Electronic Colloquium on Computational Complexity, 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 $\ell/\mathsf{polylog}(\ell)$, where ℓ is a new complexity measure 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 an approximation factor $O(\ell)$.

In particular, Pessiland is ruled out if and only if the small quantitative gap in approximation factors $\ell/\mathsf{polylog}(\ell)$ 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 $\ell/\mathsf{polylog}\ell$, improving the previous inapproximability factor of $\ell^{\Omega(1)}$ by Hirahara (FOCS 2022). Our inapproximability factor is optimal up to the $\mathsf{polylog}\ell$ factor unless $\mathsf{NP} \subseteq \mathsf{coAM}$ because the CMMSA problem with an approximation factor $O(\ell)$ is in coAM.

*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 some parameter $\ell = \ell(n) = \omega(1)$, the following hold.

- NP *is hard on average* $\iff$ *agnostic learning with factor* $\ell/\mathsf{polylog}(\ell)$ *is hard on average.*
- *A one-way function exists* $\iff$ *agnostic learning with factor* $\omega(\ell)$ *is hard on average.*

Corollary (informal)**.** *Pessiland exists if and only if average-case agnostic learning is hard with an approximation factor* $\ell/\mathsf{polylog}(\ell)$ *and is easy with an approximation factor* $O(\ell)$*.*

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 $\ell/\mathsf{polylog}(\ell)$ and $\omega(\ell)$, 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 3.16 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 $\ell(n)$ *if there exist a (not necessarily efficient) algorithm S and a family $\alpha = \left\{\alpha_n : \{0,1\}^* \to \{0,1\}^{\ell(n)}\right\}_{n\in\mathbb{N}}$ of functions such that $\mathcal{D}_n$ is statistically identical to the distribution generated by $S(n, r, \alpha_n(r))$ for a uniformly random string r.*

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 3.16). 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 C 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 as small as $O(1)$, 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 $\varepsilon, \gamma\colon \mathbb{N} \to [0,1]$, $\sigma\colon \mathbb{N} \to \mathbb{N}$, and $\ell\colon \mathbb{N} \to \mathbb{N}$, 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 random and independent samples drawn from a distribution over $\{0,1\}^n \times \{0,1\}$ that is promised to be samplable with advice complexity $\ell(n)$, distinguish the following two cases:*

[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.

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

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

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

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

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 $\varepsilon(n)$-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 $\ell(n) = \omega(1)$, $\varepsilon(n) = o(1)$, and $\sigma(n) = \ell(n)/\mathsf{polylog}\ell(n)$ such that the problem $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ is NP-hard under randomized many-one reductions for every constant $\gamma \in (0,1)$.*

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 $\ell(n) = \omega(1)$, $\varepsilon(n) = o(1)$, and $\sigma(n) = \ell(n)/\mathsf{polylog}\ell(n)$ such that for every constant $\gamma \in (0,1)$, the following are equivalent.*

- $\mathsf{DistNP} \not\subseteq \mathsf{HeurBPP}$ *(i.e., there is a problem in* NP *that is hard on average with respect to some polynomial-time samplable distribution).*
- $(\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell], \mathcal{D}) \notin \mathsf{HeurBPP}$ *for some polynomial-time samplable distribution $\mathcal{D}$. (i.e., $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ 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 $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ characterizes the existence of one-way functions when $\sigma(n) = O(\ell(n))$, based on the average-case learning algorithms in Pessiland [IL90; HN23].

