiliary-input reduction. It is known that any promise problem Π that can be reduced to inverting an auxiliary-input function via a fixed-auxiliary-input nonadaptive *black-box* reduction is contained in coAM [AGGM06; ABX08]. Our reduction above is nonadaptive but *not* black-box, since it embeds the description of the inverter for the auxiliary-input function into the learning algorithm. Nevertheless, it uses the inverter in an almost black-box manner, with the only non-black-box aspect being the bound on the description size. For this description-restricted setting, we adapt the technique of [AGGM06] to obtain the coAM upper bound. Specifically, the oracle simulated by the coAM protocol in the prior work becomes description-restricted in the presence of randomness, and this randomness can be compressed together with the seed set $\bar{r}$. This yields a simulation of a description-restricted inverter within coAM, ensuring that our reduction applies. For more details, see Section 6.3.

Theorem 1.5. We outline the implication from the average-case hardness of GapLearn, with advice complexity ℓ for sampling, to the existence of a one-way function. Let $\mathscr{D} = \{\mathscr{D}_n\}_{n\in\mathbb{N}}$ be a family of distributions over distributions $\mathcal{E}$ of samples for learning. We apply the reduction above to $\mathcal{D}_{n,\mathcal{E}}$, where n indexes $\mathscr{D}$ and $\mathcal{E}$ is a description of a distribution drawn from $\mathscr{D}_n$, and reduce the task of finding the minimum description size s of a hypothesis for $\mathcal{D}_{n,\mathcal{E}}$ (on average over $\mathcal{E}$) to inverting f_n. In this setting, the inverter I_n for each f_n has description size $O(\log n)$, and the description size of the hypothesis $\tilde{h}$ constructed above is at most $O(m(s)\cdot \ell + \mathrm{K}(n) + \log n) = O(\ell)\cdot s$ (assuming $s \geq \log n$; otherwise, the minimum hypothesis can be found by brute-force search). Thus, solving $\mathrm{Gap}_{O(\ell)}\mathrm{Learn}$ on average over $\mathcal{E}\sim\mathscr{D}_n$ reduces to inverting $f = \{f_n\}$. If the former is average-case hard, then f is a one-way function.

The converse implication, from one-way functions to the average-case hardness of GapLearn, follows from the fact that pseudorandom functions can be constructed from any one-way function [HILL99; GGM86], which implies the average-case hardness of learning even when arbitrary polynomial-time computable hypotheses are allowed [Val84].

Organization of This Paper

The remainder of this paper is organized as follows. In Section 3, we review preliminary notions for the formal arguments. In Section 4, we formally define GapDMMSA and prove the NP-hardness result via the Minzer–Zheng star-projection PCP, completing the proof of Theorem 1.8. In Section 5, we review Hirahara's reduction [Hir22] from GapDMMSA to GapLearn, which, together with the reduction in Section 4, proves Theorem 1.3. In Section 6, we present the reduction from GapLearn to inverting an auxiliary-input function and discuss its implications, including the proofs of Theorems 1.5, 1.9 and 1.10. In Section 7, we mention open problems arising from this work.

3 Preliminaries

All logarithms are base 2 unless stated otherwise. We use ϵ to represent an empty symbol. We distinguish ϵ from ε and often use ε for a small parameter such as an accuracy error. Let $\langle , \rangle$ be a

(standard) pairing function that maps $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$.

We use the notation negl to represent some negligible function, i.e., for any polynomial p and sufficiently large $n \in \mathbb{N}$, it holds that $\mathsf{negl}(n) < 1/p(n)$. We also use the notation poly to refer to some polynomial.

For each $n \in \mathbb{N}$, let $[n] := \{1, 2, \ldots, n\}$. For every $x, y \in \{0,1\}^*$, let $x \circ y$ denote the concatenation of x and y. For readability, we may omit $\circ$ from $x \circ y$. For each $x \in \{0,1\}^n$ and each $i \in [n]$, we let x_i denote the i-th bit of x. For every $x \in \{0,1\}^n$, let $x_{[k]} = x_1 \circ \cdots \circ x_k$ and $x_{[k:k']} = x_k \circ \cdots \circ x_{k'}$ for each $k \leq k' \leq n$.

For a function $f\colon \{0,1\}^n \to \{0,1\}^m$ and $y \in \mathrm{Im} f$, we define $f^{-1}(y) = \{x \in \{0,1\}^n : f(x) = y\}$.

For each $p \in [0,1]$, let $\mathrm{Ber}(p)$ be a Bernoulli distribution with parameter p. For any distribution $\mathcal{D}$, we use the notation $x \sim \mathcal{D}$ to refer to the sampling of x according to $\mathcal{D}$. For any finite set S, we use the notation $x \sim S$ to refer to the uniform sampling of x from S. For any distribution $\mathcal{D}$ and $k \in \mathbb{N}$, let $\mathcal{D}^{\otimes k}$ denote the k product distribution whose marginal distribution is identical to $\mathcal{D}$.

In this paper, we assume basic knowledge of probability theory, including the union bound, Markov's inequality, Jensen's inequality, and Hoeffding's inequality. For an event E where trials to determine whether E occurs are repeated efficiently, we say that an algorithm M performs the empirical estimation of the probability that E occurs with accuracy error $\varepsilon \in [0,1]$ and confidence error $\delta \in [0,1]$ if M computes a value v with $\Pr[E] - \varepsilon \leq v \leq \Pr[E] + \varepsilon$ with probability at least $1 - \delta$ over trials. By Hoeffding's inequality, only $O(\varepsilon^{-2} \log \delta^{-1})$ are needed for such estimation.

For any distributions $\mathcal{D}$ and $\mathcal{E}$, let $\Delta_{\mathrm{tv}}(\mathcal{D}, \mathcal{E})$ denote the total variation distance between $\mathcal{D}$ and $\mathcal{E}$. Let $\mathrm{KL}(\mathcal{D}||\mathcal{E})$ represent the KL divergence between two distributions $\mathcal{D}$ and $\mathcal{E}$.

For every distribution $\mathcal{D}$ over $\{0,1\}^*$, every $x \in \{0,1\}^*$, and $k \in \mathbb{N}$, we use the notation $\mathsf{Next}_k(x; \mathcal{D})$ to refer to the conditional distribution of the k-bit prefix of a subsequent string of x selected according to $\mathcal{D}$. If x does not match any prefix in the support of $\mathcal{D}$, we regard $\mathsf{Next}_k(x; \mathcal{D})$ as the distribution of the empty symbol.

For every promise problem $\Pi = (\Pi_{\mathrm{yes}}, \Pi_{\mathrm{no}})$ and every $x \in \Pi_{\mathrm{yes}} \cup \Pi_{\mathrm{no}}$, we define $\Pi(x)$ as

$$\Pi(x) = \begin{cases} 1 & \text{if } x \in \Pi_{\mathrm{yes}} \\ 0 & \text{if } x \in \Pi_{\mathrm{no}}. \end{cases}$$

A family $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$ of distributions is said to be (polynomial-time) samplable if there exists a polynomial-time randomized algorithm D (called a sampling algorithm or a sampler for $\mathcal{D}$) such that, for each $n \in \mathbb{N}$, the distribution of $D(1^n)$ is statistically identical to $\mathcal{D}_n$. This definition can also be extended to a distribution family $\mathcal{D} = \{\mathcal{D}_z\}_{z \in \{0,1\}^*}$ indexed by binary strings z, where the polynomial-time sampler takes the binary index z as input to sample from $\mathcal{D}_z$.

We say that a randomized algorithm A solves a promise problem Π on errorless average over $\mathcal{D}$ with failure probability $\delta \in (0,1)$ if (1) A outputs $\Pi(x)$ or $\perp$ (which represents "failure") with probability at least $3/4$ over the choice of randomness for A for every $x \in \mathsf{Support}(\mathcal{D})$, and (2) the failure probability that $A(x)$ outputs $\perp$ overwhelmingly (i.e., with probability at least $3/4$) over the choice of $x \sim \mathcal{D}$ is bounded above by δ. We say that a distributional problem $(\Pi, \{\mathcal{D}_n\}_{n \in \mathbb{N}})$ has an *errorless* heuristic algorithm A with failure probability $\delta\colon \mathbb{N} \to (0,1)$ if for all $n \in \mathbb{N}$, the randomized algorithm A solves a promise problem Π on errorless average over $\mathcal{D}_n$ with failure probability $\delta(n)$. Let $\mathsf{Avg}_\delta\mathsf{BPP}$ be the class of distributional problems that have an *errorless* heuristic algorithm with failure probability $\delta(n)$. In this paper, we only consider distributional problems $(\Pi, \mathcal{D})$ for which $\mathcal{D}$ is samplable.

We define the leaf size of a formula as the total number of literals appearing in the formula. For $\ell \in \mathbb{N}$, let $\mathsf{F}[\ell]$ denote the set of monotone formulas of leaf size at most ℓ.

3.1 Cryptography and Secret Sharing

We review some notations in cryptography.

Definition 3.1 (One-way function)**.** *A polynomial-time-computable function $f = \{f_n \colon \{0,1\}^{\mathsf{poly}(n)} \to \{0,1\}^{\mathsf{poly}(n)}\}_{n\in\mathbb{N}}$ is said to be an infinitely-often one-way function if for every polynomial-time randomized algorithm A, there exist infinitely many $n \in \mathbb{N}$ such that*

$$\Pr_{r,A}[f_n(A(1^n, f_n(r)) = f_n(r)] < \mathsf{negl}(n),$$

where $r \sim \{0,1\}^{\mathsf{poly}(n)}$ is a random seed.

The parameter n in the definition above is often referred to as a *security parameter*. For readability, we may omit the subscript n from f_n and 1^n from the input to adversaries. Note that our result for constructing an infinitely-often one-way function based on average-case hardness also extends to the standard one-way function, whose security holds for all sufficiently large security parameters, assuming average-case hardness on all sufficiently large instance sizes.

Next, we introduce an auxiliary-input variant of one-way functions, introduced by Ostrovsky and Wigderson [OW93]. Roughly speaking, auxiliary-input primitives are defined as a collection of candidates for secure primitives indexed by an auxiliary input $z \in \{0,1\}^*$ and have a relaxed security condition that for each adversary A, there exists an auxiliary input $z_A \in \{0,1\}^*$ depending on A such that the primitive indexed by z_A is secure for A.

We define an auxiliary-input function as a function family $f = \{f_z\}_{z\in\{0,1\}^*}$ indexed by binary strings z. We say that f is polynomial-time computable if each $f_z(x)$ is polynomial-time computable from (z, x).

Definition 3.2 (Auxiliary-input one-way function)**.** *A polynomial-time computable auxiliary-input function $f = \{f_z \colon \{0,1\}^{\mathsf{poly}(|z|)} \to \{0,1\}^{\mathsf{poly}(|z|)}\}_{z\in\{0,1\}^*}$ is said to be an auxiliary-input one-way function if for every polynomial-time randomized algorithm A, there exist infinitely many $z \in \{0,1\}^*$ such that*

$$\Pr_{r,A}[f_z(A(z, f_z(r))) = f_z(r)] < \mathsf{negl}(|z|),$$

where $r \sim \{0,1\}^{\mathsf{poly}(|z|)}$ is a random seed.

It is known that the existence of auxiliary-input one-way functions implies the errorless average-case hardness of NP.

Proposition 3.3 ([HS17; Nan21])**.** *If there exists an auxiliary-input one-way function, then there exist $\Pi \in \mathsf{NP}$, samplable distribution $\mathcal{D}$, and a polynomial p such that $(\Pi, \mathcal{D}) \notin \mathsf{Avg}_{1/p}\mathsf{BPP}$.*

We also review the notion of secret sharing. For each $n \in \mathbb{N}$, we define an authorized set $\mathcal{A} \subseteq [n]$ over n parties as any monotone set system over $[n]$, i.e., $S \in \mathcal{A}$ and $S \subseteq T$ imply $T \in \mathcal{A}$. Namely, every monotone function $f \colon \{0,1\}^n \to \{0,1\}$ induces an authorized set $\mathcal{A}_f$ as

$$\mathcal{A}_f = \{S \subseteq [n] : f(\chi_S) = 1\},$$

where $\chi_S \in \{0,1\}^n$ is the characteristic string for S.

Definition 3.4 (Secret sharing)**.** *A secret sharing scheme for an authorized set $\mathcal{A}$ over n parties is a pair of a polynomial-time randomized algorithm Share and a polynomial-time deterministic algorithm Rec such that*

- Share *takes a bit and then outputs* n *shares* $s_1, \ldots, s_n \in \{0,1\}^*$*;*
- Rec *takes* n *strings and outputs a bit;*
- (*Completeness*) *For every* $b \in \{0,1\}$*, every* $S \in \mathcal{A}$*, and every* $s'_1, \ldots, s'_n \in \{0,1\}^*$*,*

$$\Pr_{s_1,\ldots,s_n}[\mathsf{Rec}(s'_1, \ldots, s'_n) = b \mid s'_i = s_i \; \forall i \in S] = 1,$$

  where $(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(b)$*;*
- (*Security*) *For every unauthorized set* $S \notin \mathcal{A}$*, the shares held by* S *are perfectly independent of the shared bit. Namely,*

$$\{(s_i)_{i\in S} : (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(0)\} \equiv \{(s_i)_{i\in S} : (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(1)\},$$

  where $\equiv$ *denotes identical distributions. In particular, no randomized predictor given the shares of an unauthorized set, together with any randomness independent of the shared bit, can predict the shared bit with probability different from* $1/2$*.*

We say that an authorized set $\mathcal{A}$ *admits secret sharing of total size* s *if it holds that* $|s_1|+\cdots+|s_n| \le s$ *for every bit* b *and randomness for* Share*.*

We extend the definition above to a collection of monotone formulas. We say that a collection F of monotone formulas admits secret sharing of total size s if there exist polynomial-time algorithms Share and Rec such that for every $\varphi \in F$, $(\mathsf{Share}(\text{-};\varphi), \mathsf{Rec}(\text{-};\varphi))$ is a secret sharing scheme for the induced authorized set $\mathcal{A}_\varphi$ of total size at most s.

Theorem 3.5 ([ISN93; BL88])**.** *Any monotone formula of leaf size* ℓ *admits a secret sharing scheme for the induced authorized set of total size at most* ℓ*.*

Let $\mathfrak{S}_\ell$ be the set of monotone formulas that admits a secret sharing scheme for the induced authorized set of total size at most ℓ. Then, the above shows that $\mathfrak{S}_\ell$ contains every monotone formula of leaf size ℓ, i.e., $\mathsf{F}[\ell] \subseteq \mathfrak{S}_\ell$.

3.2 Algorithmic Information

We fix a prefix-free universal Turing machine U with an auxiliary-input tape. The output of the universal Turing machine on input d and auxiliary input y is denoted by $U(d;y)$. We assume that U is prefix-free, i.e., for any two strings d_1 and d_2, if d_1 is a prefix of d_2 and $U(d_1;y)$ halts, then $U(d_2;y)$ does not halt. We also assume that U is universal; i.e., for every Turing machine M and every string p, there exists a description p_M such that for every string y,

$$U(p_M;y) = M(p;y), \qquad |\widehat{p}_M| \le |p| + 2\lceil \log(|p|+1)\rceil + O_M(1),$$

whenever $M(p;y)$ halts. For each $t \in \mathbb{N}$ and input Π, we let $U^t(\Pi)$ denote the outcome obtained when we execute U on input Π only in t steps.

We define the Kolmogorov complexity as follows.

Definition 3.6 (Kolmogorov complexity)**.** *For every* $t \in \mathbb{N}$ *and every* $x, z \in \{0,1\}^*$*, we define the* t*-time-bounded Kolmogorov complexity of* x *given* z *as* $\mathrm{K}^t(x|z) = \min_{\Pi\in\{0,1\}^*}\{|\Pi| : U^t(\Pi;z) = x\}$*. We also define the* (*time-unbounded*) *Kolmogorov complexity of* x *given* z *as* $\mathrm{K}(x|z) = \lim_{t\to\infty} \mathrm{K}^t(x|z)$*. We omit the description "*$|z$*" if* z *is the empty string, i.e.,* $\mathrm{K}^t(x) = \min_{\Pi\in\{0,1\}^*}\{|\Pi| : U^t(\Pi) = x\}$*.*

We also extend the definition above to a finite set S of strings. For $S = \{s_1, \ldots, s_k\}$ and a string z, we define $\mathrm{K}(S|z) = \mathrm{K}(\langle s_1, \ldots, s_k\rangle|z)$.

For any $k \in \mathbb{N}$ and any $x, z \in \{0,1\}^*$ with $|z| = k|x|$, let $\mathsf{DP}_k(x; z) := z \circ \langle x, z^1\rangle_{\mathbb{F}_2} \circ \cdots \circ \langle x, z^k\rangle_{\mathbb{F}_2} \in \{0,1\}^{|z|+k}$, where $z = z^1 \circ \cdots \circ z^k$, $|z^i| = |x|$ for each i, and $\langle,\rangle_{\mathbb{F}_2}$ represents the inner product in $\mathbb{F}_2$.

We introduce Algorithmic Information Extraction Lemma, presented in [Hir22]. For m strings $f_1, \ldots, f_m \in \{0,1\}^\lambda$ and a subset $B \subseteq [m]$, we use the notation f_B to refer to $\langle f_{i_1}, \ldots, f_{i_{|B|}}\rangle$, where $B = \{i_1, \ldots, i_{|B|}\}$ $(i_1 < \ldots < i_{|B|})$.

Lemma 3.7 (Algorithmic Information Extraction Lemma [Hir22])**.** *For any $a, k, \varepsilon^{-1}, \lambda, m \in \mathbb{N}$, any $f_1, \ldots, f_m \in \{0,1\}^\lambda$, and any function $D\colon \{0,1\}^a \times (\{0,1\}^{\lambda k+k})^m \to \{0,1\}$, there exists a set $B \subseteq [m]$ such that*

$$\mathrm{K}^D(f_B) \leq |B| \cdot (mk + a + O(\log m\lambda a\varepsilon^{-1})),$$

and for every $\alpha \in \{0,1\}^a$,

$$\left|\Pr[D(\alpha, X_1, \ldots, X_m) = 1] - \Pr[D(\alpha, X'_1, \ldots, X'_m) = 1]\right| \leq \varepsilon.$$

Here, X_i is a random variable chosen according to $\mathsf{DP}_k(f_i; z_i)$ for $z_i \sim \{0,1\}^{\lambda k}$, and X'_i is a random variable identical to X_i if $i \in B$; otherwise (i.e., if $i \in [m] \setminus B$) selected according to the uniform distribution over $\{0,1\}^{\lambda k+k}$.

We also use the following fact.

Fact 3.8 ([cf. CT06])**.** *There exists a polynomial τ such that for each $p \in (0, 1/2)$ and each $x \in \{0,1\}^n$ with $\mathsf{wt}(x) \leq pn$,*

$$\mathrm{K}^{\tau(n)}(x) \leq O(H(p)n + \log n) \leq O(p^{0.75}n + \log n),$$

where $H(\cdot)$ represents the binary entropy function.

We define the universal probability and computational depth. For every $t \in \mathbb{N}$ and every string y, we define the distribution $\mathrm{Q}^t_{|y}$ as a distribution of $U^t(\Pi, y)$ for $\Pi \sim \{0,1\}^t$. When $y = \epsilon$, we omit the subscript "$|y$".

For every $t \in \mathbb{N}$ and every string $x, y \in \{0,1\}^*$,

$$\mathrm{q}^t(x|y) = -\log \Pr_{\Pi\sim\{0,1\}^t}\left[U^t(\Pi, y) = x\right].$$

For every $t \in \mathbb{N}$ and $x \in \{0,1\}^*$, the t-time-bounded computational depth $\mathrm{cd}^y(x|y)$ of x given $y$is defined as

$$\mathrm{cd}^t(x|y) := \mathrm{q}^t(x|y) - \mathrm{K}(x|y).$$

The following well-known fact shows that the t-time-bounded computational depth of a sample drawn from $\mathcal{D}$ is logarithmically small with high probability when t is sufficiently larger than the time complexity required to sample from $\mathcal{D}$.

Lemma 3.9 ([cf. HN23])**.** *For every samplable distribution $\mathcal{D} = \{\mathcal{D}_z\}_{z\in\{0,1\}^*}$, for any large enough polynomial τ, it holds that for every $z \in \{0,1\}^*$, every $i \in \mathbb{N}$, and every $t \geq \tau(|z|)$,*

$$\Pr_{x\sim\mathcal{D}_z}[\mathrm{cd}^t(x_{[i]}|z) \leq 2\log t] \geq 1 - \frac{1}{t}.$$

We derive the following lemma from the above.

Lemma 3.10. *For every samplable distribution $\mathcal{D} = \{\mathcal{D}_z\}_{z\in\{0,1\}^*}$, every polynomial p, for any large enough polynomial τ, for every long enough $z \in \{0,1\}^*$, every $i \in \mathbb{N}$, every $t \geq \tau(|z|)$, and every event E determined by $x \sim \mathcal{D}_z$ such that $\Pr_x[E] \geq 1/p(|z|)$, it holds that*

$$\mathbb{E}_{x\sim\mathcal{D}_z}[\mathrm{cd}^t(x_{[i]}|z)|E] \leq 5\log t.$$

Proof. We select a large enough polynomial τ such that it satisfies the conditions in Lemma 3.9. We also assume that $\tau(z)$ is larger enough than the time bound for sampling according to $\mathcal{D}_z$, and for each $x \in \mathsf{Support}(\mathcal{D}_z)$, it holds that $\mathrm{cd}^t(x_{[i]}|z) \leq |x| + O(1) \leq t$ (for each i and each $t \geq \tau(|z|)$). Further, we assume that $t \geq p(|z|)$ for each $t \geq \tau(|z|)$. We use Lemma 3.9 and obtain that

$$\Pr_{x\sim\mathcal{D}_z}[E \wedge \mathrm{cd}^t(x_{[i]}|z) > 4\log t] \leq \Pr_{x\sim\mathcal{D}_z}[\mathrm{cd}^t(x_{[i]}|z) > 4\log t] \leq t^{-2}.$$

Thus,

$$\Pr_{x\sim\mathcal{D}_z}[\mathrm{cd}^t(x_{[i]}|z) > 4\log t|E] \leq \frac{1}{t^2\Pr[E]} \leq \frac{p(|z|)}{t^2} \leq t,$$

and

$$\mathbb{E}_{x\sim\mathcal{D}_z}[\mathrm{cd}^t(x_{[i]}|z)|E] \leq 1\cdot(4\log t) + \frac{t}{t} \leq 5\log t.$$

□

3.3 Sampling with Advice

We consider the notion of the advice complexity of sampling as a parameter of problems.

Definition 3.11 (Advice complexity of sampling)**.** *A distribution $\mathcal{D}$ over $\{0,1\}^n$ is said to be* samplable with advice complexity ℓ *if there exists a function $\alpha\colon \{0,1\}^{2^{2n}} \to \{0,1\}^{\leq\ell}$ such that $U(\alpha(r); r, n)$ halts for every r, outputs an element of $\{0,1\}^n \cup \{\bot\}$, and*

$$\Pr_{r\sim\{0,1\}^{2^{2n}}}[U(\alpha(r); r, n) = \bot] \leq \frac{1}{4}.$$

Moreover, conditioned on $U(\alpha(r); r, n) \neq \bot$, its output distribution is required to have statistical distance at most 2^{-2n} from $\mathcal{D}$.

The equivalence to the definition presented in the introduction is discussed in Appendix B.

4 NP-Hardness of Distributional Minimum Monotone Satisfying Assignment

In this section, we show the NP-hardness of DMMSA and Theorem 1.8.

4.1 Distributional Minimum Monotone Satisfying Assignment

First, we formally introduce the Distributional Minimum Monotone Satisfying Assignment problem.

Definition 4.1 (Distributional Minimum Monotone Satisfying Assignment)**.** *Let $\mathfrak{C}$ be a set of monotone formulas. An instance of Distributional Minimum Monotone Satisfying Assignment for $\mathfrak{C}$ ($\mathfrak{C}$-DMMSA) over n variables has a pair of a distribution $\mathcal{D}$ of monotone formulas in $\mathfrak{C}$ over variables $X = \{x_1, \ldots, x_n\}$ and a weight function $w\colon X \to [0,1]$ with $\sum_{x\in X} w(x) = 1$.*

For a $\mathfrak{C}$-DMMSA instance $\mathcal{D}$ over n variables and threshold $\tau \in [0,1]$, we define the τ-value $\mathsf{val}_\tau(\mathcal{D}) \in [0,1]$ of $\mathcal{D}$ as

$$\mathsf{val}_\tau(\mathcal{D}) := \min\left\{a \in [0,1] : \exists \alpha \in \{0,1\}^n \text{ s.t. } w(\alpha) = a \text{ and } \Pr_{\varphi\sim\mathcal{D}}[\varphi(\alpha)=1] \geq \tau\right\},$$

where

$$w(\alpha) = \sum_{i:\alpha_i=1} w(x_i).$$

For $\tau_Y, \tau_N \in [0,1]$ and $\sigma, l \in \mathbb{N}$, we define the promise problem $\mathrm{Gap}_\sigma^{\tau_Y,\tau_N}\mathfrak{C}$-DMMSA as the following promise problem (Π_Y, Π_N):

$$\Pi_Y = \{(\mathcal{D}, w, 1^n, s) : \mathsf{val}_{1-\tau_Y}(\mathcal{D}) \leq s\}$$
$$\Pi_N = \{(\mathcal{D}, w, 1^n, s) : \mathsf{val}_{\tau_N}(\mathcal{D}) > \sigma \cdot s\},$$

where $n \in \mathbb{N}$, $s \in [0,1]$ is represented in polynomial size, and $(\mathcal{D}, w)$ is a $\mathfrak{C}$-DMMSA instance over n variables.