Theorem 1.5. *For every constant $\gamma \in (0,1)$, there exists a constant C such that for all functions ℓ and ε with $\omega(1) \leq \ell(n) \leq n^{O(1)}$ and $\omega(1) \leq \varepsilon(n)^{-1} \leq n^{O(1)}$, the following are equivalent.*

- *There exists an infinitely-often one-way function.*
- $(\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell], \mathcal{D}) \notin \mathsf{HeurBPP}$ *for $\sigma(n) \coloneqq C \cdot \ell(n)$ and for some polynomial-time samplable distribution $\mathcal{D}$.*
- $(\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell], \mathcal{D}) \notin \mathsf{HeurBPP}$ *for some $\sigma(n) \leq n^{O(1)}$ 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)**.** *Pessiland exists if and only if for some* $\ell(n) = \omega(1)$*,* $\varepsilon(n) = o(1)$*,* $\gamma \in (0,1)$*,* $\sigma_1(n) = \ell(n)/\mathsf{polylog}\ell(n)$ *and* $\sigma_2(n) = O(\ell(n))$*,*

- $(\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma_1}\mathrm{Learn}[\ell], \mathcal{D}) \notin \mathsf{HeurBPP}$ *and*
- $(\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma_2}\mathrm{Learn}[\ell], \mathcal{D}) \in \mathsf{HeurBPP}$*.*

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 Corollary1.6, namely a characterization of Pessiland via GapLearn with approximation gap ℓ^α versus $O(\ell)$ for some small constant $\alpha > 0$. Corollary1.6 sharpens the hardness factor from ℓ^α to $\ell/\mathsf{polylog}\ell$, 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 $\sigma_1(n) = \ell(n)/\mathsf{polylog}\ell(n)$ and $\sigma_2(n) = O(\ell(n))$ 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 $\{1,\dots,n\}$ 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* $\varepsilon,\gamma\colon \mathbb{N} \to [0,1]$ *and* $\sigma\colon \mathbb{N} \to \mathbb{N}$*, the problem* $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathfrak{C}$-CMMSA *is defined as follows: Given as input a collection* $\Phi = \{\varphi_1,\dots,\varphi_m\} \subseteq \mathfrak{C}_n$*, 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*

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

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* $\sigma(n)\cdot s$ *such that*

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

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 a function $\ell\colon \mathbb{N} \to \mathbb{N}$, 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 $\ell(n)$ literals. Hirahara [Hir22] proved NP-hardness of $\mathrm{Gap}^{0,\gamma}_{\sigma}\mathsf{F}[\ell]$-CMMSA for $\sigma(n) = \ell(n)^{\alpha}$ and $\gamma(n) = 1/\sigma(n)$ for some constant $\alpha > 0$, and left as an open question whether the inapproximability factor σ can be improved to $\ell(n)^{1-o(1)}$. Our main technical contribution is to answer this question affirmatively.

Theorem 1.8. $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathsf{F}[\ell]$-CMMSA *is* NP*-hard under randomized many-one reductions for some functions* $\ell(n) = \omega(1)$, $\gamma(n) = o(1)$, $\varepsilon(n) = \ell(n)^{-\omega(1)}$, *and* $\sigma(n) = \ell(n)/\mathsf{polylog}\ell(n)$.

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

Theorem 1.9. *For every constant* $\gamma \in [0, 1/4)$, *there exists a constant* C *such that for every* $\ell(n) = \omega(1)$ *and every* $\varepsilon(n) = o(1)$, *the following hold.*

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

More generally, for any class $\mathfrak{C} = \{\mathfrak{C}_n\}_{n\in\mathbb{N}}$ of monotone functions, we prove $\mathrm{Gap}^{\varepsilon,\gamma}_{O(\sigma)}\mathfrak{C}$-CMMSA $\in$ coAM, where $\sigma(n)$ 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. *Let* $\mathfrak{C} = \{\mathfrak{C}_n\}_{n\in\mathbb{N}}$ *be the class of monotone functions for which there exists a secret sharing scheme with total share size* $\sigma(n) \geq \omega(1)$. *For every constant* $\gamma \in [0, 1/4)$, *there exists a constant* C *such that for every* $\varepsilon(n) = o(1)$, *the following hold.*

- *If* $\mathrm{Gap}^{\varepsilon,\gamma}_{C\cdot\sigma}\mathfrak{C}$-CMMSA $\notin$ BPP, *then* DistNP $\not\subseteq$ AvgBPP (*i.e., there exists a problem in* NP *that is hard on errorless average with respect to some polynomial-time samplable distribution*).
- *If* $(\mathrm{Gap}^{\varepsilon,\gamma}_{C\cdot\sigma}\mathfrak{C}\text{-CMMSA}, \mathcal{D}) \notin$ AvgBPP *for some polynomial-time samplable distribution* $\mathcal{D}$, *then an infinitely-often one-way function exists.*[2]

[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.

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 $\sigma(n) = 1.01\ell(n)$ and the soundness error $\gamma(n)$ is sufficiently smaller than $1/(sn)$. The first item of Theorem 1.10 improves this result in that $\gamma(n)$ can be as large as a constant.[3]

Consequently, Theorems 1.8 to 1.10 reveal a sharp threshold at $\sigma(n) \approx \ell(n)$ in the complexity of $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathsf{F}[\ell]$-CMMSA: It is NP-hard if $\sigma(n) \leq \ell(n)/\mathsf{polylog}\ell(n)$; It is in coAM if $\sigma(n) \geq O(\ell(n))$, 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 B, we present a simple approximation algorithm based on linear programming that solves $\mathrm{Gap}^{0,1}_{\sigma}\mathsf{DNF}[\ell]$-CMMSA for $\sigma(n) = \ell(n)$, where $\mathsf{DNF}[\ell]$ denotes the class of monotone DNF formulas over n variables with at most $\ell(n)$ terms. (In particular, it solves $\mathrm{Gap}^{0,1}_{\sigma}\mathsf{Junta}[\ell]$-CMMSA for $\sigma(n) = 2^{\ell(n)}$.) We pose the following algorithmic challenge of improving this approximation algorithm.

Open Question 1.11. Design a randomized algorithm M such that for every $\ell(n) = \omega(1)$, every $\varepsilon(n) = o(1)$, 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 $\sigma(n) \coloneqq 2^{0.6\ell(n)}$.
- $\mathbb{E}_{x\sim\mathcal{D}_n}[t_M(x)^{\alpha}] \leq O(n)$ 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* 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].

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

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 $\sigma(n) = O(\ell(n))$ of Theorem 1.5 can be improved to $\sigma(n) = \ell(n)/\mathsf{polylog}\ell(n)$ 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 $\sigma(n) \coloneqq C \cdot \ell(n)$. 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 5.1 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 BPP-reduction that samples polynomially many formulas from the distribution in DMMSA to form a collection for CMMSA (see Proposition 5.3). Therefore, we will identify DMMSA with CMMSA below.

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

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

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 of size n to a Gap_σDMMSA instance, where the leaf size of monotone formulas is $\ell(n) = \omega(1)$ (thus, the total size of secret sharing is at most ℓ) and the approximation factor is $\sigma(n) = \ell(n)/\mathsf{polylog}\ell(n)$. This is the most technical part in this paper, where we combine the previous reductions from [DS04; Hir22] with a suitable *long code test*. 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 6. 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

Our reduction from 3-SAT to GapDMMSA refines the approach presented in [DS04; Hir22]. In particular, the previous work [Hir22, Theorem 5.2] established the NP-hardness of approximating the minimum weight of CMMSA up to a factor of ℓ^α with respect to the leaf size ℓ of the supported formulas, where $\alpha > 0$ is a small constant. In this work, we improve the approximation factor to $\ell/\mathsf{polylog}\ell$ for a super-constant $\ell(n) = \omega(1)$, where n is the instance size of CMMSA.

Coding in the Reduction

A main idea for the improvement is to change the encoding of alphabets from the previous one.