We also define the problem above when τ_Y, τ_N, σ are specified as functions in the same manner.

In this paper, we always assume that the weight function w and the size parameter s are lower bounded by the inverse of a polynomial in the number of variables, and that w and s are represented in polynomial size.

The main theorem in this section is stated as follows.

Theorem 4.2. *There exist functions $\sigma := \sigma(\ell) = \ell^{1-o(1)}, \varepsilon := \varepsilon(\ell) = o(1), \gamma := \gamma(\ell) = o(1)$ such that for every sufficiently large $\ell \in \mathbb{N}$, $\mathrm{Gap}_\sigma^{\varepsilon,\gamma}\mathsf{F}[\ell]$-DMMSA is NP-hard via a polynomial-time deterministic reduction.*

Recall that $\mathsf{F}[\ell]$ is the class of monotone formulas of leaf size at most ℓ. This immediately implies Theorem 1.8 because of the following simple proposition.

Proposition 4.3. $\mathrm{Gap}_\sigma^{\varepsilon,\gamma}\mathfrak{C}$-DMMSA *is reducible to* $\mathrm{Gap}_\sigma^{\varepsilon+1/p(m),\gamma+1/p(m)}\mathfrak{C}$-CMMSA *for every class $\mathfrak{C}$ of monotone formulas and every polynomial p via* BPP*-reduction, where m is the size of the original instance.*

Proof. Let $(\mathcal{D}, w, 1^n, s)$ be an arbitrary instance of $\mathrm{Gap}_\sigma^{\varepsilon,\gamma}\mathfrak{C}$-DMMSA. The BPP-reduction just selects $\Theta(p(m)^2 \cdot n)$ monotone formulas according to $\mathcal{D}$. Let C be a collection of the selected formulas. Then, we claim that $(C, w, 1^n, s)$ is an yes (resp. no) instance of $\mathrm{Gap}_\sigma^{\varepsilon+1/p(m),\gamma+1/p(m)}\mathfrak{C}$-CMMSA with probability at least $2/3$ over the choice of C.

By the Hoeffding's inequality, for every assignment α,

$$\left|\Pr_{\varphi\sim C}[\varphi(\alpha)=1] - \Pr_{\varphi\sim\mathcal{D}}[\varphi(\alpha)=1]\right| \leq 1/p(m).$$

with probability at least $1 - 1/(3\cdot 2^n)$ over the choice of C.

Thus, if the original instance is yes instance, then there exists an assignment such that $w(\alpha) \leq s$ and

$$\Pr_{\varphi\sim C}[\varphi(\alpha)=1] \geq 1 - \varepsilon - 1/p(m)$$

with probability at least $1 - 1/(3\cdot 2^n) \geq 2/3$. If the original instance is no instance, then by the union bound, there is no assignment of weight at most σs such that

$$\Pr_{\varphi\sim C}[\varphi(\alpha)=1] \geq \gamma + 1/p(m)$$

with probability at least $1 - 2^n/(3\cdot 2^n) = 2/3$. □

4.2 Proof of Theorem 4.2

We prove Theorem 4.2 from the following star-projection CSP hardness statement.

Definition 4.4 (Star-projection CSP)**.** *Let $m \geq 1$ and $R \geq 1$. A star-projection $(m+1)$-CSP instance Ψ consists of two disjoint sets of variables $\mathcal{X}$ and $\mathcal{Y}$, finite alphabets $\{\Sigma_u\}_{u \in \mathcal{X} \cup \mathcal{Y}}$ with $|\Sigma_u| \leq R$, a set of constraints E with a distribution μ, and, for each constraint*

$$e = (y; x_1, \ldots, x_m) \in E \qquad (y \in \mathcal{Y},\ x_i \in \mathcal{X}),$$

projection maps $\pi_{e,i} \colon \Sigma_{x_i} \to \Sigma_y$ for $i \in [m]$. A labeling A satisfies $e = (y; x_1, \ldots, x_m)$ if

$$\pi_{e,i}(A(x_i)) = A(y) \qquad \text{for every } i \in [m].$$

Let $\mathsf{val}(\Psi)$ denote the maximum, over all labelings A, of the probability that A satisfies a random constraint $e \sim \mu$.

Theorem 4.5 ([MZ26])**.** *For every integer $m \geq 2$ and every $\xi, \delta > 0$, there is an integer $R_\star(m, \xi, \delta)$ such that for every $R \geq R_\star(m, \xi, \delta)$, it is NP-hard to distinguish, given a star-projection $(m+1)$-CSP instance Ψ with alphabet size at most R, between the following two cases:*

$$\mathsf{val}(\Psi) \geq 1 - \delta \qquad \text{and} \qquad \mathsf{val}(\Psi) \leq R^{-(1-\xi)m}.$$

We now complete the reduction from star-projection CSPs to DMMSA, taking Theorem 4.5 as a black box. The proof that the Minzer–Zheng construction indeed yields the star-projection structure required by Theorem 4.5 is postponed to Section 4.3.

Reduction to DMMSA. Let Ψ be a star-projection $(m+1)$-CSP instance with alphabet size at most R. Put $U := \mathcal{X} \cup \mathcal{Y}$. We discard variables that have zero probability of appearing in a random constraint. For $u \in U$, define its occurrence weight by

$$\lambda(u) := \frac{1}{m+1} \mathbb{E}_{e=(y;x_1,\ldots,x_m)\sim\mu} \left[\mathbf{1}[y = u] + \sum_{i=1}^{m} \mathbf{1}[x_i = u] \right]. \tag{1}$$

Then $\lambda(u) > 0$ for every remaining u, and

$$\sum_{u \in U} \lambda(u) = 1. \tag{2}$$

For each $u \in U$ and $a \in \Sigma_u$, introduce a DMMSA variable $z_{u,a}$. We first assign the unnormalized weight

$$\widetilde{w}(z_{u,a}) := \lambda(u).$$

Let

$$\Lambda := \sum_{u \in U} \lambda(u) |\Sigma_u|,$$

and define the normalized DMMSA weight

$$w(z_{u,a}) := \frac{\widetilde{w}(z_{u,a})}{\Lambda} = \frac{\lambda(u)}{\Lambda}. \tag{3}$$

Thus $\sum_{u,a} w(z_{u,a}) = 1$. The size parameter in the resulting Gap-DMMSA instance will be

$$s := \frac{1}{\Lambda}. \tag{4}$$

For a Boolean assignment α to the variables $z_{u,a}$, write

$$L_u(\alpha) := \{a \in \Sigma_u : \alpha(z_{u,a}) = 1\}$$

for the list of labels selected for u.

We now define the distribution $\mathcal{D}_0$ over monotone formulas. Sample a constraint $e = (y; x_1, \ldots, x_m) \sim \mu$. Let X_e be the set of distinct variables appearing among $x_1, \ldots, x_m$, and for $x \in X_e$ let

$$I_e(x) := \{i \in [m] : x_i = x\}.$$

For $b \in \Sigma_y$ and $x \in X_e$, define

$$P_{e,x,b} := \{a \in \Sigma_x : \pi_{e,i}(a) = b \text{ for every } i \in I_e(x)\}.$$

The formula output by $\mathcal{D}_0$ on constraint e is

$$F_e := \bigvee_{b \in \Sigma_y} \left(z_{y,b} \wedge \bigwedge_{x \in X_e} \left(\bigvee_{a \in P_{e,x,b}} z_{x,a} \right) \right), \tag{5}$$

where an empty inner OR is interpreted as the constant 0.

Lemma 4.6. *The reduction above is polynomial time and satisfies the following properties.*

(a) Every formula in $\mathrm{supp}(\mathcal{D}_0)$ *is monotone and has leaf size at most* $(m+1)R$.

(b) (Completeness.) For every $\tau \in [0,1]$, *if* $\mathsf{val}(\Psi) \geq 1 - \tau$, *then*

$$\mathsf{val}_{1-\tau}(\mathcal{D}_0) \leq s.$$

(c) (Soundness.) If $\mathsf{val}(\Psi) \leq \zeta$, *then every assignment* α *of normalized weight*

$$w(\alpha) \leq \frac{1}{8} \left(\frac{5}{8} \right)^{1/(m+1)} \zeta^{-1/(m+1)} \cdot s$$

satisfies

$$\Pr_{F \sim \mathcal{D}_0}[F(\alpha) = 1] \leq \frac{3}{4}.$$

Proof. The formula in (5) is monotone by construction. We first count leaves. The center leaves contribute at most $|\Sigma_y| \leq R$. Fix $x \in X_e$ and a label $a \in \Sigma_x$. Since $I_e(x)$ is nonempty, the label a can belong to $P_{e,x,b}$ for at most one value of b: if $a \in P_{e,x,b} \cap P_{e,x,b'}$, then for any $i \in I_e(x)$ we get $b = \pi_{e,i}(a) = b'$. Hence the leaves associated with a fixed x contribute at most $|\Sigma_x| \leq R$. Since $|X_e| \leq m$, the total leaf size is at most $R + mR = (m+1)R$. This also gives polynomial-time constructibility of each local formula.

For completeness, let A be a labeling that satisfies a $1 - \tau$ fraction of constraints. Define the assignment α_A by setting $\alpha_A(z_{u,A(u)}) = 1$ for every $u \in U$ and all other variables to 0. Its unnormalized weight is

$$\sum_{u \in U} \lambda(u) = 1,$$

so its normalized weight is $1/\Lambda = s$. If A satisfies $e = (y; x_1, \ldots, x_m)$, then the branch $b = A(y)$ in (5) is satisfied: for each distinct $x \in X_e$, choosing $a_x = A(x)$ gives $\pi_{e,i}(a_x) = A(y) = b$ for

every $i \in I_e(x)$. Therefore $F_e(\alpha_A) = 1$ for every CSP constraint e satisfied by A, and hence $\Pr_{F\sim\mathcal{D}_0}[F(\alpha_A) = 1] \geq 1 - \tau$.

It remains to prove soundness. Let α be any assignment and put

$$s_u := |L_u(\alpha)|, \qquad W := \sum_{u\in U} \lambda(u) s_u.$$

Thus $W = \Lambda \cdot w(\alpha)$ is the unnormalized weight of α. We prove that if $\Pr_{F\sim\mathcal{D}_0}[F(\alpha) = 1] > 3/4$, then

$$W > \frac{1}{8}\left(\frac{5}{8}\right)^{1/(m+1)} \zeta^{-1/(m+1)}. \tag{6}$$

This implies the claimed contrapositive after dividing by Λ.

Assume $\Pr[F(\alpha) = 1] > 3/4$. If $W = 0$, then every variable appearing in the support of the constraint distribution has an empty list, and every sampled formula is unsatisfied. Hence $W > 0$. For a random constraint $e = (y; x_1, \ldots, x_m)$, define

$$T_e := s_y + \sum_{i=1}^{m} s_{x_i}.$$

By definition,

$$T_e = s_y + \sum_{i=1}^{m} s_{x_i} = \sum_{u\in U} s_u \left(\mathbf{1}[y = u] + \sum_{i=1}^{m} \mathbf{1}[x_i = u]\right),$$

where the occurrences among $x_1, \ldots, x_m$ are counted with multiplicity. Since the numbers s_u are fixed once the assignment α is fixed, taking expectation over $e \sim \mu$ gives

$$\begin{aligned}\mathbb{E}_{e\sim\mu}[T_e] &= \sum_{u\in U} s_u \mathbb{E}_{e=(y;x_1,\ldots,x_m)\sim\mu}\left[\mathbf{1}[y = u] + \sum_{i=1}^{m} \mathbf{1}[x_i = u]\right] \\ &= (m+1)\sum_{u\in U} \lambda(u) s_u = (m+1)W,\end{aligned}$$

where the second equality is exactly the definition of $\lambda(u)$ in (1).

Markov's inequality gives

$$\Pr_{e\sim\mu}[T_e > 8(m+1)W] \leq \frac{1}{8}.$$

Since $F_e(\alpha) = 1$ with probability $> 3/4$, the probability that both $F_e(\alpha) = 1$ and $T_e \leq 8(m+1)W$ hold is $> 5/8$. Call such a constraint good.

Construct a random labeling A_α as follows. For each $u \in U$, if $L_u(\alpha) \neq \emptyset$, choose $A_\alpha(u)$ uniformly from $L_u(\alpha)$; if $L_u(\alpha) = \emptyset$, choose an arbitrary fixed label. The choices are independent over u. Fix a good constraint $e = (y; x_1, \ldots, x_m)$. Since $F_e(\alpha) = 1$, there exist a label $b \in L_y(\alpha)$ and, for every $x \in X_e$, a label $a_x \in L_x(\alpha)$ such that

$$\pi_{e,i}(a_{x_i}) = b \qquad \text{for every } i \in [m].$$

If $A_\alpha(y) = b$ and $A_\alpha(x) = a_x$ for every $x \in X_e$, then A_α satisfies e. The probability of this event is

$$\frac{1}{s_y}\prod_{x\in X_e}\frac{1}{s_x} \geq \frac{1}{s_y \prod_{i=1}^{m} s_{x_i}},$$

where repetitions among $x_1, \dots, x_m$ can only increase the denominator on the right. By AM–GM and the goodness of e,

$$s_y \prod_{i=1}^{m} s_{x_i} \le \left(\frac{s_y + \sum_{i=1}^{m} s_{x_i}}{m+1} \right)^{m+1} = \left(\frac{T_e}{m+1} \right)^{m+1} \le (8W)^{m+1}.$$

Thus a good constraint is satisfied by A_α with probability at least $(8W)^{-(m+1)}$. Averaging over the random constraint and the random labeling, we obtain

$$\mathbb{E}_{A_\alpha} \left[\Pr_{e \sim \mu} [A_\alpha \text{ satisfies } e] \right] > \frac{5}{8} (8W)^{-(m+1)}.$$

Therefore some deterministic labeling satisfies more than $\frac{5}{8}(8W)^{-(m+1)}$ fraction of constraints. Since $\mathsf{val}(\Psi) \le \zeta$, we have

$$\frac{5}{8}(8W)^{-(m+1)} < \zeta.$$

Since $W = \Lambda w(\alpha) = w(\alpha)/s$, we have

$$w(\alpha) = W \cdot s > \frac{1}{8} \left(\frac{5}{8} \right)^{1/(m+1)} \zeta^{-1/(m+1)} \cdot s,$$

which is exactly (6). □

We next record the parameter substitution that will be used to prove Theorem 4.2.

Proposition 4.7. *Let K and $m \ge 2$ be integers such that*

$$m + 1 \le \log K, \qquad R := \left\lfloor \frac{K}{m+1} \right\rfloor \ge R_\star \left(m, \frac{1}{m^2}, \frac{1}{100(m+1)} \right). \tag{7}$$

Then there is a polynomial-time deterministic reduction to DMMSA instances whose formulas have leaf size at most K and whose size parameter is s, with the following gap:

YES: $\mathsf{val}_{1-1/(100(m+1))}(\mathcal{D}_0) \le s$.

NO: *Every assignment α of weight*

$$w(\alpha) \le \frac{K^{1-1/m}}{32(m+1)} \cdot s$$

satisfies

$$\Pr_{F \sim \mathcal{D}_0} [F(\alpha) = 1] \le \frac{3}{4}.$$

Proof. Apply Theorem 4.5 with

$$\xi := \frac{1}{m^2}, \qquad \delta := \frac{1}{100(m+1)},$$

and alphabet bound R from (7). In the YES case, Lemma 4.6(b) gives $\mathsf{val}_{1-\delta}(\mathcal{D}_0) \le s$.

In the NO case, put

$$\zeta := R^{-(1-1/m^2)m}.$$

We write the weight threshold supplied by Lemma 4.6(c) as $B \cdot s$, where B is the multiplicative factor

$$\begin{aligned} B &:= \frac{1}{8}\left(\frac{5}{8}\right)^{1/(m+1)} \zeta^{-1/(m+1)} \\ &= \frac{1}{8}\left(\frac{5}{8}\right)^{1/(m+1)} R^{(1-1/m^2)m/(m+1)} \\ &= \frac{1}{8}\left(\frac{5}{8}\right)^{1/(m+1)} R^{1-1/m}. \end{aligned}$$

Thus Lemma 4.6(c) says that every assignment α with

$$w(\alpha) \le B \cdot s$$

has satisfaction probability at most $3/4$. It remains to lower bound this multiplicative factor B in terms of the leaf-size parameter K.

Since $(5/8)^{1/(m+1)} \ge 5/8$, we have

$$B \ge \frac{5}{64} R^{1-1/m}. \tag{8}$$

Moreover, from $m+1 \le \log K$ and $m \ge 2$, for all sufficiently large K,

$$R = \left\lfloor \frac{K}{m+1} \right\rfloor \ge \frac{K}{2(m+1)}.$$

Combining this with (8),

$$B \ge \frac{5}{64}\left(\frac{K}{2(m+1)}\right)^{1-1/m} \ge \frac{5}{64} \cdot \frac{K^{1-1/m}}{2(m+1)} \ge \frac{K^{1-1/m}}{32(m+1)}.$$

Thus every assignment of weight at most $K^{1-1/m}s/(32(m+1))$ has satisfaction probability at most $3/4$ under $\mathcal{D}_0$. □

The reduction above gives a constant soundness threshold $3/4$. To match the formulation of Theorem 4.2, whose NO-side threshold is $o(1)$, we take an AND product of independent local formulas. For a distribution $\mathcal{D}_0$ over monotone formulas and an integer $q \ge 1$, let $\mathcal{D}_0^{\wedge q}$ be the distribution that samples $F_1, \dots, F_q \sim \mathcal{D}_0$ independently and outputs

$$F_1 \wedge F_2 \wedge \cdots \wedge F_q.$$

Every output formula is monotone, and its leaf size is at most q times the leaf size of $\mathcal{D}_0$. For every fixed assignment α,

$$\Pr_{F \sim \mathcal{D}_0^{\wedge q}}[F(\alpha) = 1] = \left(\Pr_{F \sim \mathcal{D}_0}[F(\alpha) = 1]\right)^q. \tag{9}$$

Consequently, if $\mathsf{val}_{1-\tau}(\mathcal{D}_0) \le s$, then

$$\mathsf{val}_{1-q\tau}(\mathcal{D}_0^{\wedge q}) \le s,$$

because $(1-\tau)^q \ge 1 - q\tau$. Also, if every assignment of weight at most βs has satisfaction probability at most $3/4$ under $\mathcal{D}_0$, then every such assignment has satisfaction probability at most $(3/4)^q$ under $\mathcal{D}_0^{\wedge q}$.

We are now ready to prove Theorem 4.2.

Proof of Theorem 4.2. For an integer K, call an integer $m \geq 2$ *K-admissible* if

$$m+1 \leq \log K, \qquad \left\lfloor \frac{K}{m+1} \right\rfloor \geq R_\star\left(m, \frac{1}{m^2}, \frac{1}{100(m+1)}\right).$$

For every fixed $M \geq 2$, the threshold $R_\star(M, 1/M^2, 1/(100(M+1)))$ is finite, while $\log K \to \infty$ and $\lfloor K/(M+1) \rfloor \to \infty$ as $K \to \infty$. Thus this fixed integer M is K-admissible for all sufficiently large K. In particular, the set of K-admissible integers is nonempty for all sufficiently large K. For each such K, this set is finite, since every K-admissible integer m satisfies $2 \leq m \leq \log K - 1$. Hence it has a largest element; denote it by m_K.

The same observation implies that $m_K \to \infty$: for every fixed $M \geq 2$, we have $m_K \geq M$ for all sufficiently large K. Also, by admissibility, $m_K + 1 \leq \log K$. Therefore

$$m_K \to \infty, \qquad m_K + 1 \leq \log K. \tag{10}$$

Now fix a sufficiently large final leaf-size bound ℓ and set

$$K = K(\ell) := \left\lfloor \frac{\ell}{\sqrt{\log \ell}} \right\rfloor, \qquad m := m_K, \qquad q := \lfloor \sqrt{m} \rfloor.$$

By (10), we have $q \to \infty$. Also $q \leq \sqrt{m} \leq \sqrt{\log K} \leq \sqrt{\log \ell}$, and therefore

$$qK \leq \ell.$$

Apply Proposition 4.7 with the base leaf bound K and the admissible arity parameter $m = m_K$, and then replace $\mathcal{D}_0$ by the AND-product distribution $\mathcal{D} := \mathcal{D}_0^{\wedge q}$. The resulting formulas have leaf size at most $qK \leq \ell$.

Define

$$\varepsilon(\ell) := \frac{q}{100(m+1)}, \qquad \gamma(\ell) := 2\left(\frac{3}{4}\right)^q, \qquad \sigma(\ell) := \frac{K^{1-1/m}}{32(m+1)}. \tag{11}$$

For large enough ℓ, $\gamma(\ell) \leq 1$. In the YES case, Proposition 4.7 and the amplification observation give

$$\mathsf{val}_{1-\varepsilon(\ell)}(\mathcal{D}) \leq s.$$

In the NO case, every assignment of weight at most $\sigma(\ell)s$ has satisfaction probability at most $(3/4)^q$ under $\mathcal{D}$, which is strictly smaller than $\gamma(\ell)$. Hence

$$\mathsf{val}_{\gamma(\ell)}(\mathcal{D}) > \sigma(\ell)s.$$

This is exactly the gap required in $\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{\sigma(\ell)}\mathsf{F}[\ell]$-DMMSA.

It remains only to check the asymptotics. Since $m = m_K \to \infty$ and $q = \lfloor \sqrt{m} \rfloor$, we have

$$\varepsilon(\ell) = \frac{q}{100(m+1)} = o(1), \qquad \gamma(\ell) = 2\left(\frac{3}{4}\right)^q = o(1).$$

Furthermore,

$$\frac{\log \sigma(\ell)}{\log \ell} = \left(1 - \frac{1}{m}\right)\frac{\log K}{\log \ell} - \frac{\log(32(m+1))}{\log \ell}.$$

Here $K = \lfloor \ell/\sqrt{\log \ell} \rfloor$, so $\log K / \log \ell \to 1$. Also $m + 1 \leq \log K$, and hence $\log(32(m+1))/\log \ell = o(1)$. Therefore

$$\frac{\log \sigma(\ell)}{\log \ell} \to 1,$$

that is, $\sigma(\ell) = \ell^{1-o(1)}$. This completes the proof of Theorem 4.2. □

4.3 The Star-Projection Structure of Minzer–Zheng PCP

We observe that the Minzer–Zheng theorem stated below has a star-projection structure, which yields Theorem 4.5.

Theorem 4.8 ([MZ26, Theorem 1.4, original form])**.** *For every integer $q \geq 2$ and every $\xi, \delta > 0$, there is an integer $R_{\mathrm{MZ}}(q, \xi, \delta)$ such that for every $R \geq R_{\mathrm{MZ}}(q, \xi, \delta)$, it is* NP*-hard to distinguish, given a q-CSP instance Φ with alphabet size at most R, between the following two cases:*

$$\mathsf{val}(\Phi) \geq 1 - \delta \qquad \text{and} \qquad \mathsf{val}(\Phi) \leq R^{-(1-\xi)(q-1)}.$$

We use the following structural fact from the proof of Theorem 4.8. For $q = k + 1$, the final composed PCP that Minzer–Zheng denote by Ψ_{k+1} is the CSP constructed in their Section 3.3, and their Section 5.2 proves Theorem 1.4 by analyzing this final PCP. Therefore, to derive the star-projection version used in Theorem 4.5, it remains to verify that the local constraints of Ψ_{k+1} have the star-projection form.

Lemma 4.9. *Fix $m \geq 2$ and put $q = m + 1$. The final composed PCP of Minzer–Zheng for arity q has a value-preserving representation as a star-projection $(m+1)$-CSP: its variables can be partitioned into variables $\mathcal{X}$ and center variables $\mathcal{Y}$, and for every constraint*

$$e = (y; x_1, \ldots, x_m) \qquad (y \in \mathcal{Y},\ x_i \in \mathcal{X}),$$

there are maps $\pi_{e,i} \colon \Sigma_{x_i} \to \Sigma_y$ such that the original Minzer–Zheng predicate accepts exactly when

$$\pi_{e,i}(a_i) = b \qquad \text{for every } i \in [m],$$

where a_i is the answer to x_i and b is the answer to y.

Proof. We unpack the CSP described in [MZ26, Section 3.3]. Their final PCP has two types of vertices, denoted by $\mathcal{A}$ and $\mathcal{B}$. The vertices on the $\mathcal{A}$ side have the form

$$L \oplus H_U.$$

Here U is a question in the outer PCP, H_U is the constraint subspace associated with U, and L is chosen so that $L \cap H_U = \{0\}$; thus $\oplus$ denotes a direct sum of subspaces. The alphabet at such a vertex consists of the linear functions on $L \oplus H_U$ (satisfying the side conditions associated with U).

The vertices on the $\mathcal{B}$ side are center subspaces, denoted below by R. The alphabet at such a vertex consists of the linear functions $R \to \mathbb{F}_2$.