To see the role of encoding in the reduction, let us first review the previous proof by Hirahara [Hir22]. The proof mainly follows that of [DS04] and is based on the PCP theorem of [DFKRS11; DHK15] with low (sub-constant) soundness error. The PCP system is naturally interpreted as an instance of MaxCSP over variables $x_1, x_2, \ldots, x_n$ with an alphabet set Σ of the PCP system [cf. AB09, Section 11]. In the previous reduction, in order to transform the MaxCSP problem into a CMMSA instance (since constraints in MaxCSP are not generally monotone), new binary variables $X_{i,a}$ are introduced for each pair of a variable x_i and an alphabet $a \in \Sigma$. Intuitively, $X_{i,a} = 1$ represents the *claim* that x_i is assigned the alphabet a in the original MaxCSP. Each constraint φ_C in the resulting CMMSA instance corresponds to a constraint C (say, over variables $(x_{i_1}, \ldots, x_{i_k})$) in the MaxCSP instance. Here, φ_C is defined to be a monotone DNF formula that asks whether there exists a satisfying assignment $(a_1, \ldots, a_k)$ for C such that $X_{i_j,a_j} = 1$ for each $j \in [k]$. The

weight of each $X_{i,a}$ is set propositionally to the number of the occurrences of x_i in the MaxCSP instance.

In the reduction above, we can interpret any assignment for the reduced CMMSA instance as a *coding* of an assignment for MaxCSP. Here, for each $a \in \Sigma$, the (valid) coding of a is defined as $c(a) \in \{0,1\}^{|\Sigma|}$, where $c(a)_i = 1$ if and only if $i = a$. For the reduced CMMSA instance, we expect to assign $\{X_{i,a}\}_{a\in\Sigma}$ according to the coding of an assignment to $\{x_i\}$. Specifically, for each assignment a_i to x_i, we expect to assign $c(a_i) \in \{0,1\}^{|\Sigma|}$ to $\{X_{i,a}\}_{a\in\Sigma}$. It is not hard to observe that if there exists a satisfying assignment to the MaxCSP instance, then the valid coding provides a satisfying assignment for the reduced CMMSA instance. This establishes the completeness proof for Yes instances.

The soundness proof for No instances is less straightforward and requires decoding with error correction. Specifically, we need to extract an assignment for the MaxCSP instance from an arbitrary low-weight assignment for CMMSA, which may be an invalid encoding and thus require error correction. Previously, this was performed using a probabilistic argument and the low soundness of the PCP system, but this analysis presents the following bottleneck for our purposes. Notice that the valid coding of an assignment has weight $1/|\Sigma|$, and the leaf size of CMMSA is at least $|\Sigma|$ (in fact, it is polynomial in $|\Sigma|$, depending on the number of satisfying assignment for constraints). Even if we achieve the optimal leaf size of $|\Sigma|$, in order to show the nearly linear relationship between leaf size and the inapproximability factor of weights in the DMMSA instance, we need to establish soundness even when the given assignment to DMMSA has a weight close to 1.

The difficulty in proving the soundness arises from the fact that the assignment to DMMSA can be arbitrary, with only the requirement that its weight is less than (but very close to) 1. Recall that the weight of the trivial assignment, which assigns all variables to 1, is 1. Namely, when the given assignment for DMMSA has a weight of almost 1, it may claim almost all possible alphabets simultaneously for the variables of MaxCSP. This results in very poor decoding just by randomly selecting from nearly all alphabets, and it is unclear whether the soundness of the currently known PCP systems (or even the sliding scale conjecture) can handle this for the soundness proof.

To address this difficulty, we aim to prevent such malicious coding. A natural approach is to use a local test to enforce valid codings. However, each formula in the DMMSA instance must be monotone, and it is unclear whether we can implement a local test for the coding above as monotone formulas.