We set

$$\mathcal{X} := \mathcal{A}, \qquad \mathcal{Y} := \mathcal{B}.$$

Now fix one sampled constraint of their final $(m+1)$-CSP. In the notation of Minzer–Zheng, the sampler first chooses U and a center subspace R. Then, for each $i \in [m]$, it chooses a subspace L_i with $R \subseteq L_i$, and finally chooses an $\mathcal{A}$-vertex

$$x_i := L_i' \oplus H_{U_i'} \in [L_i \oplus H_U].$$

Here, $[L_i \oplus H_U]$ denotes the equivalence class, called a clique by Minzer–Zheng, of the $\mathcal{A}$-vertex $L_i \oplus H_U$ under the equivalence relation from [MZ26, Section 3.3.3]. The center variable of the constraint is

$$y := R \in \mathcal{Y}.$$

Thus the constraint queries one center variable y and m variables $x_1, \dots, x_m$.

It remains to define the projection maps. Fix the sampled constraint e and an index i. Given an $\mathcal{A}$-label $a \in \Sigma_{x_i}$, regard a as a linear function on $L'_i \oplus H_{U'_i}$ (satisfying the side conditions associated with U'_i). Since $x_i = L'_i \oplus H_{U'_i}$ lies in the clique of $L_i \oplus H_U$, [MZ26, Lemma 3.4] gives a unique linear extension on the common span

$$S_i := L_i + H_U + H_{U'_i} = L'_i + H_U + H_{U'_i}.$$

Let this extension be

$$g_{e,i,a} \colon S_i \to \mathbb{F}_2.$$

It extends the label a and is compatible with the side conditions associated with both U'_i and U. Since $R \subseteq L_i \subseteq S_i$, the restriction $g_{e,i,a}|_R$ is a linear function on R, hence a legal center label. Define

$$\pi_{e,i}(a) := g_{e,i,a}|_R \in \Sigma_y.$$

The uniqueness assertion in Lemma 3.4 is exactly what makes $\pi_{e,i}$ a well-defined function of the left label a.

The constraint in the Minzer–Zheng construction is as follows. In the notation above, if the $\mathcal{A}$-answers are $a_1, \dots, a_m$ and the center answer is $b \in \Sigma_y$, then the constraint is precisely

$$\pi_{e,i}(a_i) = b \qquad \text{for every } i \in [m].$$

Thus every local predicate is a star-projection predicate. □

Proof of Theorem 4.5. Fix $m \geq 2$ and $\xi, \delta > 0$, and set $q := m + 1$. The proof of Theorem 4.8 constructs and analyzes the final composed PCP Ψ_q described in [MZ26, Section 3.3]. By Lemma 4.9, this CSP has a star-projection representation. Applying Theorem 4.8 with arity $q = m+1$ therefore gives the same completeness and soundness after this rewriting:

$$\mathsf{val}(\Psi) \geq 1 - \delta \qquad \text{or} \qquad \mathsf{val}(\Psi) \leq R^{-(1-\xi)(q-1)} = R^{-(1-\xi)m}.$$

Finally, define $R_\star(m, \xi, \delta)$ to be a sufficiently large threshold for this application of Minzer–Zheng's theorem, enlarging it if necessary to absorb the harmless convention that the two sides may have different alphabet sizes but both are bounded by the same parameter R. This proves Theorem 4.5. □

5 From DMMSA to Learning under Distributions with Small Advice

In this section, we present the reduction from $\mathfrak{C}$-DMMSA to learning under distributions, where the advice complexity of sampling is upper bounded by the size of secret sharing in $\mathfrak{C}$. It yields Theorem 1.3 along with Theorem 4.2 and Theorem 3.5.

Recall that $\mathfrak{S}_\ell$ is the class of monotone formulas such that the induced authorized set admits secret sharing of total length ℓ for each shared bit.

The basic ideas are the same as those presented in [Hir22]. For completeness, we give the proof below.

Lemma 5.1. *There exists a universal constant $c_U \geq 0$ such that the following holds. Let $\ell \in \mathbb{N}$, $\sigma \geq 1$, $\varepsilon \in [0,1]$, and $\gamma \in (0,1]$. The problem $\mathrm{Gap}^{\varepsilon,\gamma}_\sigma \mathfrak{S}_\ell$-DMMSA is reducible to $\mathrm{Gap}^{\varepsilon,5\gamma}_{0.49\sigma}\mathrm{Learn}[\ell + 2\lceil \log(\ell+1)\rceil + c_U]$ as follows: There exists a randomized polynomial-time algorithm R such that, for given $\mathrm{Gap}^{\varepsilon,\gamma}_\sigma \mathfrak{S}_\ell$-DMMSA-instance $z = (\mathcal{D}, w, 1^n, s)$ and randomness $z' \sim \{0,1\}^\lambda$, where $\lambda := \lambda(|z|)$, R produces a distribution $\mathcal{E}_{z,z'}$ over $\{0,1\}^{n'} \times \{0,1\}$ and a size parameter $s' := \lceil 2s\lambda \rceil$ such that*

- $\{\mathcal{E}_{z,z'}\}_{z,z'}$ *is samplable with advice complexity at most* $\ell + 2\lceil \log(\ell+1)\rceil + c_U$.
- (*Completeness*) *If* z *is an yes instance, then for every* $z' \in \{0,1\}^\lambda$*, there exists a linear-time program* h *of size at most* s' *such that*

$$\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}[h(x) = b] \geq 1 - \varepsilon.$$

- (*Soundness*) *If* z *is a no instance, then with probability* $1 - \mathsf{negl}(|z|)$ *over the choice of* $z' \sim \{0,1\}^\lambda$*, there is no program* h *of size* $0.49\sigma \cdot s'$ *such that*

$$\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}[h(x) = b] \geq \frac{1}{2} + \frac{5\gamma}{2}.$$

Proof. Let $z = (\mathcal{D}, w, 1^n, s)$ be an instance of $\mathrm{Gap}^{\varepsilon,\gamma}_\sigma\mathfrak{S}_\ell$-DMMSA. Here, $\mathcal{D}$ is a circuit representing a sampler for a distribution on n-variate monotone formulas whose authorized sets admit secret sharing of total length ℓ, and $w\colon [n] \to [0,1]$ is a weight function (i.e., $\sum_{i\in[n]} w(i) = 1$), where we identify the variable set with the index set $[n]$. Recall that we assumed that $w(i) \geq 1/\mathsf{poly}(|z|)$ for each $i \in [n]$ and $s \geq 1/\mathsf{poly}(|z|)$.

We construct the reduction R that maps z and randomness $z' \sim \{0,1\}^\lambda$ to an instance of the learning problem as in the theorem. Let $\rho \in \mathbb{N}$ be the amount of random bits $\mathcal{D}$ takes as input.

Let $(\mathsf{Share}, \mathsf{Rec})$ be the secret sharing scheme for the class $\mathfrak{S}_\ell$. Let $\lambda := \lambda(|z|) \in \mathbb{N}$ be a large enough and polynomially bounded parameter.

The reduction R first selects n random strings $f_i \sim \{0,1\}^{w(i)\cdot\lambda}$ for each $i \in [n]$ (notice that it requires $\sum_i w(i)\lambda = \lambda$ random bits) and then produces a sampler of the following distribution $\mathcal{E} := \mathcal{E}_{z,z'}$ over $\{0,1\}^{n'} \times \{0,1\}$, where $n' := \ell + \rho + \ell\lambda$.

The distribution $\mathcal{E}$. Select $r \sim \{0,1\}^\rho$ and obtain a monotone formula $\varphi := \mathcal{D}(r)$. Then, select $b \sim \{0,1\}$ and execute $\mathsf{Share}(b;\varphi)$ to obtain shares $s \in \{0,1\}^\ell$. For each position $i \in [\ell]$, let $j_i \in [n]$ be the index of the variable to which s_i is distributed. For each $j \in [n]$ and $q \in [\ell]$, let $z_j^{(q)} \sim \{0,1\}^{w(j)\lambda}$ independently, and write $z_j := z_j^{(1)} \circ \cdots \circ z_j^{(\ell)}$. Define

$$c_i := s_i \oplus \langle f_{j_i}, z_{j_i}^{(i)} \rangle_{\mathbb{F}_2}$$

for each $i \in [\ell]$. The distribution $\mathcal{E}$ is defined as the distribution of $(r \circ z_1 \circ \cdots \circ z_n \circ c_1 \circ \cdots \circ c_\ell, b) \in \{0,1\}^{n'} \times \{0,1\}$ over $r, b, \{z_j^{(q)}\}$, and the randomness for Share (note that each f_i is embedded into the description of $\mathcal{E}$). It it easy to verify that $\mathcal{E}$ has a polynomial-time sampler of description length at most

$$|z| + |\mathsf{Share}| + \sum_i |f_i| + O(\log|z|) \leq |z| + \lambda + O(\log|z|).$$

Since $\lambda \leq \mathsf{poly}(|z|)$, the above is polynomially bounded in $|z|$, and R halts in polynomial time.

First, we observe that $\{\mathcal{E}_{z,z'}\}_{z,z'}$ is samplable with advice complexity at most $\ell + 2\lceil\log(\ell+1)\rceil + c_U$ in the sense of Definition 3.11. Let $N := n' + 1$, identifying a sample $(x,b) \in \{0,1\}^{n'} \times \{0,1\}$ with $x \circ b \in \{0,1\}^N$. Let S_R be the sampler that, given 1^N, an advice string $a \in \{0,1\}^\ell$, and a random string r', outputs $(r'_{[N-\ell-1]} \circ a, r'_{N-\ell})$. By universality of U, there exists a constant c_U such that each $a \in \{0,1\}^\ell$ has an encoding $\widehat{a} \in \{0,1\}^{\leq \ell + 2\lceil\log(\ell+1)\rceil + c_U}$ with

$$U(\widehat{a}, r', N) = S_R(1^N, a, r')$$

for all r' and N. We define the advice function $\alpha_{z,z'} : \{0,1\}^{2^{2N}} \to \{0,1\}^{\leq \ell + 2\lceil\log(\ell+1)\rceil + c_U}$ on seeds $r' \in \{0,1\}^{2^{2N}}$ as follows. It computes the string $c_1 \circ \cdots \circ c_\ell$ according to the definition of $\mathcal{E}_{z,z'}$

and outputs the fixed universal-machine encoding $c_1 \widehat{\circ \cdots \circ} c_\ell$ introduced above. Namely, using the prefix of r', parse r, the strings $z_j^{(q)} \in \{0,1\}^{w(j)\lambda}$ for every $j \in [n]$ and $q \in [\ell]$, and the bit b exactly as in the sample distribution above. Let $\varphi := \mathcal{D}(r)$ and $s := \mathsf{Share}(b;\varphi)$, where Share uses fresh randomness from the remaining bits of r'. For the same indices j_i as in $\mathcal{E}$, set $c_i = s_i \oplus \langle f_{j_i}, z_{j_i}^{(i)} \rangle_{\mathbb{F}_2}$ for each $i \in [\ell]$. Then, for every z and $z' = \{f_i\}_i$, the distribution $\mathcal{E}_{z,z'}$ is statistically identical to the distribution generated by $U(\alpha_{z,z'}(r'), r', N)$ over $r' \sim \{0,1\}^{2^{2N}}$. Moreover, $U(\alpha_{z,z'}(r'), r', N)$ never outputs $\bot$.

Below, we prove the completeness and soundness. This completes the proof by selecting λ to be at least $\max\{p(|z|), p'(|z|)\}$, where p and p' are the polynomials specified below, and by setting $s' := \lceil 2s\lambda \rceil$. Since the size parameter s is inverse-polynomially lower bounded, we may increase the polynomial $\lambda(\cdot)$ so that $s\lambda \geq 100$ for every valid input. Then $s' \leq 2s\lambda + 1 \leq 2.01 s\lambda$, and hence $0.49\sigma s' \leq 0.99\sigma s\lambda$; together with Claim 5.3, this gives the desired soundness for $\mathrm{Gap}^{\varepsilon,5\gamma}_{0.49\sigma}\mathrm{Learn}[\ell + 2\lceil \log(\ell+1) \rceil + c_U]$.

Claim 5.2 (completeness)**.** *There exists a polynomial p such that for every yes instance z, every $\lambda \geq p(|z|)$, and every choice of $z' = \{f_i\}_{i\in[n]}$, there exists a linear-time program h of size at most $2s\lambda$ (and hence at most s') such that*

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] \geq 1 - \varepsilon.$$

Claim 5.3 (soundness)**.** *There exists a polynomial p' such that for every no instance z, every $\lambda \geq p'(|z|)$, the following holds with probability at least $1 - \mathsf{negl}(|z|)$ over the choice of $z' = \{f_i\}$: there is no program h of size at most $0.99\sigma \cdot s \cdot \lambda$ such that*

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] > \frac{1}{2} + 2\gamma.$$

Proof of Claim 5.2. We fix an yes instance z arbitrarily. Let $\lambda \in \mathbb{N}$ be a large enough parameter. We also fix $z' = \{f_i\}_{i\in[n]}$ arbitrarily and let $\mathcal{E} := \mathcal{E}_{z,z'}$.

Since z is an yes instance, there exists an assignment $\alpha \in \{0,1\}^n$ to $\mathcal{D}$ such that $w(\alpha) \leq s$ and

$$\Pr_{\varphi\sim\mathcal{D}}[\varphi(\alpha) = 1] \geq 1 - \varepsilon.$$

We consider the following hypothesis $h\colon \{0,1\}^{n'} \to \{0,1\}$ into which α, z, and f_i are embedded for all $i \in [n]$ with $\alpha_i = 1$. On input $x = r \circ z_1 \circ \cdots \circ z_n \circ c_1 \cdots \circ c_\ell$, where $r \in \{0,1\}^\rho$, $z_j = z_j^{(1)} \circ \cdots \circ z_j^{(\ell)}$ with $z_j^{(q)} \in \{0,1\}^{w(j)\lambda}$ for each $j \in [n]$ and $q \in [\ell]$, and $c_i \in \{0,1\}$ for each $i \in [\ell]$, the hypothesis h obtains $\varphi = \mathcal{D}(r)$. Then, for each $i \in [\ell]$, if $\alpha_{j_i} = 1$ (where $j_i \in [n]$ is the same as above) h computes s_i from c_i and $\langle f_{j_i}, z_{j_i}^{(i)} \rangle_{\mathbb{F}_2}$ and executes $\mathsf{Rec}(;\varphi)$ to reconstruct b from shares $\{s_i : i \in [\ell] \text{ s.t. } \alpha_{j_i} = 1\}$.

Let $z = z_1 \circ \cdots \circ z_n$ and $c = c_1 \cdots c_\ell$. For convenience, we identify $r \circ z \circ c$ with (r, z, c).

By the completeness of $(\mathsf{Share}, \mathsf{Rec})$, $h(r, z, c)$ can correctly reconstruct b as long as $\varphi(\alpha) = 1$ where $\varphi = \mathcal{D}(r)$. Thus, we have

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] \geq \Pr_{\varphi\sim\mathcal{D}}[\varphi(\alpha) = 1] \geq 1 - \varepsilon.$$

It is easy to verify that h halts in polynomial time in n'. The description size is bounded above

by

$$|h| \le \left(\sum_{i:\alpha_i=1} |f_i|\right) + n + |z| + O(\log(n|z|\lambda)) \le \left(\sum_{i:\alpha_i=1} w(i)\lambda\right) + \mathsf{poly}(|z|) + O(\log(|z|\lambda))$$
$$= w(\alpha)\lambda + \mathsf{poly}(|z|) + O(\log(|z|\lambda))$$
$$\le s\lambda + \mathsf{poly}(|z|) + O(\log(|z|\lambda)).$$

Since the size parameter s is inverse-polynomially lower bounded in $|z|$, the polynomial p can be chosen large enough so that every $\lambda \ge p(|z|)$ satisfies

$$\mathsf{poly}(|z|) + O(\log(|z|\lambda)) \le s\lambda.$$

Hence $|h| \le 2s\lambda \le s'$. The running time of h is $O(n') + \mathsf{poly}(|z|)$; by increasing p if necessary, this is linear in the padded input length n'. ◇

Proof of Claim 5.3. We fix a no instance z arbitrarily. Let $\lambda \in \mathbb{N}$ be a large enough parameter. We consider the random choice of $z' = \{f_i\}_{i\in[n]}$, and let $\mathcal{E} := \mathcal{E}_{z,z'}$.

Let h be an arbitrary program of description size at most $0.99\sigma \cdot s \cdot \lambda$ that maps n' bits to 1 bit. We will show that

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] \le \frac{1}{2} + 2\gamma.$$

For each $i \in [n]$, let $\tilde{f}_i \in \{0,1\}^\lambda$ be the string obtained from f_i with padding, i.e., $\tilde{f}_i = f_i \circ 0^{\lambda - w(i)\lambda}$.

We construct a distinguisher D that takes an advice string $\alpha = (b, r, r')$ and input $(Z_1Y_1, \dots, Z_nY_n)$, where $Z_i = (Z_i^{(1)}, \dots, Z_i^{(\ell)})$ takes values in $(\{0,1\}^\lambda)^\ell$ and $Y_i = (Y_i^{(1)}, \dots, Y_i^{(\ell)})$ takes values in $\{0,1\}^\ell$. The distinguisher computes (i) $\varphi = \mathcal{D}(r)$, (ii) $(s_1, \dots, s_\ell) = \mathsf{Share}(b; \varphi)$ with randomness r', (iii) $c_q = Y_{j_q}^{(q)} \oplus s_q$ for each $q \in [\ell]$ (recall that j_q is the index of the q-th share). For each $i \in [n]$, let

$$\overline{Z}_i := (Z_i^{(1)})_{[w(i)\lambda]} \circ \cdots \circ (Z_i^{(\ell)})_{[w(i)\lambda]}.$$

The distinguisher outputs 1 if and only if

$$h(r \circ \overline{Z}_1 \circ \cdots \circ \overline{Z}_n \circ c_1 \circ \cdots \circ c_\ell) = b.$$

It is easy to verify that if $Z_iY_i = \mathsf{DP}_\ell(\tilde{f}_i; Z_i)$ for all i, and $\alpha = (b, r, r')$ is selected uniformly at random, then the input to h is distributed in the same manner as the example of $\mathcal{E}$.

Let $a := |\alpha| = \mathsf{poly}(|z|)$. Note that a is independent of λ. Applying the Algorithmic Information Extraction Lemma (Lemma 3.7) with $k = \ell$, there exists a subset $B \subseteq [n]$ such that

$$\mathrm{K}(\tilde{f}_B | D) \le |B| \cdot (n\ell + a + O(\log n\lambda a\gamma^{-1})) \le \mathsf{poly}(|z|) \cdot \log\lambda, \tag{12}$$

and

$$\left| \Pr_{\alpha,\{Z_iY_i\}} [D(\alpha, Z_1Y_1, \dots, Z_nY_n) = 1] - \Pr_{\alpha,\{Z_iY_i'\}} \left[D(\alpha, Z_1Y_1', \dots, Z_nY_n') = 1\right] \right| \le \gamma \tag{13}$$

where for each $i \in [n]$, $Z_i \sim (\{0,1\}^\lambda)^\ell$, $Z_iY_i = \mathsf{DP}_\ell(\tilde{f}_i; Z_i)$, $Y_i' \equiv Y_i$ if $i \in B$ and $Y_i' \sim \{0,1\}^\ell$ otherwise.

For every fixed nonempty set $T \subseteq [n]$, the string f_T is uniformly random of length $w(T)\lambda$. Hence the standard counting argument gives

$$\Pr\left[\mathrm{K}(f_T) < w(T)\lambda - 3|z|\right] \le 2^{-3|z|+O(1)}.$$

Since the unary string 1^n is part of the instance, $n \le |z|$, and a union bound over all $T \subseteq [n]$ shows that, with probability at least $1 - \mathsf{negl}(|z|)$ over the choice of $z' = \{f_i\}$,

$$\mathrm{K}(f_T) \ge w(T)\lambda - 3|z| \qquad \text{for every nonempty } T \subseteq [n]. \tag{14}$$

Below, we condition on this event.

If $B = \emptyset$, then $w(B) = 0 < \sigma s$. Otherwise, $w(B) \ge \min_i w(i) \ge 1/\mathsf{poly}(|z|)$. We have

$$\mathrm{K}(\tilde{f}_B|D) + O(\log\lambda) \ge \mathrm{K}(f_B|D) + O(\log\lambda) \ge \mathrm{K}(f_B|h) + O(\log\lambda) \ge \mathrm{K}(f_B) - |h|,$$

where the second inequality holds because D is constructed from h. Combining this with Equation (14) and Eq. (12), we obtain

$$w(B)\lambda - 3|z| - |h| \le \mathrm{K}(\tilde{f}_B|D) + O(\log\lambda) \le p_0(|z|)\cdot\log\lambda,$$

for some universal polynomial p_0. Since $w(B) \ge \min_i w(i) \ge 1/\mathsf{poly}(|z|)$, there exists a universal polynomial p' such that for every $\lambda \ge p'(|z|)$,

$$3|z| + p_0(|z|)\log\lambda \le 0.005 w(B)\lambda.$$

Thus,

$$0.995 w(B)\lambda \le w(B)\lambda - (3|z| + p_0(|z|)\log\lambda) \le |h|.$$

Since $|h| \le 0.99\sigma\cdot s\cdot\lambda$, we have

$$w(B) \le \frac{0.99}{0.995}\sigma s < \sigma s.$$

Therefore, in both cases $B = \emptyset$ and $B \ne \emptyset$, we have $w(B) < \sigma s$.

The characteristic string $\chi_B \in \{0,1\}^n$ (i.e., $(\chi_B)_i = 1$ iff $i \in B$) satisfies that $w(\chi_B) = w(B) < \sigma s$. Since z is a no instance,

$$\Pr_{\varphi\sim\mathcal{D}}[\varphi(\chi_B) = 0] \ge 1 - \gamma.$$

Now, consider the execution of $D(\alpha, Z_1Y_1', \ldots, Z_nY_n')$. Condition on a choice of r for which $\varphi(\chi_B) = 0$. Then B is an unauthorized set for the access structure induced by φ. The input to h contains the shares of parties in B only through the bits $c_q = s_q \oplus (Y'_{j_q})^{(q)}$ with $j_q \in B$ and contains no information about the shares of parties outside B: for every share position q with $j_q \notin B$, the mask bit $(Y'_{j_q})^{(q)}$ is independent and uniform, so c_q is independent of s_q. Therefore, after adjoining the public randomness r, all Z_i's, and the independent mask bits, the entire view given to the fixed predictor h has the same distribution when the shared bit is 0 as when it is 1, by the perfect privacy of the secret sharing scheme. Hence, under this conditioning, D outputs 1 (i.e., h correctly outputs b) with probability exactly $1/2$. On the remaining γ-fraction of choices of r we use the trivial upper bound 1. Thus,

$$\Pr_{\alpha,\{Z_iY_i'\}}\left[D(\alpha, Z_1Y_1', \ldots, Z_nY_n') = 1\right] \le \frac{1}{2} + \gamma.$$

Therefore, by Eq. (13)

$$\begin{aligned}\Pr_{\alpha,\{Z_iY_i\}}[D(\alpha, Z_1Y_1, \ldots, Z_nY_n) = 1] &\le \Pr_{\alpha,\{Z_iY_i'\}}\left[D(\alpha, Z_1Y_1', \ldots, Z_nY_n') = 1\right] + \gamma \\ &\le \frac{1}{2} + 2\gamma.\end{aligned}$$

By contrast,

$$\Pr_{\alpha,\{Z_iY_i\}}[D(\alpha, Z_1Y_1, \ldots, Z_nY_n) = 1] \ge \Pr_{(x,b)\sim\mathcal{E}}[h(x) = b].$$

From the two inequalities above, we conclude that

$$\Pr_{(x,b)\sim\mathcal{E}}[h(x) = b] \leq \frac{1}{2} + 2\gamma.$$

◇

□

The lemma above implies Theorem 1.3 along with Theorem 4.2 and Theorem 3.5.

Proof of Theorem 1.3. Let $\sigma_0(L) = L^{1-o(1)}$, $\varepsilon_0(L) = o(1)$, and $\gamma_0(L) = o(1)$ be the functions from Theorem 4.2. Let c_U be the constant in Lemma 5.1. For every sufficiently large $\ell \in \mathbb{N}$, let L be the largest integer satisfying $L + 2\lceil\log(L+1)\rceil + c_U \leq \ell$. Then $L = \ell - O(\log \ell)$. By Theorem 3.5, we have $\mathsf{F}[L] \subseteq \mathfrak{S}_L$. Therefore, Theorem 4.2 and Lemma 5.1 give a randomized many-one reduction from an NP-hard problem to

$$\mathrm{Gap}^{\varepsilon_0(L),5\gamma_0(L)}_{0.49\sigma_0(L)}\mathrm{Learn}[\ell].$$

Since $L = \ell - O(\log \ell)$, renaming $0.49\sigma_0(L)$ and $5\gamma_0(L)$ as $\sigma(\ell)$ and $\gamma(\ell)$ proves Theorem 1.3. □

Note that the Theorem 1.3 states the result in a setting where only access to samples is provided (as defined in Definition 1.2). The proof above also works in the stronger setting where the learner is given the description of the underlying distribution for learning.

6 Reduction to Inverting Auxiliary-Input Functions and Consequences

In this section, we present the reduction from GapLearn to inverting AIOWF, which establishes the following coAM upper bound for DMMSA.