Long Code and Local Test by Monotone Function

These observations lead us to change the previous coding to *long code*, which allows a desired monotone local test. Let us review the long code. For each alphabet $a \in \Sigma$, we encode it by a binary string $\mathsf{lc}(a)$ of length $2^{|\Sigma|}$, where $\mathsf{lc}(a)$ is indexed by a subset $S \subseteq \Sigma$ and defined as

$$\mathsf{lc}(a)_S = \begin{cases} 1 & \text{if } a \in S \\ 0 & \text{if } a \notin S. \end{cases}$$

Thus, instead of $\{X_{i,a}\}_{a\in\Sigma}$, we introduce variables $\{X_{i,S}\}_{S\subseteq\Sigma}$ for each variable x_i in our new reduction. Furthermore, we use a *p-biased* long code, assigning the weight of each variable $X_{i,S}$ to be propositional to $p^{|S|}(1-p)^{|\Sigma|-|S|}$ for each i and S. The quantity corresponds to the probability that S is selected when each element in Σ is independently chosen with probability p. The small bias p is crucial for making the inapproximability factor nearly linear in the leaf size: when $p = 1/2$ (i.e., the unbiased case), the weight of the valid encoding is exactly $1/2$, so the best achievable inapproximability factor is 2 regardless of the leaf size. In particular, for a super-constant leaf size

$\ell = \omega(1)$, we need to set p to be sub-constant, as the best achievable gap in weight between Yes and No cases is $1/p$.

Now we describe how we locally test the long code in our reduction. Recall that (i) our objective is to reduce MaxCSP to DMMSA, so each local test must be formulated as a monotone formula, and (ii) in our soundness proof, we can assume that the weight of a given assignment for DMMSA (i.e., the weight of the encoded assignment) is bounded. Our local test builds on the one presented by Ghoshal and Lee [GL22], with slight modifications tailored to meet our specific requirements.

Ghoshal–Lee Test

We review the Ghoshal–Lee test. Let $f \in \{0,1\}^{2^{|\Sigma|}}$ be a given coding for the alphabet Σ. Our goal is to test whether $f = \mathsf{lc}(a)$ for some $a \in \Sigma$ or if f significantly differs from all $\{\mathsf{lc}(a)\}_{a\in\Sigma}$. Specifically, we aim to reject all codings that decode poorly to the original MaxCSP assignment.

We can naturally interpret f as a function $f\colon 2^\Sigma \to \{0,1\}$, where 2^Σ represents the power set of Σ. The Ghoshal–Lee test queries f on some points $x^1, \dots, x^d$ and then accepts if and only if $\wedge_i f(x^i) = 1$. The novelty of Ghoshal–Lee test lies in the correlated selection of the points $x^1, \dots, x^d \subseteq \Sigma$. Let $\rho \approx 1/(d^2 \log p^{-1})$ be a parameter representing correlation. For each $a \in \Sigma$, we select $b_a \in \{0,1\}$ so that $b_a = 1$ with probability ρ. If $b_a = 1$, then a either contained in all of $x^1, \dots, x^d$ with probability p or excluded from all of them with probability $1-p$, where p is the bias of the long code. Otherwise, if $b_a = 0$, the membership of a in each x^i is determined independently; each x^i contains a with probability p, independently across $x^1, \dots, x^d$.

The correlation above is intended to make the gap in acceptance probability between complete and sound cases exponentially large in d, which is essential for the near-linear relationship between leaf size and inapproximability in our result, as we will explain later. For completeness proof, it is easy to verify that the acceptance probability for $f = \mathsf{lc}(a)$ is at least $p \cdot \rho$, as all of $x^1, \dots, x^d$ contain a with probability at least $p \cdot \rho$.

In contrast, Ghoshal and Lee [GL22] derived the soundness from the Invariance principle [Mos10] and Gaussian bound [KS15]. Let $\mu = \mathbb{E}_x[f(x)]$, where $x \subseteq \Sigma$ is selected with bias p. Informally, their soundness proof shows that each f satisfies one of the following:

- The acceptance probability of the Ghoshal–Lee test for f is at most approximately μ^d;
- f has a coordinate with large influence,

where the influence of a coordinate i is defined as a variance on coordinate i (see Definition 3.1). Intuitively, the above shows that if f has no coordinate with large influence (and $\mu = \mathbb{E}[f]$ is bounded), the acceptance probability decreases exponentially in d, allowing us to reject f with high probability. Otherwise, f must have a coordinate with large influence (often called a notable coordinate), which appears to be a good candidate for decoding since the value of f is strongly affected by such coordinates. Note that, in general, the number of coordinates with large influence is not bounded (e.g., a parity on all coordinates). However, it is well-known that we can restrict the number of notable coordinates by adding a small noise to the queried points (see Lemma 3.3).