Theorem 6.1. *For every constant $\gamma \in [0, 1/5)$, there exists $C > 0$ such that for every $\varepsilon(\ell) = o(1)$ and every sufficiently large $\ell \in \mathbb{N}$, $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{C\cdot\ell}\mathfrak{S}_\ell$-DMMSA is in coAM.*

Note that Theorem 6.1 implies Theorem 1.9 as special cases since (i) $\mathsf{F}[\ell] \subseteq \mathfrak{S}_\ell$ (Theorem 3.5) and (ii) $\mathsf{Junta}[\ell] \subseteq \mathfrak{S}_{1.5^{\ell+o(\ell)}}$ [AN21].

Theorem 6.1 is proved across Sections 6.1 to 6.3, outlined as follows: In Section 6.1, we present a reduction from GapLearn to extrapolating a universal distribution based on the theory of inductive inference [Sol64a; Sol64b]. We further extend the reduction from DMMSA to inverting an auxiliary-input function, using a restricted form of description-restricted, context-sensitive, fixed-auxiliary-input nonadaptive reduction, detailed below. In Section 6.3, building upon [AGGM06], we observe that any promise problem Π reducible to inverting via this restricted reduction is contained in coAM, completing the proof of Theorem 6.1. In Section 6.5, we prove the existence of a one-way function under the average-case hardness of GapDMMSA and GapLearn (Theorems 1.5 and 1.10).

First, we present the restricted form of reductions to inverting auxiliary-input functions. For each *total* Turing machine M, we define an M-based (randomized) oracle $\mathcal{O}_M \colon \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ as $\mathcal{O}_M(x; \rho_{\mathsf{shared}}, \rho_{\mathsf{ind}}) = y$ if $M(x, \rho_{\mathsf{shared}}, \rho_{\mathsf{ind}})$ outputs y and halts, where ρ_{shared} is a shared randomness selected at the initialization (and used throughout), and ρ_{ind} is a random seed selected at each query access. When the reduction R given access to $\mathcal{O}_M$ is time-bounded, the total length of the shared randomness and the independent random seeds are bounded and can be selected at the initialization as a single random string ρ. In this case, we represent the M-based oracle that uses ρ as a random tape (for both ρ_{shared} and ρ_{ind}) by $\mathcal{O}_M(\text{-}; \rho)$

Definition 6.2 (Description-restricted context-sensitive reduction)**.** *Let $f = \{f_z\}_{z\in\{0,1\}^*}$ be an auxiliary-input function. A randomized oracle machine $R^?$ is said to be a description-restricted context-sensitive fixed-auxiliary-input nonadaptive (FAIN) reduction from a promise problem Π to inverting f if*

- *R is polynomial-time and nonadaptive;*
- *For every $x \in \{0,1\}^*$, the reduction $R(x)$ makes queries only of the form $(z_x, \text{-})$ for some auxiliary-input z_x determined by x;*
- *For every (possibly inefficient) total Turing machine M and for every long enough $x \in \{0,1\}^*$, if $\mathcal{O}_M(\text{-}, x; \rho)$ inverts f_{z_x} for every fixed random tape ρ used by the reduction, i.e., for every such ρ,*

$$\Pr_r\left[\mathcal{O}_M(z_x, f_{z_x}(r), x; \rho) \in f_{z_x}^{-1}(f_{z_x}(r))\right] \geq 1/2,$$

then

$$\Pr_{R,\rho}[R^{\mathcal{O}_M(\text{-},x;\rho)}(x) = \Pi(x)] \geq 3/4.$$

We will show the following key lemma in Sections 6.1 and 6.2.

Lemma 6.3. *For every constant $\gamma \in [0, 1/5)$, there exists $C \geq 1$ such that for every $\varepsilon(\ell) = o(1)$ and every sufficiently large $\ell \in \mathbb{N}$, $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{C\cdot\ell}\mathfrak{S}_\ell$-DMMSA is reducible to inverting an auxiliary-input function $f = \{f_z\}$ via a description-restricted context-sensitive FAIN reduction.*

Combined with the following theorem, which is implicit in [AGGM06; ABX08], we derive Theorem 6.1.

Theorem 6.4. *If a promise problem Π is reducible to inverting an auxiliary-input function $f = \{f_z\}$ via a description-restricted context-sensitive FAIN reduction, then $\Pi \in \mathsf{coAM}$.*

We will observe Theorem 6.4 in Section 6.3.

6.1 Inductive Inference: From Learning to Universal Extrapolation

Towards proving Lemma 6.3, we first reduce learning to extrapolation under the time-bounded universal distribution.

Lemma 6.5. *Let $\mathcal{D} = \{\mathcal{D}_{z,z'}\}_{z\in Z, z'\in\{0,1\}^{r(z)}}$, where $Z \subseteq \{0,1\}^*$ and $r(z) \leq \mathsf{poly}(|z|)$, be a samplable distribution satisfying that there exist a function $n\colon Z \to \mathbb{N}$, a polynomially bounded function $s = s(z, z')$, a function $\eta\colon Z \to [0, 1]$, and polynomial p such that for every $z \in Z$ and $z' \in \{0,1\}^{r(z)}$,*

- *$\mathcal{D}_{z,z'}$ is a distribution over $\{0,1\}^{n(z)} \times \{0,1\}$, where $n(z) \leq \mathsf{poly}(|z|)$;*
- *there exists a $p(\cdot)$-time program h of description size $s(z, z')$ such that*

$$\Pr_{(x,b)\sim\mathcal{D}_{z,z'}}[h(x) = b] \geq 1 - \eta(z).$$

Then, for every $\varepsilon \in (0, 1/2)$, there exist constants $\eta_0 > 0$ and $c := c_\varepsilon > 0$ with the following property. For every polynomial-time computable polynomially bounded sample-size function $m = m(z, z')$,

there is a polynomial-time computable polynomially bounded time bound $t_m = t_m(z)$ such that, if $\eta(z) \le \eta_0$ for every long enough $z \in Z$, then for every long enough $z \in Z$,

$$\Pr_{z'}\left[m(z,z') < c\cdot(s(z,z')+\log|z|) \text{ or } \Pr_{(x^1,b^1),\dots,(x^m,b^m),i,\mathsf{Next}_1}\left[\mathsf{Next}_1\left(xb_{<i};\mathrm{Q}_{|z}^{t_m(z)}\right)=b_i\right]\ge 1-\varepsilon\right]\ge 1-o_{|z|}(1),$$

where, inside the probability over samples, m denotes $m(z,z')$, $i \sim [m]$, $(x^1,b^1),\dots,(x^m,b^m) \sim \mathcal{D}_{z,z'}$, and $xb_{<i} = x^1\circ\cdots\circ x^m\circ b^1\circ\cdots\circ b^{i-1}$.

Proof. Let $\varepsilon > 0$ and let $c := C_1 c_0^2 \varepsilon^{-2}$, where c_0 is a large enough universal constant that depends only on the universal Turing machine and the way of encoding and is specified later, and C_1 is a sufficiently large universal constant. Choose $\eta_0 > 0$ sufficiently small. Fix a polynomial-time computable polynomially bounded sample-size function $m = m(z,z')$. Let M_m be a polynomial upper bound such that $m(z,z') \le M_m(|z|)$ for all relevant z, z'. Let τ be a sufficiently large polynomial, in particular large enough for p and the time-complexity of sampling according to $\mathcal{D}$, and set

$$t_m(z) := M_m(|z|)\cdot\tau(|z|).$$

Fix a long enough $z \in Z$ arbitrarily. Let $n := n(z) \le \mathsf{poly}(|z|)$ and $\eta := \eta(z)$, and write $t := t_m(z)$.

We first take care of the randomness z'. For this fixed z and the sample-size function m, let $\overline{\mathcal{D}}_{z,m}$ be the mixed distribution obtained by first choosing $Z' \sim \{0,1\}^{r(z)}$ and then drawing $m(z,Z')$ independent samples from $\mathcal{D}_{z,Z'}$; its output is the concatenated string XB. This mixed distribution is samplable from the advice z in polynomial time. Enlarging τ if necessary, by applying Lemma 3.9 to $\overline{\mathcal{D}}_{z,m}$, we have

$$\Pr_{Z',(X,B)\sim\mathcal{D}_{z,Z'}^{\otimes m(z,Z')}}\left[\mathrm{cd}^t(XB|z) > 2\log t\right] \le \frac{1}{t}.$$

Put $\theta_z := 1/\log|z|$. By Markov's inequality, with probability at least $1-\theta_z$ over Z',

$$\Pr_{(X,B)\sim\mathcal{D}_{z,Z'}^{\otimes m(z,Z')}}\left[\mathrm{cd}^t(XB|z) > 2\log t\right] \le \frac{1}{t\theta_z}.$$

For every such z', since $\mathrm{cd}^t(XB|z) \le t$, we have

$$\mathbb{E}_{(X,B)\sim\mathcal{D}_{z,z'}^{\otimes m(z,z')}}[\mathrm{cd}^t(XB|z)] \le 2\log t + \frac{1}{\theta_z} \le C_{\mathrm{dep}}\log|z|, \tag{15}$$

for a universal constant C_{dep}, using that t is polynomially bounded in $|z|$. We call such z' depth-good. The probability that z' is not depth-good is $o_{|z|}(1)$.

Now fix an arbitrary depth-good $z' \in \{0,1\}^{r(z)}$ such that $m(z,z') \ge c\cdot(s(z,z')+\log|z|)$. Put $m := m(z,z')$ for readability, and let $s := s(z,z')$. Let $X_1,\dots,X_m,B_1,\dots,B_m$ be random variables representing the values of $x_1,\dots,x_m,b_1,\dots,b_m$, respectively, under $\mathcal{D}_{z,z'}^{\otimes m}$. Let $X = X_1\circ\cdots\circ X_m$, $B = B_1\circ\cdots\circ B_m$, and $XB_{<i} = X_1\circ\cdots\circ X_m\circ B_1\circ\cdots\circ B_{i-1}$ for each $i\in[m]$.

First, we evaluate the following quantity:

$$\begin{aligned}\mathrm{KL}\left(B\,\middle|\,X\,\middle\|\,(\mathrm{Q}_{|z}^t)_{[mn+1:mn+m]}\,\middle|\,(\mathrm{Q}_{|z}^t)_{[mn]}\right) &= \mathbb{E}_{(x,b)\sim(X,B)}\left[\log\frac{\Pr[B=b|X=x]}{\Pr[(\mathrm{Q}_{|z}^t)_{[mn+1:mn+m]}=b|(\mathrm{Q}_{|z}^t)_{[mn]}=x]}\right]\\ &\le \mathbb{E}_{(x,b)\sim(X,B)}\left[\log\frac{1}{\Pr[(\mathrm{Q}_{|z}^t)_{[mn+m]}=xb|(\mathrm{Q}_{|z}^t)_{[mn]}=x]}\right].\end{aligned}$$

For now, we assume the following claim and continue the proof.

Claim 6.6. *For each z' and each $(x,b) \sim (X,B)$,*

$$\Pr\left[(\mathrm{Q}^t_{|z})_{[mn+m]} = xb \middle| (\mathrm{Q}^t_{|z})_{[mn]} = x\right] \geq 2^{-c_0(s+\mathrm{cd}^t(x,b|z)+\gamma^{0.75}_{x,b}m+\log|z|)},$$

where $\gamma_{x,b} = |\{i \in [m] : h(x^i) \neq b^i\}|/m$ for the size-s program h in the statement.

By the claim above and Equation (15),

$$\begin{aligned}
\mathrm{KL}\Big(B \,\Big|\, X \,\Big\|\, (\mathrm{Q}^t_{|z})_{[mn+1:mn+m]} \,\Big|\, (\mathrm{Q}^t_{|z})_{[mn]}\Big) &\leq \mathbb{E}_{(x,b)}\left[c_0(s + \mathrm{cd}^t(x,b|z) + \gamma^{0.75}_{x,b}m + \log|z|)\right] \\
&= c_0 s + c_0\mathbb{E}_{(x,b)}[\mathrm{cd}^t(x,b|z)] + c_0\log|z| + c_0\mathbb{E}_{x,b}\left[\gamma^{0.75}_{x,b}\right]\cdot m \\
&\leq c_0 s + c_0(C_{\mathrm{dep}}+1)\log|z| + c_0\mathbb{E}_{x,b}\left[\gamma^{0.75}_{x,b}\right]\cdot m \\
&\leq c_0 s + c_0(C_{\mathrm{dep}}+1)\log|z| + c_0\mathbb{E}_{x,b}\left[\gamma_{x,b}\right]^{0.75}\cdot m \\
&= c_0 s + c_0(C_{\mathrm{dep}}+1)\log|z| + c_0\eta^{0.75}\cdot m,
\end{aligned}$$

where the third inequality uses the depth-goodness of z', and the last inequality follows from Jensen's inequality.

By the chain rule for the KL divergence,

$$\begin{aligned}
\frac{1}{m}\sum_{i\in[m]}\mathrm{KL}\Big(B_i \,\Big|\, XB_{<i} \,\Big\|\, (\mathrm{Q}^t_{|z})_{[mn+i]} \,\Big|\, (\mathrm{Q}^t_{|z})_{[mn+i-1]}\Big) &\leq \frac{c_0 s}{m} + \frac{c_0(C_{\mathrm{dep}}+1)\log|z|}{m} + c_0\eta^{0.75} \\
&\leq \varepsilon^2 + c_0\eta^{0.75},
\end{aligned}$$

where the last inequality follows from $m \geq c(s+\log|z|)$ and the choice of the constant C_1 in $c = C_1 c_0^2 \varepsilon^{-2}$. By choosing η_0 sufficiently small, for long enough z,

$$\mathbb{E}_{i\sim[m]}\left[\mathrm{KL}\Big(B_i|XB_{<i} \,\Big\|\, (\mathrm{Q}^t_{|z})_{[mn+i]} \,\Big|\, (\mathrm{Q}^t_{|z})_{[mn+i-1]}\Big)\right] \leq 2\varepsilon^2.$$

By Pinsker's inequality,

$$\begin{aligned}
&\mathbb{E}_{i\sim[m],xb_{<i}\sim XB_{<i}}\left[\Delta_{\mathrm{tv}}\left((B_i|XB_{<i}=xb_{<i}),((\mathrm{Q}^t_{|z})_{[mn+i]}|(\mathrm{Q}^t_{|z})_{[mn+i-1]}=xb_{<i})\right)\right] \\
&\leq \sqrt{\mathbb{E}_{i\sim[m]}\left[\mathrm{KL}\Big(B_i \,\Big|\, XB_{<i} \,\Big\|\, (\mathrm{Q}^t_{|z})_{[mn+i]} \,\Big|\, (\mathrm{Q}^t_{|z})_{[mn+i-1]}\Big)\right]}/2 \leq \varepsilon.
\end{aligned}$$

Notice that, given $xb_{<i} \sim XB_{<i}$, the probability of $((\mathrm{Q}^t_{|z})_{[mn+i]}|(\mathrm{Q}^t_{|z})_{[mn+i-1]} = xb_{<i})$ is equivalent to $\mathsf{Next}_1(xb_{<i}; \mathrm{Q}^t_{|z})$. Thus,

$$\begin{aligned}
\Pr_{i,XB_{<i},\mathsf{Next}_1}\left[\mathsf{Next}_1(xb_{<i};\mathrm{Q}^t_{|z}) \neq h(x_i)\right] &\leq \Pr_{i,XB_{<i},B_i}\left[B_i \neq h(x_i)\right] + \varepsilon \\
&\leq \eta + \varepsilon.
\end{aligned}$$

By the union bound,

$$\Pr_{i,XB_{<i},\mathsf{Next}_1}\left[\mathsf{Next}_1(xb_{<i};\mathrm{Q}^t_{|z}) \neq B_i\right] \leq 2\eta + \varepsilon,$$

which is less than 2ε for long enough z by the choice of η_0. For non-depth-good z' we lose only an $o_{|z|}(1)$ fraction, while for any z' with $m(z,z') < c(s(z,z')+\log|z|)$ the disjunction in the statement is already true. Since $\varepsilon > 0$ is arbitrary, this yields the lemma.

Now, we present the deferred proof of the claim and complete the proof of the lemma.

Proof of Claim 6.6. First, we show that

$$\Pr[\mathrm{Q}^{2t}_{|z} = xb] \geq 2^{-O(s+\gamma^{0.75}_{x,b}m+\log|z|)} \cdot \Pr[(\mathrm{Q}^{t}_{|z})_{[mn]} = x].$$

Since $t = t_m(z)$ dominates $m \cdot \tau(|z|)$, for each t-time program Π that is given z and produces x as a prefix, there exists a $2t$-time program Π' that is given z and produces xb as follows: Π' first executes Π in t steps (with auxiliary input z) and then truncates it to the first mn bits (which corresponds to x). Then, Π' executes h for each x^i in x to obtain $\tilde{b} = h(x^1) \circ \cdots \circ h(x^m)$ and takes bit-wise xor with $e \in \{0,1\}^m$, where $e_i = h(x^i) \oplus b^i$, to obtain b. Finally Π' outputs xb.

Notice that the error vector e has Hamming weight $\gamma_{x,b}m$ and by Fact 3.8, it is reconstructed from an $O(\gamma^{0.75}_{x,b}m + \log|z|)$-size program when τ is large enough. Thus, the description size of Π' is at most

$$|\Pi| + O(\log|\Pi| + |h| + \gamma^{0.75}_{x,b}m + \log|z|) \leq |\Pi| + O(|h| + \gamma^{0.75}_{x,b}m + \log|z|).$$

Thus, we have

$$\Pr[\mathrm{Q}^{2t}_{|z} = xb] \geq 2^{-O(s+\gamma^{0.75}_{x,b}m+\log|z|)} \cdot \Pr[(\mathrm{Q}^{t}_{|z})_{[mn]} = x],$$

where the probability on the right sums over all t-time programs Π that are given z and produce x as a prefix.

By contrast, we have

$$\Pr[\mathrm{Q}^{2t}_{|z} = xb] \leq \Pr[\mathrm{Q}^{t}_{|z} = xb] \cdot 2^{\mathrm{cd}^t(x,b|z)} \leq \Pr[(\mathrm{Q}^{t}_{|z})_{[mn+m]} = xb] \cdot 2^{\mathrm{cd}^t(x,b|z)}.$$

Thus, we conclude

$$\begin{aligned}\Pr[(\mathrm{Q}^{t}_{|z})_{[mn+m]} = xb | (\mathrm{Q}^{t}_{|z})_{[mn]} = x] &= \frac{\Pr[(\mathrm{Q}^{t}_{|z})_{[mn+m]} = xb]}{\Pr[(\mathrm{Q}^{t}_{|z})_{[mn]} = x]} \\ &\geq 2^{-c_0(s+\mathrm{cd}^t(x,b|z)+\gamma^{0.75}_{x,b}m+\log|z|)}\end{aligned}$$

by selecting large enough $c_0 > 0$. ◇

□

6.2 Description-Restricted Reduction to Inverting Auxiliary-Input Functions

Next, we extend the reduction in the previous section to inverting an auxiliary-input function via a description-restricted context-sensitive FAIN reduction, which completes the proof of Lemma 6.3.

Lemma 6.7. *Let $\mathcal{D} = \{\mathcal{D}_{z,z'}\}_{z\in Z, z'\in\{0,1\}^{r(z)}}$, where $Z \subseteq \{0,1\}^*$ and $r(z) \leq \mathsf{poly}(|z|)$, be a samplable distribution satisfying the assumptions of Lemma 6.5 with error function η.*

For every $\varepsilon, \delta \in (0, 1/2)$, there exist a constant $c := c_{\varepsilon,\delta} > 0$ and $\eta_0 > 0$ such that the following holds for every polynomial-time computable polynomially bounded sample-size function $m = m(z, z')$: there exist a polynomial-time nonadaptive randomized oracle machine h and an auxiliary-input function $\{f_z\}_{z\in Z}$ such that if $\eta(z) \leq \eta_0$ for every long enough $z \in Z$, then for every long enough $z \in Z$ and every oracle $\mathcal{I}$ that inverts f_z,

$$\Pr_{z'}\left[m(z,z') < c \cdot (s(z,z') + \log|z|) \text{ or } \Pr_{S}\left[\Pr_{h,(x,b)}\left[h^{\mathcal{I}}(S,x,z) = b\right] \geq 1-\varepsilon\right] \geq 1-\delta\right] \geq 1 - o_{|z|}(1),$$

where S consists of $m(z,z')$ independent samples from $\mathcal{D}_{z,z'}$, and $(x,b) \sim \mathcal{D}_{z,z'}$.

To show the lemma, we use the following result.

Theorem 6.8 ([IL90; HN23]). *For every samplable distribution $\mathcal{D} = \{\mathcal{D}_z\}_{z\in Z}$, where $Z \subseteq \{0,1\}^*$, every constant c, and every sufficiently large polynomial-time computable polynomially bounded time bound $t = t(z)$, there exist a polynomial-time randomized nonadaptive oracle machine UE and an auxiliary-input function $\{f_z\}_{z\in Z}$ such that for every long enough $z \in Z$ and every oracle $\mathcal{I}$ that inverts f_z,*

$$\Pr_{x\sim\mathcal{D}_z}\left[\forall i \in [|x|]\ \Delta_{\mathrm{tv}}\left(\mathsf{UE}^{\mathcal{I}}(z, x_{[i-1]}), \mathsf{Next}_1(x_{[i-1]}; \mathrm{Q}_{|z}^{t(z)})\right) \le \frac{1}{|z|^c}\right] \ge 1 - \frac{1}{|z|^c}.$$

Proof of Lemma 6.7. Apply Lemma 6.5 with error parameter $\varepsilon^2\delta/16$, and let c_0 be the sample-complexity constant obtained there. We take the sample-complexity constant c in this lemma to be at least c_0. Now fix an arbitrary polynomial-time computable polynomially bounded sample-size function $m = m(z, z')$, and let $t_m(z)$ be the corresponding time bound in Lemma 6.5.

We define a samplable distribution $\mathcal{D}' = \{\mathcal{D}'_z\}_{z\in Z}$ as follows: sample $z' \sim \{0,1\}^{r(z)}$, put $m = m(z, z')$, draw $(x^1, b^1), \dots, (x^m, b^m) \sim \mathcal{D}_{z,z'}$, and output the string $y = x^1 \circ \cdots \circ x^m \circ b^1 \circ \cdots \circ b^m$. We apply Theorem 6.8 for $\mathcal{D}'$ with the time bound t_m and obtain a randomized polynomial-time oracle machine UE and an auxiliary-input function $\{f_z\}_{z\in Z}$ such that for every long enough $z \in Z$ and every oracle $\mathcal{I}$ that inverts f_z,

$$\Pr_{z'\sim\{0,1\}^{r(z)}, y\sim\mathcal{D}_{z,z'}^{\otimes m(z,z')}}\left[\forall i \in [|y|]\ \Delta_{\mathrm{tv}}\left(\mathsf{UE}^{\mathcal{I}}(z, y_{[i-1]}), \mathsf{Next}_1(y_{[i-1]}; \mathrm{Q}_{|z}^{t_m(z)})\right) \le \frac{1}{|z|}\right] \ge 1 - \frac{1}{|z|^2}.$$

By Markov's inequality,

$$\Pr_{z'}\left[\Pr_{(x^1,b^1),\dots,(x^m,b^m)}\left[\forall i \in [m]\ \Delta_{\mathrm{tv}}\left(\mathsf{UE}^{\mathcal{I}}(z, xb_{<i}), \mathsf{Next}_1(xb_{<i}; \mathrm{Q}_{|z}^{t_m(z)})\right) \le \frac{1}{|z|}\right] \ge 1 - \frac{1}{|z|}\right] \ge 1 - \frac{1}{|z|},$$

where $m = m(z, z')$ and $xb_{<i} = x^1 \circ \cdots \circ x^m \circ b^1 \circ \cdots \circ b^{i-1}$.

By Lemma 6.5 and the union bound, it holds that with probability $1 - o_{|z|}(1)$ over the choice of z', either $m(z, z') < c \cdot (s(z, z') + \log|z|)$, or both

$$\Pr_{(x^1,b^1),\dots,(x^m,b^m)}\left[\forall i \in [m]\ \Delta_{\mathrm{tv}}\left(\mathsf{UE}^{\mathcal{I}}(z, xb_{<i}), \mathsf{Next}_1(xb_{<i}; \mathrm{Q}_{|z}^{t_m(z)})\right) \le \frac{1}{|z|}\right] \ge 1 - \frac{1}{|z|}$$

and

$$\Pr_{(x^1,b^1),\dots,(x^m,b^m),i,\mathsf{Next}_1}\left[\mathsf{Next}_1\left(xb_{<i}; \mathrm{Q}_{|z}^{t_m(z)}\right) = b_i\right] \ge 1 - \frac{\varepsilon^2\delta}{16}$$

hold, where again $m = m(z, z')$. Below we only consider z' for which the second statement holds; for z' satisfying the first statement, the disjunction in the statement is already true.

Now we consider a randomized polynomial-time oracle machine h defined as

$$h^{\mathcal{I}}(S, x^*, z; i) = \mathsf{UE}^{\mathcal{I}}(z, xb_{<i}^{i\to *})$$

where $i \sim [m]$ is a part of internal randomness, $m = |S|$, $xb_{<i}^{i\to *} := x^1 \circ \cdots \circ x^{i-1}x^*x^i \circ \cdots \circ x^m \circ b^1 \circ \cdots \circ b_{<i}$, and $S = \{(x^1, b^1), \dots, (x^m, b^m)\}$.