The Modification of the Test

Recall that we need to choose p to be $o(1)$ for our purposes. The Ghoshal–Lee test has an acceptance probability at most p even for valid long codes, resulting in a poor satisfiability probability of $o(1)$ for yes cases in the resulting DMMSA instance. To address this, we modify the test by changing the predicate. Note that the previous work [GL22] applied the AND predicate to establish a tight

inapproximability result for the Densest-Subgraph (i.e., biased MaxAND) problem, but we do not need to restrict the predicate to AND as long as it is monotone.

Our modification is very simple: we repeat the Ghoshal–Lee test M times and take the OR of the results, which yields monotone d-DNF formulas of size M as predicates. In our setting, we select d and M such that $d \approx \mathsf{polylog} p^{-1}$ and $M \approx p^{-1}\rho^{-1} = p^{-1}\mathsf{polylog} p^{-1}$. This choice ensures (i) the acceptance probability of $1 - o(1)$ for valid long codes, and (ii) the acceptance probability of $o(1)$ for invalid codes with no large influence coordinate and weight $1 - \Omega(1)$. Since the weight of valid long codes is p, the gap of weights between two cases (i) and (ii) is $\Omega(p^{-1})$, which is nearly linear in the leaf size $Md \approx p^{-1}\mathsf{polylog} p^{-1}$. Note that the correlated selection in the Ghoshal–Lee test is essential for this relationship; if each queried point were selected independently, the acceptance probability in complete cases would drop to p^d. To compensate, we would need to set $M \approx p^{-d}$, resulting in the leaf size $Md > p^{-d}$, but the gap in weights (i.e., inapproximability factor) would remain at most p^{-1}.

Reduction to DMMSA with Long Code Test

Finally, we explain how we combine the modified Ghoshal–Lee test with the proof of NP-hardness for DMMSA. Recall that the previous work [GL22] relied on the Small Set Expansion Hypothesis, which is known to be equivalent to the Unique Games Conjecture with an additional expansion property [RST12]. Their approach required a strong connection between a global bias across all variables and each long code component [see GL22, Section 3.2]. In contrast, for our purposes, we do not need such a strong connection, allowing us rely solely on the PCP theorem. However, the MaxCSP resulting from the PCP theorem is much less structured than those obtained from the Small Set Expansion Hypothesis or the Unique Games Conjecture. It is known that a test that distinguishes long codes from codes with no notable coordinates using a predicate φ is automatically transformed into an NP-hardness result for MaxCSP with constraints φ *under the Unique Games Conjecture* [KKMO07; ODo14, Theorem 7.40]. Our reduction, however, does not follow this framework, as we do not assume the Unique Games Conjecture. Instead, our approach is more ad hoc, leveraging an implicit pairwise checkability of the Ghoshal–Lee test.

To show the NP-hardness, we use the two-prover projection PCP system with low soundness error, presented by Moshkovitz and Raz [MR10]. This PCP system can be interpreted as a MaxCSP instance with a constraint graph represented by a biregular graph (L, R, E), where each vertex in $L \cup R$ is regarded as a variable, and each edge $e \in E$ is regarded as a constraint. Each vertex $v \in L$ (resp. $v \in R$) takes an alphabet in a set Σ_L (resp. Σ_R), with $|\Sigma_L| \geq |\Sigma_R|$. Each $e \in E$ is associated with a *partial* function $\pi_e \colon \Sigma_L \to \Sigma_R$, representing the acceptable relations of alphabets assigned to the endpoints of each edge. Specifically, a constraint $(v, v') \in E$ is satisfied if v (resp. v') is assigned with $a \in \Sigma_L$ (resp. $b \in \Sigma_L$) so that $\pi_e(a) = b$; if $\pi_e(a)$ is undefined, the constraint is not satisfied. We apply this PCP theorem above in the setting where $\omega(1) \leq |\Sigma_L|, |\Sigma_R| \leq o(\log n)$.