When $(x^1, b^1), \dots, (x^m, b^m)$ and (x^*, b^*) are independently and identically distributed according to $\mathcal{D}_{z,z'}$, the pair $(xb_{<i}^{i\to *}, b^*)$ is identically distributed as $(xb_{<i}, b^i)$. Thus, it holds that

$$\Pr_{S,x^*}\left[\Delta_{\mathrm{tv}}\left(h^{\mathcal{I}}(S, x^*, z; i), \mathsf{Next}_1(xb_{<i}^{i\to *}; \mathrm{Q}_{|z}^{t_m(z)})\right) \le \frac{1}{|z|}\right] \ge 1 - \frac{1}{|z|}$$

where $i \sim [m]$, and

$$\Pr_{S,(x^*,b^*),i,\mathsf{Next}_1}\left[\mathsf{Next}_1\left(xb^{i\to *}_{<i};\mathrm{Q}^{t_m(z)}_{|z}\right)=b^*\right]\geq 1-\frac{\varepsilon^2\delta}{16}.$$

By Markov's inequality,

$$\Pr_{S,x^*}\left[\Pr_{b^*,i,\mathsf{Next}_1}\left[\mathsf{Next}_1\left(xb^{i\to *}_{<i};\mathrm{Q}^{t_m(z)}_{|z}\right)=b^*\right]\geq 1-\frac{\varepsilon}{4}\right]\geq 1-\frac{\varepsilon\delta}{4}.$$

By the union bound, it holds that with probability at least $1-(\varepsilon\delta/4+1/|z|)\geq 1-\varepsilon\delta/2$ over the choice of S and x^*,

$$\begin{aligned}
&\Pr_{b^*,h}\left[h^{\mathcal{I}}(S,x^*,z;i)=b^*\right]\\
&\quad\geq \Pr_{b^*,i,\mathsf{Next}_1}\left[\mathsf{Next}_1\left(xb^{i\to *}_{<i};\mathrm{Q}^{t_m(z)}_{|z}\right)=b^*\right]-\Delta_{\mathrm{tv}}\left(h^{\mathcal{I}}(S,x^*,z;i),\mathsf{Next}_1(xb^{i\to *}_{<i};\mathrm{Q}^{t_m(z)}_{|z})\right)\\
&\quad\geq 1-\frac{\varepsilon}{4}-\frac{1}{|z|}\geq 1-\frac{\varepsilon}{2}. \qquad (16)
\end{aligned}$$

By Markov's inequality,

$$\Pr_{S}\left[\Pr_{x^*}\left[\textit{Equation (16)} \text{ holds}\right]\geq 1-\frac{\varepsilon}{2}\right]\geq 1-\delta.$$

Thus, with probability at least $1-\delta$ over S,

$$\Pr_{(x^*,b^*),h}\left[h^{\mathcal{I}}(S,x^*,z;i)\neq b^*\right]\leq \frac{\varepsilon}{2}\cdot 1+1\cdot\frac{\varepsilon}{2}=\varepsilon.$$

Therefore, combining this conclusion with the first statement $m(z,z')<c\cdot(s(z,z')+\log|z|)$, we obtain

$$\Pr_{z'}\left[m(z,z')<c\cdot(s(z,z')+\log|z|) \text{ or } \Pr_{S}\left[\Pr_{h,(x^*,b^*)}\left[h^{\mathcal{I}}(S,x^*,z)=b^*\right]\geq 1-\varepsilon\right]\geq 1-\delta\right]\geq 1-o_{|z|}(1).$$

Since UE is nonadaptive, it is easily verified that h is also nonadaptive. □

Now, we prove Lemma 6.3.

Proof of Lemma 6.3. For a large enough constant C, fix a sufficiently large $\ell\in\mathbb{N}$ and consider $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{C\cdot\ell}\mathfrak{S}_\ell$-DMMSA.

Let $z=(\mathcal{D},w,1^n,s)$ be an instance of $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{C\cdot\ell}\mathfrak{S}_\ell$-DMMSA. Let R be the reduction in Lemma 5.1, where we select a large enough polynomial $\lambda(\cdot)$ and let $\lambda:=\lambda(|z|)$. Then $R(z,z')$, where $z'\sim\{0,1\}^\lambda$, produces a distribution $\mathcal{E}_{z,z'}$ over samples. Let $N_{z,z'}$ denote the length of a sample after identifying (x,b) with $x\circ b$. Recall from the proof of Lemma 5.1 that, in addition to the formal advice-complexity witness via U, the construction gives a fixed sampler S_R with ℓ bits of local advice. Let q_R be the seed-length polynomial of S_R. Thus, for each (z,z'), there exists a function

$$\beta_{z,z'}:\{0,1\}^{q_R(N_{z,z'})}\to\{0,1\}^\ell$$

such that the distribution of

$$S_R(1^{N_{z,z'}},\beta_{z,z'}(r),r)$$

for $r \sim \{0,1\}^{q_R(N_{z,z'})}$ is statistically identical to $\mathcal{E}_{z,z'}$. For the concrete choice of $\beta_{z,z'}$ given in the proof of Lemma 5.1, the map $r \mapsto \beta_{z,z'}(r)$ is polynomial-time computable given (z, z'). Moreover, S_R is total, runs in polynomial time, and never outputs $\perp$.

Let $\varepsilon_{\mathrm{ext}} := 1/4 - 5\gamma/4$. Let c_{ext} be the sample-complexity constant guaranteed by Lemma 6.7 for $\varepsilon = \varepsilon_{\mathrm{ext}}$ and $\delta = 1/32$. We apply Lemma 6.7 to the polynomial-time computable polynomially bounded sample-size function

$$m_0(z, z') := \lceil c_{\mathrm{ext}}(\lceil 2s\lambda \rceil + \log |z|) \rceil ,$$

and obtain h and $\{f_z\}$. For the current instance, write $s' := \lceil 2s\lambda \rceil$ and $m := m_0(z, z')$; this value is independent of z'. We also define a randomized oracle machine H given access to $\mathcal{I}$ as follows:

$$H^{\mathcal{I}}(x; \bar{r}, z, z') := h^{\mathcal{I}}(S_{z,z',\bar{r}}, x, z).$$

Here $\bar{r} = (r^1, \ldots, r^m)$ is composed of m random seeds for S_R, and

$$S_{z,z',\bar{r}} = \{(x^1, b^1), \ldots, (x^m, b^m)\},$$

where

$$(x^i, b^i) = S_R(1^{N_{z,z'}}, \beta_{z,z'}(r^i), r^i)$$

for each $i \in [m]$. By construction,

$$m = O(s' + \log |z|) = O(\lambda s + \log |z|).$$

In the completeness argument below, Lemma 5.1 gives s' as a uniform upper bound on the hypothesis size for all z', so this choice of m makes the lower-bound statement in Lemma 6.7 false for every z'.

For each $z' \in \{0,1\}^{\lambda}$, $\bar{r} \in (\{0,1\}^{q_R(N_{z,z'})})^m$, and $\rho \in \{0,1\}^{a(|z|)}$ (where a is a computable function), we define the event $E_{z,z',\bar{r}}$ as follows:

> Event $E^{\rho}_{z,z',\bar{r}}$: We perform the empirical estimation of the quantity
>
> $$\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}[H^{\mathcal{I}(\text{-},z;\rho)}(x; \bar{r}, z, z') = b]$$
>
> with additive accuracy error $\pm(1/16 - 5\gamma/16)$ and failure probability at most $2^{-|z|}$ by using polynomially (in $|z|$) many samples from $\mathcal{E}_{z,z'}$. Let $\tilde{p}$ be the approximation value. Then, it holds that $\tilde{p} \geq 5/8 + 15\gamma/8$.

Notice that given $(z, z', \bar{r})$, the trial whether $E^{\rho}_{z,z',\bar{r}}$ occurs is performed in polynomial time in $|z|$ given access to $\mathcal{I}(\text{-}, z; \rho)$.

Now we describe the polynomial-time randomized oracle machine A that attempts to solve $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{C.\ell}\mathfrak{S}_{\ell}$-DMMSA. For a given instance z and oracle access to $\mathcal{I}(\text{-}, z; \rho)$, the algorithm $A^{\mathcal{I}(\text{-},z;\rho)}(z)$ empirically estimates the quantity $\Pr_{z',\bar{r},E}[E^{\rho}_{z,z',\bar{r}}]$ with additive accuracy error $\pm 1/32$ and failure probability at most $2^{-|z|}$. This is performed in polynomial time in $|z|$, and the queries to $\mathcal{I}$ are made nonadaptively since $h^{\mathcal{I}}$ is nonadaptive. Let $\tilde{q}$ be the estimation value. If $\tilde{q} \geq 13/16$, then A outputs 1; otherwise, A outputs 0.

We have seen that A is nonadaptive. Thus, it suffices to show the following claims:

Claim 6.9. *For every oracle $\mathcal{O}$, computable function a, and for every long enough instance z, if z is an yes instance, and $\mathcal{O}(\text{-}, z; \rho)$ inverts f_z for all $\rho \in \{0,1\}^{a(|z|)}$, then*

$$\Pr_{A,\rho}[A^{\mathcal{O}(\text{-},z;\rho)}(z) = 1] \geq 1 - 2^{-|z|}.$$

Claim 6.10. *For each total Turing machine I, computable function a, and for every long enough instance z, if z is a no instance, and $\mathcal{O}_I(\text{-}, z; \rho)$ inverts f_z for all $\rho \in \{0,1\}^{a(|z|)}$, then*

$$\Pr_{A,\rho}[A^{\mathcal{O}_I(\text{-},z;\rho)}(z) = 0] > 3/4.$$

Proof of Claim 6.9. It suffices to show that $\Pr_{z',\bar{r},E}[E^{\rho}_{z,z',\bar{r}}] \geq 7/8$ for all ρ because it implies that

$$\tilde{q} \geq \Pr_{z',\bar{r},E}[E^{\rho}_{z,z',\bar{r}}] - 1/32 \geq 27/32 > 13/16$$

with probability at least $1 - 2^{-|z|}$ over the choice of randomness. In this case, A outputs 1.

When z is an yes instance, Lemma 5.1 shows that for every z', there exists a polynomial-time program h^* of size $s' = \lceil 2s\lambda \rceil$ such that

$$\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}[h^*(x) = b] \geq 1 - \varepsilon(\ell).$$

Thus, as long as z is an yes instance and ℓ is sufficiently large, $\mathcal{E}_{z,z'}$ satisfies the conditions of Lemma 6.5. For all $\rho \in \{0,1\}^{a(|z|)}$, Lemma 6.7 (recall that it was applied for $\varepsilon_{\text{ext}} = 1/4 - 5\gamma/4$ and $\delta = 1/32$) shows that if $\mathcal{O}(\text{-}, z; \rho)$ inverts f_z, then

$$\Pr_{z',S}\left[\Pr_{h,(x,b)\sim\mathcal{E}_{z,z'}}\left[h^{\mathcal{O}(\text{-},z;\rho)}(S,x,z) = b\right] \geq \frac{3}{4} + \frac{5\gamma}{4}\right] \geq 1 - \frac{1}{32} - o(1) \geq \frac{15}{16}$$

for long enough z. Namely, it holds that

$$\Pr_{z',\bar{r}}\left[\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}(\text{-},z;\rho)}(x;\bar{r},z,z') = b\right] \geq \frac{3}{4} + \frac{5\gamma}{4}\right] \geq \frac{15}{16}.$$

For any z' and $\bar{r}$ satisfying the event above, with probability at least $1 - 2^{-|z|}$ over the choice of randomness for $E^{\rho}_{z,z',\bar{r}}$,

$$\tilde{p} \geq \Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}(\text{-},z;\rho)}(x;\bar{r},z,z') = b\right] - (1/16 - 5\gamma/16) \geq 11/16 + 25\gamma/16 > 5/8 + 15\gamma/8,$$

where the last inequality follows from $\gamma < 1/5$.

Thus,

$$\Pr_{z',\bar{r},E}[E^{\rho}_{z,z',\bar{r}}] \geq \frac{15}{16} \cdot (1 - 2^{-|z|}) \geq \frac{7}{8},$$

as desired. ◇

Proof of Claim 6.10. For each instance z, we define the test T_z for $\bar{r}$ and ρ as follows:

$$T_z(\bar{r},\rho) = 1 \iff \Pr_{z'\sim\{0,1\}^{\lambda}}\left[\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}_I(\text{-},z;\rho)}(x;\bar{r},z,z') = b\right] \geq \frac{1}{2} + \frac{5\gamma}{2}\right] \geq \frac{1}{2}.$$

We claim that if z is long enough, then

$$\Pr_{A,\rho}[A^{\mathcal{O}_I(\text{-},z;\rho)}(z) = 1] \geq 1/4 \implies \Pr_{\bar{r},\rho}[T_z(\bar{r},\rho) = 1] \geq 1/16. \tag{17}$$

First, we assume (17) and prove the claim by contraposition. We assume that $\Pr_{A,\rho}[A^{\mathcal{O}_I(\text{-},z;\rho)}(z) = 0] \leq 3/4$ and derive that z is not a no instance.

Since A always outputs 0 or 1, we have $\Pr_{A,\rho}[A^{\mathcal{O}_I(\cdot,z;\rho)}(z)=1] \geq 1/4$. Thus, by (17), we have $\Pr_{\bar{r},\rho}[T_z(\bar{r},\rho)=1] \geq 1/16$.

Let $G_z^I = \{(\bar{r},\rho) : T_z(\bar{r},\rho)=1\}$. Since $\Pr_{\bar{r},\rho}[(\bar{r},\rho) \in G_z^I] = \Pr_{\bar{r},\rho}[T_z(\bar{r},\rho)=1] \geq 1/16 > 0$, it holds that $G_z^I \neq \emptyset$. In addition, because I is fixed and total, the set G_z^I is decidable by exhaustive enumeration given (z,I): the number of possible z' and samples is finite, every computation of I halts, and the map $r \mapsto \beta_{z,z'}(r)$ is computable given (z,z'). Thus, by letting $(\bar{r}^*,\rho^*)$ be the lexicographically first element in G_z^I, we have

$$\mathrm{K}(\bar{r}^*,\rho^*) \leq O(\mathrm{K}(G_z^I)) \leq O(|z|+|I|).$$

Since $T_z(\bar{r}^*,\rho^*)=1$,

$$\Pr_{z'\sim\{0,1\}^\lambda}\left[\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}_I(\cdot,z;\rho^*)}(x;\bar{r}^*,z,z')=b\right] \geq \frac{1}{2}+\frac{5\gamma}{2}\right] \geq \frac{1}{2}.$$

Remember that

$$H^{\mathcal{O}_I(\cdot,z;\rho^*)}(x;\bar{r},z,z') = h^{\mathcal{O}_I(\cdot,z;\rho^*)}(S_{z,z',\bar{r}},x,z),$$

where

$$(x^i,b^i) = S_R(1^{N_{z,z'}},\beta_{z,z'}(r^i),r^i)$$

for each $i \in [m]$. Fix z' and write $\bar{r}^* = (r^{*,1},\dots,r^{*,m})$. Let $a_i := \beta_{z,z'}(r^{*,i}) \in \{0,1\}^\ell$ for each $i \in [m]$. Consider the time-unbounded program $h_{z'}^*$ that hardwires z, I, a shortest description of $(\bar{r}^*,\rho^*)$, and $a_1,\dots,a_m$. On input x, it reconstructs $S_{z,z',\bar{r}^*}$ by computing

$$(x^i,b^i) = S_R(1^{N_{z,z'}},a_i,r^{*,i})$$

for each $i \in [m]$, and then outputs $h^{\mathcal{O}_I(\cdot,z;\rho^*)}(S_{z,z',\bar{r}^*},x,z)$. Since $|z'|=\lambda$ is determined by z, the description size of $h_{z'}^*$ (as a time-unbounded program) is at most

$$\mathrm{K}(\bar{r}^*,\rho^*)+|z|+|I|+m\ell+O(\log|z|) \leq O(\lambda s+\log|z|)\ell+O(|z|+|I|).$$

Since the DMMSA size parameter s is inverse-polynomially lower bounded and the machine I is fixed in this soundness argument, by choosing the polynomial $\lambda=\lambda(|z|)$ sufficiently large we have, for every long enough z,

$$O(\lambda s+\log|z|)\ell+O(|z|+|I|) \leq C_0\cdot\lambda s\ell$$

for a large enough constant $C_0>0$.

Namely, with probability at least $1/2$ over the choice of z', there exists a program h^* of description size at most $C_0\cdot\ell\cdot s\cdot\lambda$ satisfying that

$$\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}[h^*(x)=b] \geq \frac{1}{2}+\frac{5\gamma}{2}.$$

By choosing C sufficiently large, say so that $C_0 \leq 0.98C$, and applying Lemma 5.1, z is not a no instance of $\mathrm{Gap}_{C\cdot\ell}^{\varepsilon(\ell),\gamma}\mathfrak{S}_\ell$-DMMSA.

In the remainder, we observe the implication (17), which completes the proof.

Suppose that $\Pr_{A,\rho}[A^{\mathcal{O}_I(\cdot,z;\rho)}(z)=1] \geq 1/4$. Then, it holds that $\Pr_{z',\bar{r},E}[E^\rho_{z,z',\bar{r}}] \geq 25/32$ with probability at least $1/8$ over ρ; otherwise, for at least $(7/8)$-fraction of ρ, with probability $1-2^{-|z|}$ over the choice of randomness for A,

$$\tilde{q} \leq \Pr_{z',\bar{r},E}[E^\rho_{z,z',\bar{r}}]+1/32 < 26/32 = 13/16,$$

and $\Pr_A[A^{\mathcal{O}_I(\cdot,z;\rho)}(z)=1] = 1 - \Pr_A[A^{\mathcal{O}_I(\cdot,z;\rho)}(z)=0] \leq 2^{-|z|}$. Thus,

$$\Pr_{A,\rho}[A^{\mathcal{O}_I(\cdot,z;\rho)}(z)=1] \leq \frac{1}{8} + 2^{-|z|} < \frac{1}{4},$$

which contradicts the assumption.

Furthermore, for any ρ satisfying that $\Pr_{z',\bar{r},E}[E^{\rho}_{z,z',\bar{r}}] \geq 25/32$, it holds that

$$\Pr_{z',\bar{r}}\left[\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}_I(\cdot,z;\rho)}(x;\bar{r},z,z')=b\right] \geq \frac{1}{2}+\frac{5\gamma}{2}\right] \geq \frac{3}{4}. \tag{18}$$

Otherwise, with probability at least $1/4$ over z' and $\bar{r}$, it holds that with probability at least $1-2^{-|z|}$ over the choice of randomness in $E^{\rho}_{z,z',\bar{r}}$,

$$\begin{aligned}\tilde{p} &\leq \Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}_I(\cdot,z;\rho)}(x;\bar{r},z,z')=b\right] + (1/16 - 5\gamma/16)\\ &< (1/2) + 5\gamma/2 + 1/16 - 5\gamma/16 = 9/16 + 35\gamma/16 < 5/8 + 15\gamma/8,\end{aligned}$$

where the last inequality follows from $\gamma < 1/5$. Therefore, $\Pr_{z',\bar{r},E}[\neg E_{z,z',\bar{r}}] \geq (1/4)\cdot(1-2^{-|z|}) > 7/32$, which contradicts that $\Pr_{z',\bar{r},E}[E^{\rho}_{z,z',\bar{r}}] \geq 25/32$.

By Markov's inequality, Equation (18) implies

$$\Pr_{\bar{r}}[T_z(\bar{r},\rho)=1] = \Pr_{\bar{r}}\left[\Pr_{z'}\left[\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}_I(\cdot,z;\rho)}(x;\bar{r},z,z')=b\right] \geq \frac{1}{2}+\frac{5\gamma}{2}\right] \geq \frac{1}{2}\right] \geq \frac{1}{2}.$$

Thus, we obtain

$$\Pr_{\bar{r},\rho}[T_z(\bar{r},\rho)=1] \geq \frac{1}{8}\cdot\frac{1}{2} = \frac{1}{16},$$

as desired. ◇

□

6.3 CoAM Bound for Description-Restricted Context-Sensitive FAIN Reductions

In this section, we present the proof of Theorem 6.4. Below, we consider AM protocols as satisfying the following modified syntax: Any verifier V of an AM protocol rejects the given instance (as the standard formulation) by outputting $\perp$ or halts and outputs some message in $\{0,1\}^*$ (which means accepting). For an AM-protocol (P,V) and common input x, let $\langle P,V\rangle(x)$ denote the resulting message.

The following is the crucial lemma.

Lemma 6.11 ([AGGM06; ABX08])**.** *For every polynomial-time FAIN reduction R to inverting an auxiliary-input $f=\{f_x\}$, there exist an AM protocol (P,V), a total Turing machine M, and a polynomial p such that*

- *$\mathcal{O}_M(\cdot,x;\rho)$ inverts f_x for all $x\in\{0,1\}^*$ and $\rho\in\{0,1\}^{p(|x|)}$;*
- *for all large enough x,*

$$\Delta_{\mathrm{tv}}\left(\langle P,V\rangle(x), \mathcal{O}_M(\cdot,x;\rho)|_{R(x)}\right) \leq \frac{1}{|x|},$$

where $\mathcal{O}_M(\cdot,x;\rho)|_{R(x)} = (\mathcal{O}_M(q^1,x;\rho),\ldots,\mathcal{O}_M(q^k,x;\rho))$ for queries $q^1,\ldots,q^k$ produced by $R(x)$;

- *for all large enough x and for any prover $\tilde{P}$,*

$$\Delta_{\mathrm{tv}}\left(\langle \tilde{P}, V\rangle(x), \mathcal{O}_M(\text{-}, x; \rho)|_{R(x)}\right) \leq \Pr_{r_R,\rho}\left[\langle \tilde{P}, V\rangle(x) = \perp\right] + \frac{1}{|x|}.$$

The proof is implicit in [AGGM06]. Below, we highlight the ideas.

Proof (sketch). For each polynomial-time FAIN reduction R, the AM-protocol (P, V) performs as follows on common input x (let $k := k(|x|)$ be the query complexity):

- Let z_x be the unique auxiliary-input queried by $R(x)$. Although R queries strings of the form (z_x, y), we omit the first element z_x below for readability.
- The verifier V randomly selects a threshold τ from polynomially many candidates from $[2, 3]$ and defines τ-light and τ-heavy queries as follows:

$$y \text{ is } \tau\text{-heavy} \iff \Pr[y \sim Q_x] \geq \tau \cdot \Pr_r[y = f_{z_x}(r)]$$
$$y \text{ is } \tau\text{-light} \iff \Pr[y \sim Q_x] < \tau \cdot \Pr_r[y = f_{z_x}(r)],$$

  where Q_x is a distribution of queries by $R(x)$ at a position selected uniformly at random.
- The verifier V execute $R(x)$ $\ell := \mathsf{poly}(|x|)$ times and obtain ℓ *sets* of queries $\bar{y}^{(1)}, \ldots, \bar{y}^{(\ell)}$ (i.e., each $\bar{y}^{(i)}$ is composed of k queries). Then, V sends them and make the prover return the value of $|f_{z_x}^{-1}(y)|$ for each query y contained in the sets with the claim whether y is τ-heavy or τ-light (building upon the lower-bound protocol, the entropy estimation protocol, and the hiding protocol, see [AGGM06, Appendix D]). As a result, with probability at least $1 - 1/(4|x|)$ over the choice of verifier's randomness (used up to this stage), it holds that (i) the honest prover P that sends the correct sizes and claims is accepted, and (ii) as long as the prover is not rejected, all but $1/p(|x|)$-fraction sets $\bar{y}^{(i)}$ satisfy that the claims about τ-heavy and τ-light are correct, and for all τ-light queries y in it, the claimed size $\tilde{s}_y$ satisfies that

$$(1 - 1/p(|x|))|f_{z_x}^{-1}(y)| \leq \tilde{s}_y \leq (1 + 1/p(|x|))|f_{z_x}^{-1}(y)|. \tag{19}$$

  where p is a sufficiently large polynomial.
- Next, V selects a value j from $0, 1, \ldots, m-1$ uniformly at random, where $m = \mathsf{poly}(n)$ is a large enough polynomial. Then, for each claimed size $\tilde{s}_y$, the verifier calculates

$$\tilde{\ell}_y = \lfloor \log(\tilde{s}_y/(1 + j/m)) \rfloor$$

  and selects a $\mathsf{poly}(n)$-wise independent hash function $h_y \colon \{0,1\}^{n_x} \to \{0,1\}^{\tilde{\ell}_y - c\log n}$, where poly is a large enough polynomial, n_x is the input size of f_{z_x}, and c is a large enough absolute constant selected. Then, we can show that with probability at least $1 - 1/(4|x|)$ over the choice of j, it holds that

$$\tilde{\ell}_y = \lfloor \log |f_{z_y}^{-1}(y)| \rfloor \tag{20}$$

  for all τ-light queries y satisfying Equation (19) (see [AGGM06, Lemma 11.1]). Furthermore, by letting h_y random enough, we can observe that with probability $1 - 1/(4|x|)$, it holds that the size of $I_y = h_y^{-1}(0^{\tilde{\ell}_y - c\log n}) \cap f_{z_x}^{-1}(y)$ is bounded by a fixed polynomial $q(|x|)$ for all τ-light queries y satisfying Equations (19) and (20). Thus, V can expect the prover to send

all elements in I_y for such y's. Let $\tilde{I}_y \subseteq I_y$ be the set sent by the prover. Note that whether $\tilde{I}_y \subseteq I_y$ is easily checked. In addition, V checks whether $|\tilde{I}_y| \leq q(|x|)$ for all y claimed to be τ-light, and the average of $|\tilde{I}_y|$ (taken over y claimed as τ-light) is enough close to the expected size (i.e., a fixed polynomial), and if not, V rejects the proof. We can show that the average of $|I_y|$ (i.e., the correct value) passes this test with probability at least $1 - 1/(8|x|)$ (over the choice of $\{h_y\}$).

- We can prove that for j and $\{h_y\}$ satisfying the events above (selected with probability at least $1 - 5/(8|x|)$ by the union bound), (i) the honest prover P that sends I_y correctly is not rejected, and (ii) as long as the prover is not rejected, the prover must send correct I_y for each τ-light query y in all but $1/(8|x|)$-fraction of query sets (we call such a query set *good*), and these *good* sets are a subset of the sets for which the prover sends the correct claim about τ-heavy and τ-light (see [AGGM06, Theorem 10 and Remark 9]).

- Finally, the verifier selects the random position i from $[\ell]$ and answers the i-th query set as follows: for each query y in the set, (i) if y is claimed as τ-heavy, then V does not return any inverse, and (ii) if y is claimed as τ-light, then V returns the lexicographically first element in I'_y (this must be an inverse of y). Notice that, as long as the i-th query set is *good*, V answers each τ-light query y by the lexicographically first element in I_y.

Now, we define a Turing machine M as follows: On input (z_x, y, x) and shared randomness ρ_{shared} and (independent) random seeds ρ_{ind},

- M first selects the threshold $\tau \in [2, 3]$ by using ρ_{shared} as V does.

- Then, M determines whether the query is τ-light or τ-heavy according to R and f_{z_x} (which takes exponential time).

- If y is τ-heavy, then M does not return any inverse of y.

- If y is τ-light, then M selects $\mathsf{poly}(n)$-wise independent hash function $h_y \colon \{0,1\}^{n_z} \to \{0,1\}^{\ell_y - c\log n}$ by using ρ_{ind} as V does, where n_z is the input size of f_z and $\ell_y = \lfloor \log |f_z^{-1}(y)| \rfloor$. Then M returns lexicographically the first element in $I_y = h_y^{-1}(0^{\ell_y - c\log n}) \cap f_z^{-1}(y)$ (if $I_y = \emptyset$ and $f_z^{-1}(y) \neq \emptyset$, M returns some inverse of y).

We can observe that the M-based oracle $\mathcal{O}_M(\text{-}, x; \rho)$ inverts $\{f_{z_x}\}$ for all x and ρ because for any choice of $\tau \in [2, 3]$, M returns some inverse as long as $y \in \mathrm{Im} f_z$ is τ-light, and it holds that

$$\Pr_r[f_{z_x}(r) \text{ is } \tau\text{-heavy}] = \sum_{y:\tau\text{-heavy}} \Pr_r[y = f_{z_x}(r)] \leq \sum_{y:\tau\text{-heavy}} \tau^{-1} \Pr[y \sim Q_x] \leq 1/2.$$

Namely,

$$\Pr_r\left[\mathcal{O}_M(z_x, f_{z_x}(r), x; \rho) \in f_{z_x}^{-1}(f_{z_x}(r))\right] \geq 1/2.$$

We have seen that, with probability at least

$$1 - \left(\frac{1}{4|x|} + \frac{5}{8|x|} + \frac{1}{8|x|}\right) = 1 - \frac{1}{|x|}$$

over the choice of V's randomness, any prover $\tilde{P}$ is forced to answer to the queries produced by $R(x)$ as $\mathcal{O}(\text{-}, x; \rho)$, i.e., no inverse for all τ-heavy queries and lexicographically the first element in $I_y = h_y^{-1}(0^{\ell_y - c\log n}) \cap f_z^{-1}(y)$ for all τ-light queries y unless it is rejected. Thus, we obtain that

$$\Delta_{\mathrm{tv}}\left(\langle\tilde{P},V\rangle(x),\mathcal{O}_M(\text{-},x;\rho)|_{R(x)}\right)\leq\Pr_{r_R,\rho}\left[\langle\tilde{P},V\rangle(x)=\perp\right]+\frac{1}{|x|}.$$

By the correctness for the honest prover P, we also obtain that

$$\Delta_{\mathrm{tv}}\left(\langle P,V\rangle(x),\mathcal{O}_M(\text{-},x;\rho)|_{R(x)}\right)\leq\frac{1}{|x|}.$$

□

Now, we complete the proof of Theorem 6.4.

Proof of Theorem 6.4. Let R be a description-restricted context-sensitive FAIN reduction from a promise problem Π to inverting an auxiliary-input function $\{f_z\}$. Let (P_0,V_0), M, and p be the AM protocol, Turing machine, and polynomial obtained by applying Lemma 6.11 for R, respectively.

We construct a coAM-protocol (P,V) as follows: For a given common input x, (P,V) first executes $R(x)$ to obtain a query set and $(P_0,V_0)(x)$, where V_0 embeds the query set at the random position i on which V outputs answers (see the proof of Lemma 6.11). If V_0 rejects the proof, then V outputs 0. Otherwise, V_0 must output an answer set for the query by $R(x)$. In this case, V continues executing $R(x)$ with the answer set and outputs 1 if $R(x)=0$ (otherwise, V outputs 0).

First, we show the completeness. Suppose that the given x is long enough and a no instance. Then, by Lemma 6.11,

$$\Pr[\langle P,V\rangle(x)=1]\geq\Pr_{R,\rho}[R^{O_M(\text{-},x;\rho)}(x)=0]-\Delta_{\mathrm{tv}}\left(\langle P_0,V_0\rangle(x),\mathcal{O}_M(\text{-},x;\rho)|_{R(x)}\right)\geq\frac{3}{4}-\frac{1}{|x|}. \tag{21}$$

Next, we show the soundness. Let $\tilde{P}$ be an arbitrary prover. Note that $\tilde{P}$ induces a prover $\tilde{P}_0$ for V_0 since V first performs as V_0. Let x be a long enough yes instance.

For contradiction, suppose that

$$\Pr[\langle\tilde{P},V\rangle(x)=1]>2/3.$$

Then, it must hold that $\Pr[\langle\tilde{P}_0,V\rangle(x)=\perp]<1/3$ since V outputs 0 in such cases. Thus, by Lemma 6.11,

$$\Pr[\langle\tilde{P},V\rangle(x)=1]\leq\Pr_{R,\rho}[R^{O_M(\text{-},x;\rho)}(x)=0]+\Delta_{\mathrm{tv}}\left(\langle\tilde{P}_0,V_0\rangle(x),\mathcal{O}_M(\text{-},x;\rho)|_{R(x)}\right)<\frac{1}{4}+\frac{1}{3}+\frac{1}{|x|}<\frac{2}{3},$$

which is a contradiction. Therefore, we have

$$\Pr[\langle\tilde{P},V\rangle(x)=1]\leq 2/3. \tag{22}$$

From Equations (21) and (22), we conclude that $\Pi\in\mathsf{coAM}$.

□

6.4 CoAM bound for Problems Reducible to GapLearn

In this section, we prove Theorem 1.12 using an approach similar to that of Theorem 6.1. We begin by formally defining the relevant concepts.

We say that a promise problem $\Pi=(\Pi_{\mathrm{yes}},\Pi_{\mathrm{no}})$ is paddable if, for every polynomial p, there exists a polynomial-time computable function f such that for every $x\in\Pi_{\mathrm{yes}}\cup\Pi_{\mathrm{no}}$,

- $|f(x)|\geq p(|x|)$, and

- if $x \in \Pi_{\text{yes}}$ (resp. $x \in \Pi_{\text{no}}$), then $f(x) \in \Pi_{\text{yes}}$ (resp. $f(x) \in \Pi_{\text{no}}$).

We also define a parametric-honest nonadaptive reduction from a promise problem Π to $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ as a polynomial-time randomized oracle machine that satisfies the standard properties of a nonadaptive reduction and, additionally, the following: there exists a constant $\xi > 0$ such that for every instance x of Π, every query $(1^n, 1^s, \mathcal{E})$ made by $R(x)$ satisfies $n \geq |x|^{\xi}$ and $s \geq |x|^{\xi}$.

Note that if a paddable problem is reducible to $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ via a parametric-honest nonadaptive reduction, then for any fixed polynomial p, we can assume that the reduction $R(x)$ only makes queries to instances $(1^n, 1^s, \mathcal{E})$ such that $n \geq p(|x|)$ and $s \geq p(|x|)$, by padding the original instance to one of sufficiently large polynomial length relative to p and ξ.

Because of Theorem 6.4, it suffices to show the following lemma for Theorem 1.12.

Lemma 6.12. *For every constant $\gamma \in (0,1)$, there exists a constant C such that for every $\varepsilon(\ell) = o(1)$ and every sufficiently large $\ell \in \mathbb{N}$, the following holds: If a paddable promise problem Π is reducible to $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{C\cdot\ell}\mathrm{Learn}[\ell]$ via a parametric-honest nonadaptive reduction, then Π is reducible to inverting an auxiliary-input function $f = \{f_z\}$ via a description-restricted context-sensitive FAIN reduction.*

Proof. Let z be an instance of Π, and let R be a parametric-honest nonadaptive reduction. Without loss of generality, by randomly permuting the queries, we may assume that the marginal distribution of each query is identical and can be sampled given z simply by executing R. Let $\lambda(\cdot)$ be a sufficiently large polynomial to be specified later. Since Π is paddable, we can assume that every query $(1^n, 1^s, \mathcal{E})$ in the domain satisfies $n, s \geq \lambda(|z|)$, and we define $\lambda := \lambda(|z|)$.

For each randomness $z' \sim \{0,1\}^{\mathsf{poly}(|z|)}$ used by the sampler for the marginal query distribution, let $\mathcal{E}_{z,z'}$ denote the corresponding distribution over samples, let $n(z,z')$ denote its sample length, and let $s_q(z')$ denote the size parameter in the corresponding GapLearn query. Since the queries are instances of $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{C\cdot\ell}\mathrm{Learn}[\ell]$, by Definition 3.11, for each pair (z,z') there exists a function

$$\alpha_{z,z'} : \{0,1\}^{2^{2(n(z,z')+1)}} \to \{0,1\}^{\leq \ell}$$

such that $U(\alpha_{z,z'}(r), r, n(z,z')+1)$ outputs $\perp$ with probability at most $1/4$, and, conditioned on a non-$\perp$ output, the resulting distribution of labeled examples is within statistical distance $2^{-2(n(z,z')+1)}$ of $\mathcal{E}_{z,z'}$. Thus, samples can be approximated by rejection sampling from U, discarding $\perp$ outcomes. Let $q := q(|z|)$ be the query complexity of R, and let $p(|z|) := 4q(|z|)$.

Let $c := c_\gamma \geq 1$ be the sample-complexity constant obtained from Lemma 6.7, and define the polynomial-time computable polynomially bounded sample-size function

$$m_0(z,z') := \left\lceil c \cdot (s_q(z') + \log|z|) \right\rceil.$$

Applying Lemmas 6.5 and 6.7 to this sample-size function, we obtain a randomized oracle machine h and a polynomial-time computable function $f = \{f_z\}$ such that for every long enough $z \in \{0,1\}^*$ and every oracle $\mathcal{I}$ that inverts f_z,

$$\Pr_{z'}\left[m_0(z,z') < c\cdot(s^*(z') + \log|z|) \text{ or } \Pr_{S,h}\left[\Pr_{(x,b)}\left[h^{\mathcal{I}}(S,x,z) = b\right] \geq \frac{3}{4} + \frac{\gamma}{4}\right] \geq \frac{3}{4}\right] \geq 1 - \frac{1}{4p(|z|)}, \tag{23}$$

where S consists of $m_0(z,z')$ independent samples from $\mathcal{E}_{z,z'}$, $(x,b) \sim \mathcal{E}_{z,z'}$, and $s^*(z')$ is the minimum size polynomial-time program h such that

$$\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}[h(x) = b] \geq 1 - \varepsilon(\ell).$$

We present an efficient and nonadaptive oracle machine A such that for every total Turing machine M and for every long enough z, if $M(\text{-}, z)$ inverts f_z, the algorithm $A^{M(\text{-},z)}$ solves $\mathrm{Gap}_{C\cdot\ell}^{\varepsilon(\ell),\gamma}\mathrm{Learn}[\ell]$ correctly for all z' satisfying the event in Equation (23). By the union bound, with probability at least $1 - q(|z|)/p(|z|) \geq 3/4$, it correctly answers all queries from $R(z)$, resulting in a description-restricted context-sensitive FAIN reduction from Π to inverting f.

Below, we fix z arbitrarily and drop the description "$, z$" from $M(\text{-}, z)$. Let n and s be the parameters of the GapLearn instance generated from z', so that $s = s_q(z')$ and $m_0(z, z') = \lceil c(s + \log|z|)\rceil$.

By the standard probabilistic argument, taking $N = O(\log|z|)$ independent sample sets $S^1, \ldots, S^N$, each of size $m_0(z, z')$, allows us to reduce the confidence error of learning from $1/4$ to $1/8p(|z|)$, i.e.,

$$\Pr_{S^1,\ldots,S^N,h}\left[\exists i \in [N] \text{ s.t. } \Pr_{(x,b)}\left[h^M(S^i, x, z) = b\right] \geq \frac{3}{4} + \frac{\gamma}{4}\right] \geq 1 - \frac{1}{8p(|z|)}.$$

The algorithm A is given access to samples drawn from $\mathcal{E}_{z,z'}$ and a size parameter 1^s, sets m to be $m_0(z, z') = \lceil c(s + \log|z|)\rceil$, and checks whether the following occurs: For each sample set $\bar{S} = (S^1, \ldots, S^N)$ and the randomness for h, let $E_{\bar{S},h}$ be the event that there exists $i \in [N]$ such that the empirical estimation of the probability $\Pr_{(x,b)}[h^M(S^i, x, z) = b]$ within accuracy error $\pm(1-\gamma)/16$ and negligible confidence error exceeds $(5 + 3\gamma)/8$. This trial is examined by using samples drawn from $\mathcal{E}_{z,z'}$. The algorithm A empirically estimates the probability that $E_{\bar{S},h}$ occurs within accuracy $\pm 1/(32p(|z|))$ with negligible confidence error. If the estimated probability is at least $1 - 3/(16p(|z|))$, the algorithm A outputs 1; otherwise, A outputs 0.

As in the proof of Claim 6.9, we can observe that if $s^*(z') \leq s$, the event above occurs with probability $1 - \mathsf{negl}(n)$ (over the randomness of A). Therefore, A outputs 1 with probability at least $1 - \mathsf{negl}(n)$ if the given instance is a Yes instance and selected by using the random seed z' satisfying Equation (23).

By contrast, suppose that for given $(\mathcal{E}_{z,z'}, 1^s)$, the algorithm A outputs 1 with probability at least $1/3$. We claim that

$$\Pr_{S^1,\ldots,S^N,h}\left[\exists i \in [N] \text{ s.t. } \Pr_{(x,b)}\left[h^M(S^i, x, z) = b\right] \geq \frac{1}{2} + \frac{\gamma}{2}\right] \geq 1 - \frac{1}{4p(|z|)}.$$

Otherwise, with probability at least $1/4p(|z|)$ over $\bar{S} = (S^1, \ldots, S^N)$ and h, the event $E_{\bar{S},h}$ occurs with negligible probability. Thus, $E_{\bar{S},h}$ occurs with probability at most $1 - 1/(4p(|z|)) + \mathsf{negl}(|z|) \ll 1 - 3/(16p(|z|)) - 1/(32p(|z|))$, and hence A outputs 1 only when the empirical estimation fails. The failure of the emprical estimation occurs only with negligible probability, which implies that A outputs 1 with negligible probability. This is a contradiction.

Let $\mathcal{Z}'$ be the set of z' such that z' satisfies the event in Equation (23) and A outputs 1 for $\mathcal{E}_{z,z'}$ with probability at least $1/3$. First, we consider the case of $\mathcal{Z}' \neq \emptyset$.

Then, we have

$$\Pr_{z'\sim\mathcal{Z}',S^1,\ldots,S^N,h}\left[\exists i \in [N] \text{ s.t. } \Pr_{(x,b)}\left[h^M(S^i, x, z) = b\right] \geq \frac{1}{2} + \frac{\gamma}{2}\right] \geq 1 - \frac{1}{4p(|z|)}. \tag{24}$$

Since each rejection-sampling trial succeeds with probability at least $3/4$, a standard tail bound for geometric random variables gives a universal constant $B_\perp$ such that, for every m and $P \geq 2$, if $T_1, \ldots, T_m$ are the stopping times of m independent rejection samples, then

$$\Pr\left[\sum_{j=1}^{m}\lceil\log(T_j + 1)\rceil \leq B_\perp(m + \log P) \wedge \forall j\, T_j \leq \lceil 2^{B_\perp(m+\log P)}\rceil\right] \geq 1 - \frac{1}{16P}.$$

The stopping indices can then be included in a standard tuple description using $O(\sum_j \log(T_j+1)+m)$ bits.

Let

$$n_{\max}(z) := \max_{z' \in \mathcal{Z}'} n(z,z'), \qquad m_{\max}(z) := \max_{z' \in \mathcal{Z}'} m_0(z,z'),$$

and put $L_{\max}(z) := 2^{2(n_{\max}(z)+1)}$ and $t_z = \lceil 2^{B_\perp(m_{\max}(z)+\log p(|z|))} \rceil$. A string w consists of a finite collection of random tapes $r^i_{j,t} \in \{0,1\}^{L_{\max}(z)}$ for $i \in [N]$, $j \in [m_{\max}(z)]$, and $t \in [t_z]$, together with one common random tape long enough for executing h^M in the maximum-size experiment. For a particular z', write

$$r^i_{j,t}[z'] := r^i_{j,t[2^{2(n(z,z')+1)}]}.$$

We define a test for w as $T^M_z(w) = 1$ if and only if with probability at least $1-1/2p(|z|)$ over $z' \sim \mathcal{Z}'$, there exist $i \in [N]$, stopping indices $t_1, \ldots, t_{m_0(z,z')} \in [t_z]$, and advice strings $a_1, \ldots, a_{m_0(z,z')} \in \{0,1\}^{\le \ell}$ such that

$$\sum_{j=1}^{m_0(z,z')} \lceil \log(t_j+1) \rceil \le B_\perp\big(m_0(z,z') + \log p(|z|)\big),$$

$U(a_j, r^i_{j,t_j}[z'], n(z,z')+1) \neq \perp$ for every $j \in [m_0(z,z')]$, and, for the sample set $S^i = \{(x^i_j, b^i_j)\}_{j \in [m_0(z,z')]}$ defined by

$$x^i_j \circ b^i_j = U(a_j, r^i_{j,t_j}[z'], n(z,z')+1),$$

it holds that

$$\Pr_{(x,b)}\big[h^M(S^i,x,z) = b\big] \ge \frac{1}{2} + \frac{\gamma}{2}.$$

Let G^M_z be a *good* seed set defined as $G^M_z = \{w : T^M_z(w) = 1\}$. To see that $G^M_z \neq \emptyset$, draw w uniformly and generate the sample sets in Equation (24) by the rejection sampler using the witnessing advice functions $\alpha_{z,z'}$. By parametric honesty and padding, the total statistical-distance loss over all samples is negligible in $|z|$; moreover, the stopping-index tail bound above fails with probability at most $1/(16p(|z|))$. Combining these facts with Equation (24) gives a positive probability of w satisfying the test T^M_z. Notice that G^M_z is recursively enumerable given (z, M): for each finite candidate w, all quantified domains in the definition of $T^M_z(w)$ are finite, M is total, and U is fixed. Thus, if w^* is the first element of G^M_z found by universal search, then (i) $\mathrm{K}(w^*) \le O(|z|+|M|)$, and (ii) with probability at least $1 - 1/2p(|z|)$ over $z' \sim \mathcal{Z}'$, it holds that there exist $i^* \in [N]$, stopping indices $t_1, \ldots, t_{m_0(z,z')}$, and advice strings $a_1, \ldots, a_{m_0(z,z')} \in \{0,1\}^{\le \ell}$ satisfying the stopping-index bound above and, for S^{i^*} reconstructed from w^*,

$$\Pr_{(x,b)}[h^M(S^{i^*},x,z) = b] \ge \frac{1}{2} + \frac{\gamma}{2}.$$

For such a fixed z', write $m = m_0(z,z')$. The program reconstructs S^{i^*} using the m selected rejection trials in row i^*; the value $n(z,z')$ is the input length of this learning instance and can also be hardwired with $O(\log|z|)$ bits. Writing $h^* := h^M(S^{i^*}, \text{-}, z)$, its description size is at most

$$\begin{aligned} |h^*| &\le \mathrm{K}(w^*) + \mathrm{K}(i^*) + \sum_{j=1}^{m} |a_j| + O\left(\sum_{j=1}^{m} \log(t_j+1) + m\right) + O(|z|) \\ &\le m \cdot \ell + O(B_\perp(m + \log p(|z|))) + O(|z| + |M|) \\ &\le C' \cdot s \cdot \ell, \end{aligned}$$

by selecting large enough λ and then taking ℓ sufficiently large; here $s \geq \lambda(|z|)$ by parametric honesty and the fixed machine M is absorbed into the polynomial choice of λ. Thus, these instances are not No instances for $\mathrm{Gap}^{\varepsilon(\ell),\gamma}_{C\cdot\ell}\mathrm{Learn}[\ell]$, for a sufficiently large constant C depending only on c and $B_\perp$. Therefore, A outputs 1 (with probability at least $1/3$) as a false positive for at most $1/2p(|z|)$-fraction of z' satisfying the event in Equation (23) for both cases of $\mathcal{Z}' \neq \emptyset$ and $\mathcal{Z}' = \emptyset$.

Thus, by the union bound, the error probability that A cannot output the correct answer (with probability at least $2/3$) is at most $1/4p(|z|) + 1/2p(|z|) \leq 1/p(|z|)$ over the choice of instances. □

6.5 One-Way Functions from Average-Case Hardness

In this section, we prove Theorems 1.5 and 1.10.

Theorem 6.13 (Theorem 1.10 Item 1)**.** *For every $\sigma \in \mathbb{N}$, let $\mathfrak{C}[\sigma]$ be the class of monotone functions that admit secret sharing of total size σ. For every constant $\gamma \in [0, 1/5)$, there exists $C \geq 1$ such that for every $\varepsilon(\sigma) = o(1)$ and every sufficiently large $\sigma \in \mathbb{N}$, if $\mathrm{Gap}^{\varepsilon(\sigma),\gamma}_{C\cdot\sigma}\mathfrak{C}[\sigma]$-CMMSA $\notin \mathsf{BPP}$, then $(\Pi, \mathcal{D}) \notin \mathsf{Avg}_{1/\mathsf{poly}}\mathsf{BPP}$ for some $\Pi \in \mathsf{NP}$ and samplable distribution $\mathcal{D}$.*

Proof. Let $\Pi := \mathrm{Gap}^{\varepsilon(\sigma),\gamma}_{C\cdot\sigma}\mathfrak{C}[\sigma]$-CMMSA. By the reduction from CMMSA to DMMSA and Lemma 6.3, there exist an auxiliary-input function $f = \{f_x\}$ and a description-restricted context-sensitive FAIN reduction R from Π to inverting f for some $C \geq 1$. Thus, $\Pi \notin \mathsf{BPP}$ implies the existence of an auxiliary-input one-way function, which in turn implies $(\Pi, \mathcal{D}) \notin \mathsf{Avg}_{1/\mathsf{poly}}\mathsf{BPP}$ for some $\Pi \in \mathsf{NP}$ and samplable distribution $\mathcal{D}$ by Proposition 3.3. □

Theorem 6.14 (Theorem 1.10 Item 2)**.** *For every $\sigma \in \mathbb{N}$, let $\mathfrak{C}[\sigma]$ be the class of monotone functions that admit secret sharing of total size σ. For every constant $\gamma \in [0, 1/5)$, there exists $C \geq 1$ such that for every $\varepsilon(\sigma) = o(1)$ and every sufficiently large $\sigma \in \mathbb{N}$, if $(\mathrm{Gap}^{\varepsilon(\sigma),\gamma}_{C\cdot\sigma}\mathfrak{C}[\sigma]$-CMMSA$, \mathcal{D}) \notin \mathsf{Avg}_{1/p}\mathsf{BPP}$ for some samplable distribution $\mathcal{D}$ and polynomial p, then there exists an infinitely-often one-way function.*

Proof. Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be an arbitrary samplable distribution, where $\mathcal{D}_n$ is a distribution over instances of instance size n for $\Pi = \mathrm{Gap}^{\varepsilon(\sigma),\gamma}_{C\cdot\sigma}\mathfrak{C}[\sigma]$-CMMSA. Let p be an arbitrary polynomial.

By the reduction from CMMSA to DMMSA and Lemma 6.3, there exist an auxiliary-input function $f = \{f_x\}$ and a description-restricted context-sensitive FAIN reduction R from Π to inverting f.

Now, we define a one-way function $g = \{g_n\}$ as follows: for each $n \in \mathbb{N}$,

$$g_n(r, r_f) := (x, f_x(r_f)),$$

where $x \sim \mathcal{D}_n$ sampled by using r, and r_f is a random seed for f_x. Since $\mathcal{D}$ is samplable, g is polynomial-time-computable function.