In the reduction, we introduce new variables for the long code, i.e., for each $v \in L$ (resp. $v' \in R$), we introduce variables $\{X_{v,S}\}_{S \subseteq \Sigma_L}$ (resp. $\{X_{v',T}\}_{T \subseteq \Sigma_R}$), which take binary values for a DMMSA instance. Note that the left and right vertices have different roles: left alphabets specify acceptable right alphabets, but the converse does not hold. Furthermore, the two sides have different alphabet sets, which yields different long codings, and we need to consider them separately.

Implementing the Long-Code Test with Consistency

We define the distribution on monotone formulas in the DMMSA instance by specifying its sampler. First, it selects an edge $e = (v, v') \in E$ uniformly at random. We interpret $\{X_{v,S}\}_{S \subseteq \Sigma_L}$ (resp.

$\{X_{v',T}\}_{T\subseteq\Sigma_R}$) as a function $f_v\colon 2^{\Sigma_L}\to\{0,1\}$ (resp. $g_{v'}\colon 2^{\Sigma_R}\to\{0,1\}$), which is expected to be a valid long code. The goal is to check the following conditions simultaneously:

1. f_v and $g_{v'}$ are both valid long codes for alphabets $a\in\Sigma_L$ and $b\in\Sigma_R$, respectively;
2. the constraint on e is satisfied, i.e., $\pi_e(a)=b$.

The idea to accomplish the above is very intuitive: we apply the modified Ghoshal-Lee test to f_v, but instead of checking

$$\bigvee_{i\in[M]} f_v(x^{(i,1)})\wedge f_v(x^{(i,2)})\wedge\cdots\wedge f_v(x^{(i,d)}),$$

we replace f_v with $g_{v'}\circ\pi_e$ on $x^{(i,2)},\dots,x^{(i,d)}$, and check

$$\bigvee_{i\in[M]} f_v(x^{(i,1)})\wedge g_{v'}\circ\pi_e(x^{(i,2)})\wedge\cdots\wedge g_{v'}\circ\pi_e(x^{(i,d)}),$$

where $\pi_e(x)\in 2^{\Sigma_R}$ is defined as $i\in\pi_e(x)$ iff $\exists j\in x$ s.t. $\pi_e(j)=i$.

It is not hard to observe that if $(f_v,g_{v'})$ are valid long codes for (a,b) with $\pi_e(a)=b$, then the above is satisfied with the same acceptance probability as the original test. This completes the proof of completeness for the reduction. In contrast, the soundness proof requires much less immediate argument, as discussed below. For simplicity, we ignore the small noise used to limit the number of notable coordinates in this section.

The soundness proof relies on the following pairwise checkability of the Ghoshal–Lee test. Let $f,g\colon 2^{\Sigma_L}\to\{0,1\}$ be a pair of possibly invalid coding for Σ_L with $\max\{\mathbb{E}_x[f],\mathbb{E}_x[g]\}\le\mu$. For convenience, let us call the test $f(x^1)\wedge(\wedge_{2\le i\le d}g(x^i))$, as performed in the reduction, the *pairwise* Ghoshal–Lee test. Then, each (f,g) satisfies one of the following:

- The acceptance probability of the *pairwise* Ghoshal–Lee test for (f,g) is roughly at most μ^d;
- f and g has a *common* notable coordinate.

This pairwise testability follows from the underlying functional analysis, particularly the multilinear Gaussian stability bound of [Mos10; KS15], used in the soundness analysis of [GL22]. For a formal description, see Theorem 5.14 and Section 5.4.