Suppose that there is no one-way function. By the standard amplification from weak one-way functions to strong one-way functions, for a negligible function $\nu(n)$, there exists a polynomial-time randomized algorithm M such that for every $n \in \mathbb{N}$,

$$\Pr_{r,r_f,\rho}\left[M(g_n(r, r_f); \rho) \in g_n^{-1}(g_n(r, r_f))\right] \geq 1 - \nu(n),$$

where ρ denotes the random tape of M. By Markov's inequality,

$$\Pr_{x\sim\mathcal{D}_n}\left[\Pr_{\rho,r_f}[M(x, f_x(r_f); \rho) \in \mathcal{D}_n^{-1}(x) \times f_x^{-1}(f_x(r_f))] \geq 1 - \sqrt{\nu(n)}\right] \geq 1 - \sqrt{\nu(n)}. \tag{25}$$

where $\mathcal{D}_n^{-1}(x)$ represents the seed set for the sampler of $\mathcal{D}_n$ such that x is sampled.

Let M' be the Turing machine that outputs the second element of M.

We construct the algorithm A that solves Π on average as follows: For a given $x \sim \mathcal{D}_n$, the algorithm A first empirically estimates the probability that $M'(x, f_x(r_f); \rho)$ inverts $f_x(r_f)$ over the choice of (ρ, r_f) within additive accuracy $1/100$ and confidence probability at least $1 - \mathsf{negl}(n)$. If the estimated probability is at least $49/50$, then A executes $R^{\mathcal{O}_{M'}(\text{-},x;\rho)}(x)$ independently a constant number of times and outputs the majority answer. Otherwise, A outputs $\perp$. Recall that since M' halts in polynomial time, A can simulate these executions in polynomial time.

Since the failure of the empirical estimation occurs only with negligible probability, we first ignore it. If x passes the empirical test, then

$$\Pr_{\rho, r_f}[M'(x, f_x(r_f); \rho) \in f_x^{-1}(f_x(r_f))] \geq \frac{49}{50} - \frac{1}{100} = \frac{97}{100}.$$

Let G_x be the set of random tapes ρ for which $M'(\text{-}, x; \rho)$ inverts f_x, i.e.,

$$\Pr_{r_f}[M'(x, f_x(r_f); \rho) \in f_x^{-1}(f_x(r_f))] \geq \frac{1}{2}.$$

The preceding inequality implies $\Pr_\rho[\rho \in G_x] \geq 47/50$. For every fixed $\rho \in G_x$, the deterministic machine obtained from M' by hardwiring ρ satisfies the fixed-randomness premise of the reduction definition; hence the reduction R outputs the correct answer with probability at least $3/4$ over its own randomness. Therefore one execution of $R^{\mathcal{O}_{M'}(\text{-},x;\rho)}(x)$ is correct with probability at least $(47/50) \cdot (3/4) > 2/3$, and the constant repetition above boosts this probability to at least $3/4$. Furthermore, by Equation (25) and the choice of negligible ν, at least with probability $1 - 1/p(n)$ over $x \sim \mathcal{D}_n$, the instance x passes the test.

Therefore, we conclude that for every $n \in \mathbb{N}$,

$$\Pr_{x \sim \mathcal{D}_n}\left[\Pr_A[A(x) = \Pi(x)] \geq 3/4\right] \geq 1 - \frac{1}{p(n)},$$

and for all $n \in \mathbb{N}$ and $x \in \mathsf{Support}(\mathcal{D}_n)$,

$$\Pr_A[A(x) \in \{\Pi(x), \perp\}] \geq 1 - \mathsf{negl}(n),$$

where the negligible terms from empirical estimation are absorbed by the confidence parameters and the final constant repetition.

Thus, $\Pi = \mathrm{Gap}^{\varepsilon(\sigma),\gamma}_{C\cdot\sigma}\mathfrak{C}[\sigma]\text{-CMMSA} \in \mathsf{Avg}_{1/p}\mathsf{BPP}$.

□

Next, we prove Theorem 1.5. The implication from the existence of one-way functions (Item 1) to the average-case hardness of GapLearn (Item 3) immediately follows from the well-known fact that a pseudorandom function is constructed from any one-way function [GGM86; HILL99] and it implies the average-case hardness of PAC learning under the uniform distribution even allowing any polynomially large hypothesis [Val84] (see also [HN23]). The implication from Item 3 to Item 2 is immediate because Item 3 requires $\sigma(n) \geq C\ell$, and an algorithm for the smaller gap $C\ell$ also solves every larger-gap promise problem. Thus, we will see only the remaining direction.

Theorem 6.15 (Theorem 1.5 Item 2 ⇒ Item 1)**.** *If there is no infinitely-often one-way function, then there exists a constant C such that for every $\varepsilon(\ell), \gamma(\ell) = o(1)$, every sufficiently large $\ell \in \mathbb{N}$, every samplable distribution $\mathscr{D} = \{\mathscr{D}_n\}_n$ over instances of $\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{C\cdot\ell}\mathrm{Learn}[\ell]$, and every polynomial p, there exists a randomized algorithm that solves $\mathrm{Gap}^{\varepsilon(\ell),\gamma(\ell)}_{C\cdot\ell}\mathrm{Learn}[\ell]$ on average with probability at least $1 - 1/p(n)$ over $\mathscr{D}_n$ for every $n \in \mathbb{N}$.*

Proof. Fix ℓ and write $\varepsilon := \varepsilon(\ell)$ and $\gamma := \gamma(\ell)$. Let $\mathscr{D} = \{\mathscr{D}_n\}_{n\in\mathbb{N}}$ be an arbitrary samplable distribution as in the statement and p be an arbitrary polynomial. We define $\mathcal{D} = \{\mathcal{D}_{n,r}\}$ as follows: for each $n \in \mathbb{N}$ and random seed r for sampling from $\mathscr{D}_n$, the seed r determines an instance $(1^n, 1^{s_q(r)}, \mathcal{E})$, and $\mathcal{D}_{n,r}$ is the corresponding distribution over samples in $\{0,1\}^n \times \{0,1\}$. Recall that $\mathcal{D}_{n,r}$ is samplable with advice complexity ℓ. As in the previous subsection, reconstructed samples are obtained by rejection sampling from U on the canonical $(n+1)$-bit encoding, discarding $\perp$ outcomes; their conditional distribution is within statistical distance $2^{-2(n+1)}$ of $\mathcal{D}_{n,r}$.

Let $c \geq 1$ be the sample-complexity constant obtained from Lemma 6.7, and define the polynomial-time computable polynomially bounded sample-size function

$$m_0(n,r) := \lceil c \cdot (s_q(r) + \log n) \rceil .$$

Applying Lemmas 6.5 and 6.7 to this sample-size function, we obtain a randomized oracle machine h and a polynomial-time computable function $f = \{f_n\}$ such that for every large enough $n \in \mathbb{N}$ and every oracle $\mathcal{I}$ that inverts f_n,

$$\Pr_r\left[m_0(n,r) < c \cdot (s^*(r) + \log n) \text{ or } \Pr_{S,h}\left[\Pr_{(x,b)}\left[h^{\mathcal{I}}(S,x,1^n) = b\right] \geq \frac{3}{4} + \frac{\gamma}{4}\right] \geq \frac{3}{4}\right] \geq 1 - \frac{1}{4p(n)},$$

where S consists of $m_0(n,r)$ independent samples from $\mathcal{D}_{n,r}$, $(x,b) \sim \mathcal{D}_{n,r}$, and $s^*(r)$ is the minimum size polynomial-time program h such that

$$\Pr_{(x,b)\sim\mathcal{D}_{n,r}}[h(x) = b] \geq 1 - \varepsilon.$$

Suppose that there is no one-way function. Again using the standard weak-to-strong amplification of one-way functions, for a negligible function $\nu(n)$, there exists a polynomial-time randomized algorithm M such that for every $n \in \mathbb{N}$,

$$\Pr_{\rho,u}[M(1^n, f_n(u); \rho) \in f_n^{-1}(f_n(u))] \geq 1 - \nu(n),$$

where ρ is the random tape of M. Let G_n be the set of tapes ρ for which $M(\text{-};\rho)$ inverts f_n with probability at least $1/2$ over u. Then $\Pr_\rho[\rho \in G_n] \geq 1 - 2\nu(n)$. Applying the preceding guarantee with the deterministic oracle $\mathcal{I} = M(\text{-};\rho)$ for each $\rho \in G_n$ and then averaging over ρ (viewing this tape as part of the randomness of h^M), and absorbing the negligible loss by the choice of ν, gives

$$\Pr_r\left[m_0(n,r) < c \cdot (s^*(r) + \log n) \text{ or } \Pr_{S,h}\left[\Pr_{(x,b)}\left[h^M(S,x,1^n) = b\right] \geq \frac{3}{4} + \frac{\gamma}{4}\right] \geq \frac{3}{4}\right] \geq 1 - \frac{1}{4p(n)}, \quad (26)$$

where S has size $m_0(n,r)$.

Below, we consider only such values of r and present an algorithm A that performs correctly for these values of r.

By the standard probabilistic argument, taking $N = O(\log n)$ independent sample sets $S^1, \ldots, S^N$, each of size $m_0(n,r)$, allows us to reduce the confidence error of learning from $1/4$ to $1/8p(n)$, i.e.,

$$\Pr_{S^1,\ldots,S^N,h}\left[\exists i \in [N] \text{ s.t. } \Pr_{(x,b)}\left[h^M(S^i,x,1^n) = b\right] \geq \frac{3}{4} + \frac{\gamma}{4}\right] \geq 1 - \frac{1}{8p(n)}.$$

The algorithm A is given access to samples drawn from $\mathcal{D}_{n,r}$ and a size parameter 1^s, sets m to be $m_0(n,r) = \lceil c(s + \log n)\rceil$, and checks whether the following occurs: For each sample set $\bar{S} = (S^1, \ldots, S^N)$ and the randomness for h, let $E_{\bar{S},h}$ be the event that there exists $i \in [N]$ such

that the empirical estimation of the probability $\Pr_{(x,b)}[h^M(S^i, x, 1^n) = b]$ within accuracy error $\pm(1-\gamma)/16$ and negligible confidence error exceeds $(5+3\gamma)/8$. This trial is examined by using samples drawn from $\mathcal{D}_{n,r}$. The algorithm A empirically estimates the probability that $E_{\bar{S},h}$ occurs within accuracy $\pm 1/(32p(n))$ with negligible confidence error. If the estimated probability is at least $1 - 3/(16p(n))$, the algorithm A outputs 1; otherwise, A outputs 0.

As in the proof of Claim 6.9, we can observe that if $s^*(r) \leq s$, the event above occurs with probability $1 - \mathsf{negl}(n)$ (over the randomness of A). Therefore, A outputs 1 with probability at least $1 - \mathsf{negl}(n)$ if the given instance is a Yes instance and selected by using the random seed r satisfying Equation (26).

By contrast, suppose that for given $(\mathcal{D}_{n,r}, 1^s)$, the algorithm A outputs 1 with probability at least 1/3. We claim that

$$\Pr_{S^1,\ldots,S^N,h}\left[\exists i \in [N] \text{ s.t. } \Pr_{(x,b)}\left[h^M(S^i, x, 1^n) = b\right] \geq \frac{1}{2} + \frac{\gamma}{2}\right] \geq 1 - \frac{1}{4p(n)}.$$

Otherwise, with probability at least $1/4p(n)$ over $\bar{S} = (S^1, \ldots, S^N)$ and h, the event $E_{\bar{S},h}$ occurs with negligible probability. Thus, $E_{\bar{S},h}$ occurs with probability at most $1 - 1/(4p(n)) + \mathsf{negl}(n) \ll 1 - 3/(16p(n)) - 1/(32p(n))$, and hence A outputs 1 only when the empirical estimation fails. The failure of the empirical estimation occurs only with negligible probability, which implies that A outputs 1 with negligible probability. This is a contradiction.

Let $\mathcal{R}$ be the set of r such that r satisfies the event in Equation (26) and A outputs 1 for $\mathcal{D}_{n,r}$ with probability at least 1/3. First, we consider the case where $\mathcal{R} \neq \emptyset$.

Then, we have

$$\Pr_{r\sim\mathcal{R},S^1,\ldots,S^N,h}\left[\exists i \in [N] \text{ s.t. } \Pr_{(x,b)}\left[h^M(S^i, x, 1^n) = b\right] \geq \frac{1}{2} + \frac{\gamma}{2}\right] \geq 1 - \frac{1}{4p(n)}. \tag{27}$$

The same stopping-index tail bound gives, for every m and $P \geq 2$,

$$\Pr\left[\sum_{j=1}^{m} \lceil \log(T_j + 1) \rceil \leq B_\perp(m + \log P) \wedge \forall j\, T_j \leq \lceil 2^{B_\perp(m+\log P)} \rceil\right] \geq 1 - \frac{1}{16P},$$

where $T_1, \ldots, T_m$ are the stopping times of m independent rejection samples. As above, these indices can be encoded as a tuple using $O(\sum_j \log(T_j + 1) + m)$ bits.

Let $m_{\max}(n) := \max_{r\in\mathcal{R}} m_0(n, r)$ and $t_n = \lceil 2^{B_\perp(m_{\max}(n)+\log p(n))} \rceil$. A string w consists of a finite collection of random seeds $r^i_{j,t} \in \{0,1\}^{2^{2(n+1)}}$ for $i \in [N]$, $j \in [m_{\max}(n)]$, and $t \in [t_n]$, together with one common random tape long enough for executing h^M in the maximum-size experiment. We define a test for w as $T^M_n(w) = 1$ if and only if with probability at least $1 - 1/2p(n)$ over $r \sim \mathcal{R}$, there exist $i \in [N]$, stopping indices $t_1, \ldots, t_{m_0(n,r)} \in [t_n]$, and advice strings $a_1, \ldots, a_{m_0(n,r)} \in \{0,1\}^{\leq \ell}$ such that

$$\sum_{j=1}^{m_0(n,r)} \lceil \log(t_j + 1) \rceil \leq B_\perp\big(m_0(n, r) + \log p(n)\big),$$

$U(a_j, r^i_{j,t_j}, n+1) \neq \perp$ for every $j \in [m_0(n,r)]$ and, for the sample set $S^i = \{(x^i_j, b^i_j)\}_{j\in[m_0(n,r)]}$ defined by

$$x^i_j \circ b^i_j = U(a_j, r^i_{j,t_j}, n+1),$$

it holds that

$$\Pr_{(x,b)}\left[h^M(S^i, x, 1^n) = b\right] \geq \frac{1}{2} + \frac{\gamma}{2}.$$

Let G_n^M be a *good* seed set defined as $G_n^M = \{w : T_n^M(w) = 1\}$. To see that $G_n^M \neq \emptyset$, draw w uniformly and generate the sample sets in Equation (27) by the rejection sampler using the witnessing advice functions for the corresponding distributions $\mathcal{D}_{n,r}$. Their joint distribution is within statistical distance at most $Nm_{\max}(n)2^{-2(n+1)} = \mathsf{negl}(n)$ of the product of the target distributions; moreover, the stopping-index tail bound above fails with probability at most $1/(16p(n))$. Combining these facts with Equation (27) gives a positive probability of w satisfying the test T_n^M. Notice that G_n^M is recursively enumerable given $(1^n, M)$: for each finite candidate w, all quantified domains in the definition of $T_n^M(w)$ are finite, M is total, and U is fixed. Thus, if w^* is the first element of G_n^M found by universal search, then (i) $\mathrm{K}(w^*) \leq O(\log n + |M|)$, and (ii) with probability at least $1 - 1/2p(n)$ over $r \sim \mathcal{R}$, it holds that there exist $i^* \in [N]$, stopping indices $t_1, \ldots, t_{m_0(n,r)}$, and advice strings $a_1, \ldots, a_{m_0(n,r)} \in \{0,1\}^{\leq \ell}$ satisfying the stopping-index bound above and, for S^{i^*} reconstructed from w^* using the corresponding rejection trials and advice strings,

$$\Pr_{(x,b)}[h^M(S^{i^*}, x, 1^n) = b] \geq \frac{1}{2} + \frac{\gamma}{2}.$$

For such a fixed r, write $m = m_0(n, r)$. Writing $h^* := h^M(S^{i^*}, \text{-}, 1^n)$, its description size is at most

$$\begin{aligned} |h^*| &\leq \mathrm{K}(w^*) + \mathrm{K}(i^*) + \sum_{j=1}^{m} |a_j| + O\left(\sum_{j=1}^{m} \log(t_j + 1) + m\right) + O(\log n) \\ &\leq m \cdot \ell + O(B_\perp(m + \log p(n))) + O(\log n + |M|) \\ &\leq C' \cdot s \cdot \ell, \end{aligned}$$

where the last inequality holds by assuming $\log n \leq s$ and taking n large enough, ℓ sufficiently large, and C' large enough to absorb the fixed machine M and the stopping-index cost; otherwise, we can efficiently find the hypothesis of size s by brute-force search. Thus, these instances are not No instances for $\mathrm{Gap}^{\varepsilon,\gamma}_{C \cdot \ell}\mathrm{Learn}[\ell]$, for a sufficiently large constant C depending only on c and $B_\perp$. Therefore, A outputs 1 (with probability at least $1/3$) as a false positive for at most $1/2p(n)$-fraction of r satisfying the event in Equation (26) in both cases of $\mathcal{R} \neq \emptyset$ and $\mathcal{R} = \emptyset$.

Thus, by the union bound, the error probability that A cannot output the correct answer (with probability at least $2/3$) is at most $1/4p(n) + 1/2p(n) \leq 1/p(n)$ over the choice of instances. □

7 Open Problems

We list a couple of open problems.

- Can we explain the difficulty of matching the inapproximability factors in Corollary 1.4 and Theorem 1.5? For example, is there an oracle under which the factor in Theorem 1.5 cannot be improved further?
- Can we establish sharp thresholds to rule out *Heuristica* for agnostic learning? Previous work [HN21] is insufficient, as it relies on agnostic boosting [Fel10; KK09], which causes a cubic blowup in hypothesis size when boosting accuracy to a constant.
- Can we prove Theorems 1.3 and 1.8 in the realizable case where $\varepsilon = 0$?
- Can we prove Theorems 1.3 and 1.8 for a superconstant function $\ell = \omega(1)$?

- Can we establish a similar sharp threshold result for other related problems, such as MINKT* and MCSP*, by using or extending the techniques developed in this paper? Here, MINKT* (resp. MCSP*) is the problem of, given a partial string (resp. a partial truth table), determining the size of the minimum program (resp. minimum circuit) that produces a string consistent with the partial string.

Acknowledgment

Shuichi Hirahara was supported by JST, FOREST Grant Number JPMJFR226Y. Mikito Nanashima was supported by JST, ACT-X Grant Number JPMJAX24CJ. We thank an anonymous reviewer for pointing out that the Occam bound is not necessary for our proof, which simplified the argument.

References

[ABBGKLPSV23] Divesh Aggarwal, Huck Bennett, Zvika Brakerski, Alexander Golovnev, Rajendra Kumar, Zeyong Li, Spencer Peters, Noah Stephens-Davidowitz, and Vinod Vaikuntanathan. "Lattice Problems beyond Polynomial Time". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2023, pp. 1516–1526. DOI: 10.1145/3564246.3585227.

[ABFKP08] Michael Alekhnovich, Mark Braverman, Vitaly Feldman, Adam R. Klivans, and Toniann Pitassi. "The complexity of properly learning simple concept classes". In: *J. Comput. Syst. Sci.* 74.1 (2008), pp. 16–34. DOI: 10.1016/j.jcss.2007.04.011.

[ABMP01] Michael Alekhnovich, Samuel R. Buss, Shlomo Moran, and Toniann Pitassi. "Minimum Propositional Proof Length Is NP-Hard to Linearly Approximate". In: *J. Symb. Log.* 66.1 (2001), pp. 171–191. DOI: 10.2307/2694916.

[ABX08] Benny Applebaum, Boaz Barak, and David Xiao. "On Basing Lower-Bounds for Learning on Worst-Case Assumptions". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2008, pp. 211–220. DOI: 10.1109/FOCS.2008.35.

[ACMTV21] Eric Allender, Mahdi Cheraghchi, Dimitrios Myrisiotis, Harsha Tirumala, and Ilya Volkovich. "One-Way Functions and a Conditional Variant of MKTP". In: *Proceedings of the Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*. 2021, 7:1–7:19. DOI: 10.4230/LIPIcs.FSTTCS.2021.7.

[AGGM06] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. "On basing one-way functions on NP-hardness". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2006, pp. 701–710. DOI: 10.1145/1132516.1132614.

[AN21] Benny Applebaum and Oded Nir. "Upslices, Downslices, and Secret-Sharing with Complexity of 1.5^n". In: *Proceedings of the International Cryptology Conference (CRYPTO)*. 2021, pp. 627–655. DOI: 10.1007/978-3-030-84252-9_21.

[AR08] Michael Alekhnovich and Alexander A. Razborov. "Resolution Is Not Automatizable Unless W[P] Is Tractable". In: *SIAM J. Comput.* 38.4 (2008), pp. 1347–1363. DOI: 10.1137/06066850X.

[BB15] Andrej Bogdanov and Christina Brzuska. "On Basing Size-Verifiable One-Way Functions on NP-Hardness". In: *Proceedings of the Theory of Cryptography Conference (TCC)*. 2015, pp. 1–6. DOI: 10.1007/978-3-662-46494-6_1.

[Ben23] Huck Bennett. "The Complexity of the Shortest Vector Problem". In: *SIGACT News* 54.1 (2023), pp. 37–61. DOI: 10.1145/3586165.3586172.

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. "Cryptographic Primitives Based on Hard Learning Problems". In: *Proceedings of the International Cryptology Conference (CRYPTO)*. 1993, pp. 278–291. DOI: 10.1007/3-540-48329-2_24.

[BL88] Josh Cohen Benaloh and Jerry Leichter. "Generalized Secret Sharing and Monotone Functions". In: *Proceedings of the International Cryptology Conference (CRYPTO)*. 1988, pp. 27–35. DOI: 10.1007/0-387-34799-2_3.

[BT06a] Andrej Bogdanov and Luca Trevisan. "Average-Case Complexity". In: *Foundations and Trends in Theoretical Computer Science* 2.1 (2006). DOI: 10.1561/0400000004.

[BT06b] Andrej Bogdanov and Luca Trevisan. "On Worst-Case to Average-Case Reductions for NP Problems". In: *SIAM J. Comput.* 36.4 (2006), pp. 1119–1159. DOI: 10.1137/S0097539705446974.

[CDKM00] Robert D. Carr, Srinivas Doddi, Goran Konjevod, and Madhav V. Marathe. "On the red-blue set cover problem". In: *Proceedings of the Symposium on Discrete Algorithms (SODA)*. 2000, pp. 345–353.

[CJW20] Lijie Chen, Ce Jin, and R. Ryan Williams. "Sharp threshold results for computational complexity". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2020, pp. 1335–1348. DOI: 10.1145/3357713.3384283.

[CNW16] Moses Charikar, Yonatan Naamad, and Anthony Wirth. "On Approximating Target Set Selection". In: *Proceedings of the Approximation, Randomization, and Combinatorial Optimization (APPROX/RANDOM)*. 2016, 4:1–4:16. DOI: 10.4230/LIPICS.APPROX-RANDOM.2016.4.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)* Wiley, 2006. ISBN: 978-0-471-24195-9.

[Dan16] Amit Daniely. "Complexity theoretic limitations on learning halfspaces". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2016, pp. 105–117. DOI: 10.1145/2897518.2897520.

[DFKRS11] Irit Dinur, Eldar Fischer, Guy Kindler, Ran Raz, and Shmuel Safra. "PCP Characterizations of NP: Toward a Polynomially-Small Error-Probability". In: *Comput. Complex.* 20.3 (2011), pp. 413–504. DOI: 10.1007/s00037-011-0014-4.

[DHK15] Irit Dinur, Prahladh Harsha, and Guy Kindler. "Polynomially Low Error PCPs with polyloglog n Queries via Modular Composition". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2015, pp. 267–276. DOI: 10.1145/2746539.2746630.

[DLS14] Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. "From average case complexity to improper learning complexity". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2014, pp. 441–448. DOI: 10.1145/2591796.2591820.

[DS04] Irit Dinur and Shmuel Safra. "On the hardness of approximating label-cover". In: *Inf. Process. Lett.* 89.5 (2004), pp. 247–254. DOI: 10.1016/j.ipl.2003.11.007.

[DS16] Amit Daniely and Shai Shalev-Shwartz. "Complexity Theoretic Limitations on Learning DNF's". In: *Proceedings of the Conference on Learning Theory (COLT)*. 2016, pp. 815–830.

[DV21] Amit Daniely and Gal Vardi. "From Local Pseudorandom Generators to Hardness of Learning". In: *Proceedings of the Conference on Learning Theory (COLT)*. 2021, pp. 1358–1394.

[Fel10] Vitaly Feldman. "Distribution-Specific Agnostic Boosting". In: *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*. 2010, pp. 241–250.

[FF93] Joan Feigenbaum and Lance Fortnow. "Random-Self-Reducibility of Complete Sets". In: *SIAM J. Comput.* 22.5 (1993), pp. 994–1005. DOI: 10.1137/0222061.

[FS12] Yoav Freund and Robert E Schapire. "Boosting: Foundations and Algorithms. Adaptive computation and machine learning". In: *MIT Press* 2 (2012), p. 8.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to construct random functions". In: *J. ACM* 33.4 (1986), pp. 792–807. DOI: 10.1145/6490.6503.

[GK23] Halley Goldberg and Valentine Kabanets. "Improved Learning from Kolmogorov Complexity". In: *Proceedings of the Computational Complexity Conference (CCC)*. 2023, 12:1–12:29. DOI: 10.4230/LIPICS.CCC.2023.12.

[GL22] Suprovat Ghoshal and Euiwoong Lee. "A characterization of approximability for biased CSPs". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2022, pp. 989–997. DOI: 10.1145/3519935.3520072.

[GM84] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption". In: *J. Comput. Syst. Sci.* 28.2 (1984), pp. 270–299. DOI: 10.1016/0022-0000(84)90070-9.

[GM97] Michael H. Goldwasser and Rajeev Motwani. "Intractability of Assembly Sequencing: Unit Disks in the Plane". In: *Algorithms and Data Structures, 5th International Workshop, WADS '97, Halifax, Nova Scotia, Canada, August 6-8, 1997, Proceedings*. 1997, pp. 307–320. DOI: 10.1007/3-540-63307-3_70.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. "Proofs that Yield Nothing But Their Validity for All Languages in NP Have Zero-Knowledge Proof Systems". In: *J. ACM* 38.3 (1991), pp. 691–729. DOI: 10.1145/116825.116852.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "A Pseudorandom Generator from any One-way Function". In: *SIAM J. Comput.* 28.4 (1999), pp. 1364–1396. DOI: 10.1137/S0097539793244708.

[HILNO23] Shuichi Hirahara, Rahul Ilango, Zhenjian Lu, Mikito Nanashima, and Igor C. Oliveira. "A Duality between One-Way Functions and Average-Case Symmetry of Information". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2023, pp. 1039–1050. DOI: 10.1145/3564246.3585138.

[Hir22] Shuichi Hirahara. "NP-Hardness of Learning Programs and Partial MCSP". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2022, pp. 968–979. DOI: 10.1109/FOCS54457.2022.00095.

[Hir23] Shuichi Hirahara. "Capturing One-Way Functions via NP-Hardness of Meta-Complexity". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2023, pp. 1027–1038. DOI: 10.1145/3564246.3585130.

[HLN24] Shuichi Hirahara, Zhenjian Lu, and Mikito Nanashima. "Optimal Coding for Randomized Kolmogorov Complexity and Its Applications". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2024, pp. 369–378. DOI: 10.1109/FOCS61266.2024.00030.

[HN21] Shuichi Hirahara and Mikito Nanashima. "On Worst-Case Learning in Relativized Heuristica". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2021, pp. 751–758. DOI: 10.1109/FOCS52979.2021.00078.

[HN23] Shuichi Hirahara and Mikito Nanashima. "Learning in Pessiland via Inductive Inference". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2023, pp. 447–457. DOI: 10.1109/FOCS57990.2023.00033.

[HN24] Shuichi Hirahara and Mikito Nanashima. "One-Way Functions and Zero Knowledge". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2024, pp. 1731–1738. DOI: 10.1145/3618260.3649701.

[HS17] Shuichi Hirahara and Rahul Santhanam. "On the Average-Case Complexity of MCSP and Its Variants". In: *Proceedings of the Computational Complexity Conference (CCC)*. 2017, 7:1–7:20. DOI: 10.4230/LIPIcs.CCC.2017.7.

[IL89] Russell Impagliazzo and Michael Luby. "One-way Functions are Essential for Complexity Based Cryptography (Extended Abstract)". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 1989, pp. 230–235. DOI: 10.1109/SFCS.1989.63483.

[IL90] Russell Impagliazzo and Leonid A. Levin. "No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 1990, pp. 812–821. DOI: 10.1109/FSCS.1990.89604.

[Ila23] Rahul Ilango. "SAT Reduces to the Minimum Circuit Size Problem with a Random Oracle". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2023, pp. 733–742. DOI: 10.1109/FOCS57990.2023.00048.

[Imp11] Russell Impagliazzo. "Relativized Separations of Worst-Case and Average-Case Complexities for NP". In: *Proceedings of the Conference on Computational Complexity (CCC)*. 2011, pp. 104–114. DOI: 10.1109/CCC.2011.34.

[Imp95] Russell Impagliazzo. "A Personal View of Average-Case Complexity". In: *Proceedings of the Structure in Complexity Theory Conference*. 1995, pp. 134–147. DOI: 10.1109/SCT.1995.514853.

[IRS22] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. "Robustness of average-case meta-complexity via pseudorandomness". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2022, pp. 1575–1583. DOI: 10.1145/3519935.3520051.

[ISN93] Mitsuru Ito, Akira Saito, and Takao Nishizeki. "Multiple Assignment Scheme for Sharing Secret". In: *J. Cryptol.* 6.1 (1993), pp. 15–20. DOI: 10.1007/BF02620229.

[KK09] Adam Kalai and Varun Kanade. "Potential-Based Agnostic Boosting". In: *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*. 2009, pp. 880–888.

[Ko91] Ker-I Ko. "On the Complexity of Learning Minimum Time-Bounded Turing Machines". In: *SIAM J. Comput.* 20.5 (1991), pp. 962–986. DOI: 10.1137/0220059.

[Lee06] Troy Lee. "Kolmogorov Complexity and Formula Size Lower Bounds". PhD thesis. University of Amsterdam, 2006.

[Liv10] Noam Livne. "On the Construction of One-Way Functions from Average Case Hardness". In: *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*. 2010, pp. 301–309.

[LP20] Yanyi Liu and Rafael Pass. "On One-way Functions and Kolmogorov Complexity". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2020, pp. 1243–1254. DOI: 10.1109/FOCS46700.2020.00118.

[LP21] Yanyi Liu and Rafael Pass. "On the Possibility of Basing Cryptography on EXP$\neq$ BPP". In: *Proceedings of the International Cryptology Conference (CRYPTO)*. 2021, pp. 11–40. DOI: 10.1007/978-3-030-84242-0_2.

[LP22] Yanyi Liu and Rafael Pass. "On One-Way Functions from NP-Complete Problems". In: *Proceedings of the Computational Complexity Conference (CCC)*. 2022, 36:1–36:24. DOI: 10.4230/LIPIcs.CCC.2022.36.

[LP23a] Yanyi Liu and Rafael Pass. "On One-Way Functions and Sparse Languages". In: *Proceedings of the Theory of Cryptography Conference (TCC)*. 2023, pp. 219–237. DOI: 10.1007/978-3-031-48615-9_8.

[LP23b] Yanyi Liu and Rafael Pass. "One-Way Functions and the Hardness of (Probabilistic) Time-Bounded Kolmogorov Complexity w.r.t. Samplable Distributions". In: *Proceedings of the International Cryptology Conference (CRYPTO)*. 2023, pp. 645–673. DOI: 10.1007/978-3-031-38545-2_21.

[LS24] Zhenjian Lu and Rahul Santhanam. "Impagliazzo's Worlds Through the Lens of Conditional Kolmogorov Complexity". In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. 2024, 110:1–110:17. DOI: 10.4230/LIPICS.ICALP.2024.110.

[LV19] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 4th Edition*. Texts in Computer Science. Springer, 2019. ISBN: 978-3-030-11297-4. DOI: 10.1007/978-3-030-11298-1.

[MZ26] Dor Minzer and Kai Zhe Zheng. "Near Optimal Hardness of Approximating k-CSP". In: *Proceedings of the Symposium on Theory of Computing (STOC)* (2026).

[Nan21] Mikito Nanashima. "On Basing Auxiliary-Input Cryptography on NP-Hardness via Nonadaptive Black-Box Reductions". In: *Proceedings of the Innovations in Theoretical Computer Science Conference (ITCS)*. 2021, 29:1–29:15. DOI: 10.4230/LIPIcs.ITCS.2021.29.

[Nao91] Moni Naor. "Bit Commitment Using Pseudorandomness". In: *J. Cryptol.* 4.2 (1991), pp. 151–158. DOI: 10.1007/BF00196774.

[NOV06] Minh-Huyen Nguyen, Shien Jin Ong, and Salil P. Vadhan. "Statistical Zero-Knowledge Arguments for NP from Any One-Way Function". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2006, pp. 3–14. DOI: 10.1109/FOCS.2006.71.

[NR06] Moni Naor and Guy N. Rothblum. "Learning to impersonate". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2006, pp. 649–656. DOI: 10.1145/1143844.1143926.

[OW93] Rafail Ostrovsky and Avi Wigderson. "One-Way Fuctions are Essential for Non-Trivial Zero-Knowledge". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 1993, pp. 3–17. DOI: 10.1109/ISTCS.1993.253489.

[Pei16] Chris Peikert. "A Decade of Lattice Cryptography". In: *Found. Trends Theor. Comput. Sci.* 10.4 (2016), pp. 283–424. DOI: 10.1561/0400000074.

[RS21] Hanlin Ren and Rahul Santhanam. "Hardness of KT Characterizes Parallel Cryptography". In: *Proceedings of the Computational Complexity Conference (CCC)*. 2021, 35:1–35:58. DOI: 10.4230/LIPIcs.CCC.2021.35.

[Sch90] Robert E. Schapire. "The Strength of Weak Learnability". In: *Mach. Learn.* 5 (1990), pp. 197–227. DOI: 10.1007/BF00116037.

[Sol64a] Ray J. Solomonoff. "A Formal Theory of Inductive Inference. Part I". In: *Inf. Control.* 7.1 (1964), pp. 1–22. DOI: 10.1016/S0019-9958(64)90223-2.

[Sol64b] Ray J. Solomonoff. "A Formal Theory of Inductive Inference. Part II". In: *Inf. Control.* 7.2 (1964), pp. 224–254. DOI: 10.1016/S0019-9958(64)90131-7.

[TV07] Luca Trevisan and Salil P. Vadhan. "Pseudorandomness and Average-Case Complexity Via Uniform Reductions". In: *Computational Complexity* 16.4 (2007), pp. 331–364. DOI: 10.1007/s00037-007-0233-x.

[Vad17] Salil P. Vadhan. "On Learning vs. Refutation". In: *Proceedings of the Conference on Learning Theory (COLT)*. 2017, pp. 1835–1848.

[Val84] Leslie G. Valiant. "A Theory of the Learnable". In: *Commun. ACM* 27.11 (1984), pp. 1134–1142. DOI: 10.1145/1968.1972.

[Vio05] Emanuele Viola. "On Constructing Parallel Pseudorandom Generators from One-Way Functions". In: *Proceedings of the Conference on Computational Complexity (CCC)*. 2005, pp. 183–197. DOI: 10.1109/CCC.2005.16.

[Wat12] Thomas Watson. "Relativized Worlds without Worst-Case to Average-Case Reductions for NP". In: *TOCT* 4.3 (2012), 8:1–8:30. DOI: 10.1145/2355580.2355583.

[Wee06] Hoeteck Wee. "Finding Pessiland". In: *Proceedings of the Theory of Cryptography Conference (TCC)*. 2006, pp. 429–442. DOI: 10.1007/11681878_22.

A An Approximation Algorithm for DNF-MMSA

We present an elementary approximation algorithm for DNF-MMSA based on LP relaxation.

Theorem A.1. *Given a collection of monotone DNF formulas of size ℓ (i.e., the number of terms) and a weight function w for variables such that there exists an assignment of weight at most s^* and satisfies at least $(1-\epsilon)$-fraction of the monotone DNF formulas, it is feasible in polynomial time to find a satisfying assignment of weight at most $\ell \cdot s^*/(1-\eta)$ that satisfies at least $(1-\epsilon/\eta)$-fraction of the monotone DNF formulas for any parameter $\eta > 0$.*

Moreover, when all the DNF constraints are satisfiable by an assignment of weight s^* (i.e., $\epsilon = 0$), the algorithm finds an assignment of weight at most $\ell \cdot s^*$ that satisfies all the constraints.

Proof. First, we consider the case in which $\epsilon > 0$. Let $x_1, \ldots, x_n$ be the variables for the given collection of DNFs and $w\colon [n] \to [0,1]$ be the weight function, i.e., $\sum_i w(i) = 1$. Let m be the number of DNFs. We introduce new variables $z_1, z_2, \ldots, z_m$ for each DNF in the collection and $y_{i,1}, y_{i,2}, \ldots, y_{i,j_i}$ for each term in the i-th DNF (i.e., $j_i \le \ell$).

We represent the MMSA instance as the following integer programming (IP):

$$
\begin{aligned}
\textbf{min} \quad & \sum_{i \in [n]} w(i) x_i \\
\textbf{s.t.} \quad & \sum_{i \in [m]} z_i \ge (1-\epsilon) m \\
& \sum_{j \in [j_i]} y_{i,j} \ge z_i \quad \forall i \in [m] \\
& x_k \ge y_{i,j} \quad \forall (i,j) \text{ and a variable } x_k \text{ relevant to the } j\text{-th term in the } i\text{-th DNF} \\
& x_k, y_{i,j}, z_i \in \{0,1\} \quad \forall i, j, k
\end{aligned}
$$

it is not hard to see that the optimal value v_{IP} of the above IP is at most s^*. Now we relax the range $\{0,1\}$ to $[0,1]$ and solve the resulting linear programming (LP) in polynomial time. Let $\alpha \in [0,1]^n$ be the resulting assignment to $x_1, \ldots, x_n$. Then the value $w(\alpha) = \sum w(i)\alpha_i$ is at most $v_{IP} \le s^*$.

We round α to a binary assignment $\tilde{\alpha} \in \{0,1\}^n$ as follows: for each $i \in [n]$,

$$
\tilde{\alpha}_i = \begin{cases} 1 & \text{if } (\ell/(1-\eta)) \cdot \alpha_i \ge 1 \\ 0 & \text{otherwise.} \end{cases}
$$

Then, $w(\tilde{\alpha}) \le (\ell/(1-\eta)) \cdot w(\alpha) \le (\ell/(1-\eta)) s^*$. Thus, it suffices to observe that $\tilde{\alpha}$ satisfies at least $(1-\epsilon/\eta)$-fraction of the monotone DNF formulas.

We consider the values assigned to $y_{i,j}$'s and z_i's. Since $\mathbb{E}_i[z_i] \ge 1-\epsilon$, by Markov's inequality, at least $(1-\epsilon/\eta)$-fraction of z_i takes the value at least $1-\eta$. For such i, there must exist $y_{i,j} \ge (1-\eta)/\ell$ since $y_{i,j} \ge 0$. For such (i,j), the relevant variable x_k must take the value at least $(1-\eta)/\ell$, which must be assigned to 1 in $\tilde{\alpha}$. Thus, $\tilde{\alpha}$ satisfies these at least $(1-\epsilon/\eta)$-fraction of the monotone DNF formulas.

In the satisfiable case where $\epsilon = 0$, the same algorithm for $\eta = 0$ finds an assignment of weight at most $\ell \cdot s^*$ that satisfies all the constraints. □

B Advice Complexity of Sampling

In this appendix, we demonstrate the robustness of advice complexity of sampling. We first recall the definition of the advice complexity of sampling from Definition 3.11.

Definition B.1 (Advice complexity of sampling)**.** *A distribution $\mathcal{D}$ over $\{0,1\}^n$ is said to be* samplable with advice complexity ℓ *if there exists a function $\alpha : \{0,1\}^{2^{2n}} \to \{0,1\}^{\leq \ell}$ such that every computation $U(\alpha(r); r, n)$ halts with an output in $\{0,1\}^n \cup \{\perp\}$,*

$$\Pr_{r \sim \{0,1\}^{2^{2n}}}[U(\alpha(r); r, n) = \perp] \leq \frac{1}{4},$$

and the distribution of $U(\alpha(r); r, n)$ conditioned on $U(\alpha(r); r, n) \neq \perp$ has statistical distance at most 2^{-2n} from $\mathcal{D}$. The function α need not be computable.

To compare this definition with Definition 1.1, it is instructive to think that $\alpha(r)$ consists of a constant-size description d_S for a sampling procedure S and an advice string.

Definition B.2 (Coding property)**.** *We say that a distribution $\mathcal{D}$ over $\{0,1\}^n$ has a* coding property with error A *if*

$$\mathrm{K}(x \mid n) \leq -\log \mathcal{D}(x) + A$$

for every $x \in \mathsf{Support}(\mathcal{D})$.

Definition B.3 (∞-Rényi divergence)**.** *For two distributions P and Q over $\{0,1\}^n$ such that $\mathsf{Support}(P) \subseteq \mathsf{Support}(Q)$, define*

$$\Delta_\infty(P \parallel Q) \coloneqq \log \max\left\{ \frac{P(x)}{Q(x)} \,\middle|\, x \in \mathsf{Support}(P) \right\}.$$

Definition B.4 (Universal distribution)**.** *The* universal distribution *over $\{0,1\}^n$, denoted by m_n, is defined by*

$$\mathrm{m}_n(x) \coloneqq \frac{2^{-\mathrm{K}(x|n)}}{Z_n}, \qquad Z_n \coloneqq \sum_{y \in \{0,1\}^n} 2^{-\mathrm{K}(y|n)}.$$

We now present the equivalence among the three different measures up to a constant factor.

Proposition B.5. *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$ be a family of distributions such that $\mathsf{Support}(\mathcal{D}_n) \subseteq \{0,1\}^n$, and let $\ell : \mathbb{N} \to \mathbb{N}$ satisfy $\ell(n) \geq 1$ and $\ell(n) = o(n)$. The following are equivalent.*

1. *$\mathcal{D}_n$ is samplable with advice complexity $O(\ell(n))$ for all sufficiently large n.*
2. *$\Delta_\infty(\mathcal{D}_n \parallel \mathrm{m}_n) = O(\ell(n))$ for all sufficiently large n.*
3. *$\mathcal{D}_n$ has a coding property with error $O(\ell(n))$ for all sufficiently large n.*

Proof. We first observe the equivalence between Items 2 and 3. Item 2 is equivalent to

$$-\log \mathrm{m}_n(x) \leq -\log \mathcal{D}_n(x) + O(\ell(n)) \tag{28}$$

for every $x \in \mathsf{Support}(\mathcal{D}_n)$. By definition, $-\log \mathrm{m}_n(x) = \mathrm{K}(x \mid n) + \log Z_n$. Since $\mathrm{K}(y \mid n) \leq n + O(1)$ for every $y \in \{0,1\}^n$, we have $Z_n \geq \Omega(1)$; and $Z_n \leq 1$ follows from Kraft's inequality. Thus $|\log Z_n| = O(1)$, and (28) is equivalent to Item 3.

We prove Item 3 $\Rightarrow$ Item 1. Assume that, for some constant $C \geq 1$,

$$\mathrm{K}(x \mid n) \leq -\log \mathcal{D}_n(x) + C\ell(n) \tag{29}$$

for every $x \in \mathsf{Support}(\mathcal{D}_n)$ for all sufficiently large n. Since $\ell(n) = o(n)$, we have $C\ell(n) \leq n$ for all sufficiently large n.

Consider the following fixed sampling procedure. Its advice string contains two integers T and t. Using the random string $r \in \{0,1\}^{2^{2n}}$, it reads T pairs

$$(d^{(1)}, v^{(1)}), \dots, (d^{(T)}, v^{(T)}), \qquad d^{(i)} \in \{0,1\}^{2n}, \quad v^{(i)} \in \{0, \dots, 2^{3n+3} - 1\}.$$

If $t = 0$, the procedure outputs $\perp$. Otherwise, it dovetails the computations $U(y; n)$ over all prefixes y of $d^{(t)}$ and outputs the first value in $\{0,1\}^n$ that appears. The output is unique because the domain of U is prefix-free.

We now specify the advice string. For a uniformly random $d \in \{0,1\}^{2n}$, let $X(d) = x$ if $U(y; n) = x \in \{0,1\}^n$ for some prefix y of d, and leave $X(d)$ undefined if no such prefix exists. Define

$$Q(x \mid n) \coloneqq \sum_{y \in \{0,1\}^{\le 2n} : U(y;n) = x} 2^{-|y|} \ge 2^{-\mathrm{K}(x|n)}.$$

It follows from (29) that

$$\frac{\mathcal{D}_n(x)}{2^{C\ell(n)} Q(x \mid n)} \le 1. \tag{30}$$

For every $x \in \{0,1\}^n$, define the threshold

$$\theta_x \coloneqq 2^{3n+3} \cdot \mathcal{D}_n(x) / (2^{C\ell(n)} Q(x \mid n)).$$

By (30), we have $0 \le \theta_x \le 2^{3n+3}$. Set $T \coloneqq 2^{C\ell(n)+3}$. Since $C\ell(n) \le n$, the T pairs above use at most $T(5n+3) \le 2^{2n}$ bits for all sufficiently large n. Call the i-th trial accepting if $X(d^{(i)})$ is defined and, for $x \coloneqq X(d^{(i)})$,

$$v^{(i)} < \lfloor \theta_x \rfloor.$$

For a given random string r, define t to be the first accepting index in $[T]$. If no such index exists, define $t \coloneqq 0$. The advice program $\alpha(r)$ is the fixed procedure together with self-delimiting encodings of T and t, and therefore

$$|\alpha(r)| \le 2\lceil \log(T+1) \rceil + O(\log\log(T+2)) = O(\ell(n)).$$

With this choice of t, every advised computation halts with an output in $\{0,1\}^n \cup \{\perp\}$.

It remains to analyze the distribution. For a fixed $i \in [T]$, let $x^{(i)} = X(d^{(i)})$ if $X(d^{(i)})$ is defined, and leave $x^{(i)}$ undefined otherwise. Let $A_i(x)$ be the event that the i-th trial accepts with output x, and let $A_i \coloneqq \bigcup_{x \in \{0,1\}^n} A_i(x)$. Define $p \coloneqq \Pr[A_i]$.

We need two estimates. First, we show that one trial accepts with probability large enough, so the procedure rarely outputs $\perp$. For every $x \in \{0,1\}^n$, the probability that $x^{(i)} = x$ is $Q(x \mid n)$, and

$$\Pr[A_i(x)] = Q(x \mid n)\, 2^{-(3n+3)} \lfloor \theta_x \rfloor.$$

Replacing θ_x by $\lfloor \theta_x \rfloor$ decreases it by less than one, and hence

$$\Pr[A_i(x)] \ge 2^{-C\ell(n)} \mathcal{D}_n(x) - 2^{-(3n+3)} Q(x \mid n). \tag{31}$$

Summing over x, and using $\sum_x Q(x \mid n) \le 1$, we obtain

$$p \ge 2^{-C\ell(n)} - 2^{-(3n+3)} \ge 2^{-C\ell(n)-1} \tag{32}$$

for all sufficiently large n. Therefore, the sampling procedure outputs $\perp$ exactly when no trial accepts, and

$$\Pr[U(\alpha(r); r, n) = \perp] = (1-p)^T \le \exp(-pT) \le e^{-4} < \frac{1}{4}.$$

Second, we show that rounding down changes the conditional output distribution by only a negligible amount. Rounding down can only decrease the acceptance probability, and by (31), for every $x \in \{0,1\}^n$,

$$0 \le 2^{-C\ell(n)}\mathcal{D}_n(x) - \Pr[A_i(x)] \le 2^{-(3n+3)}Q(x \mid n).$$

Summing over x gives

$$0 \le 2^{-C\ell(n)} - p \le 2^{-(3n+3)}.$$

The numerator in the following bound is exactly the total probability mass lost by rounding down, so the output distribution of one trial conditioned on acceptance has statistical distance at most

$$\frac{2^{-C\ell(n)} - p}{p} \le 2^{C\ell(n)-(3n+3)+1} \le 2^{-2n} \tag{33}$$

from $\mathcal{D}_n$. Conditioned on a non-$\perp$ output, the first accepting trial has the one-trial output distribution conditioned on acceptance, so (33) proves the required sampling guarantee. This proves Item 3 $\Rightarrow$ Item 1.

We prove Item 1 $\Rightarrow$ Item 3. Assume that $\mathcal{D}_n$ is samplable with advice complexity $L_n = O(\ell(n))$, witnessed by a function α. Define

$$P_n(x) := \mathbb{E}_{r\sim\{0,1\}^{2^{2n}}}\left[\sum_{a:U(a;r,n)=x} 2^{-|a|}\right].$$

The family $\{P_n\}$ is a uniformly lower semicomputable semimeasure: one can dovetail all computations, and Kraft's inequality gives $\sum_x P_n(x) \le 1$. Since the actual advice program is one of the terms in the sum,

$$P_n(x) \ge 2^{-L_n}\Pr_r[U(\alpha(r);r,n) = x]. \tag{34}$$

If $\mathcal{D}_n(x) \ge 2^{-2n+1}$, then

$$\Pr_r[U(\alpha(r);r,n) = x \mid U(\alpha(r);r,n) \ne \perp] \ge \mathcal{D}_n(x) - 2^{-2n} \ge \frac{1}{2}\mathcal{D}_n(x).$$

Since the non-$\perp$ probability is at least $3/4$, (34) gives

$$P_n(x) \ge \frac{3}{8}\,2^{-L_n}\mathcal{D}_n(x).$$

It follows from the coding theorem for lower semicomputable semimeasures (see, e.g., [Lee06; LV19]) that

$$\mathrm{K}(x \mid n) \le -\log \mathcal{D}_n(x) + L_n + O(1).$$

If instead $0 < \mathcal{D}_n(x) < 2^{-2n+1}$, then $-\log \mathcal{D}_n(x) > 2n-1$, whereas a literal description gives $\mathrm{K}(x \mid n) \le n + O(1)$, and the same inequality follows for all sufficiently large n. Since $L_n = O(\ell(n))$, Item 3 follows. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il