First, we consider the warm-up case where $\Sigma_L=\Sigma_R$ and all constraints are permutations (i.e., a unique game instance) to provide intuition for the soundness proof. In this case, the distribution of $\pi_e(x)$ is identical to that of x, where $x\in 2^{\Sigma_L}$ is drawn from the p-biased product distribution, regardless of e. We assign weights to the variables $\{X_{v,S}\}_{S\subseteq\Sigma_L}$ and $\{X_{v',T}\}_{T\subseteq\Sigma_R}$ according to the p-biased distribution (for S and T).

Using a standard probabilistic argument, we can show that any DMMSA assignment with sufficiently small weight ensures that, for almost all edges, the endpoints (v,v') are assigned to codes $(f_v,g_{v'})$ with weights bounded by a constant μ, i.e., $\mathbb{E}_x[f_v(x)],\mathbb{E}_x[g_{v'}(x)]\le\mu$. We call such an edge "good." For any good edge $e=(v,v')$, it holds that $\mathbb{E}_x[g_{v'}\circ\pi_e(x)]\le\mu$, as $\pi_e(x)$ is identically distributed to x. Thus, the (OR of) pairwise Ghoshal–Lee test for $(f_v,g_{v'}\circ\pi_e)$ presents two possibilities: (i) the test is rejected with high probability, or (ii) there exists a common notable coordinate for $(f_v,g_{v'}\circ\pi_e)$.

Suppose a good edge satisfies case (ii). For each notable coordinate i of $g_{v'}\circ\pi_e$, it is not hard to verify that the coordinate $\pi_e(i)$ is notable for $g_{v'}$. Namely, any good edge e satisfying case (ii) has $(i,\pi_e(i))$ (i.e., an acceptable pair of alphabets) contained in the product set of notable coordinates

of f_v and $g_{v'}$. Note that the product set is defined per vertex and does not depend on the edge e. Consequently, by assigning values to the original MaxCSP, where each vertex is assigned a value randomly chosen from the notable coordinates of the corresponding codes induced by the DMMSA assignment, a notable fraction of good edges satisfying case (ii) should be satisfied.

In the reduction, we observe that the size of the set of notable coordinates decreases as the bias p increases, independently of the soundness of the original PCP theorem. Thus, by selecting p to be a large enough sub-constant relative to the soundness, we ensure that the fraction of good edges satisfying case (ii) is not too large; otherwise, the random assignment above would violate the soundness of the PCP theorem. Therefore, for any bounded assignment to DMMSA, almost all edges must fall under case (i), meaning the assignment does not satisfy the constraint selected according to the distribution.

To remove the assumption that π_e are permutations, we observe that the same argument above can be applied if, for $x \in 2^{\Sigma_L}$ drawn from the p-biased product distribution, $\pi_e(x) \in 2^{\Sigma_R}$ is distributed according to a p'-biased product distribution for some $p' \in [0,1]$ regardless of e. In this case, we assign weights to the right variables $\{X_{v',T}\}_{T \subseteq \Sigma_R}$ according to the p'-biased distribution (for T). To achieve this, in the actual reduction, we leverage the hardness of MaxCSP with the additional structure, which we call *Balanced* Label-Cover (Definition 4.3). The problem has a balanced structure such that for each edge e and each right alphabet $b \in \Sigma_R$, the number of the left alphabets $a \in \Sigma_L$ such that $\pi_e(a) = b$ is the same. To establish the base NP-hardness of Balanced Label-Cover, we combine the alphabet reduction technique developed in [MR10] to obtain the 2-query projection PCP with the hardness of approximation for 3SAT established in [Hås01], which has a specific structure in which each variable appears in the same number of clauses.

For details about the NP-hardness of Balanced Label-Cover, see Section 4. For the analysis of the reduction in the asymmetric case, see Section 5.

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 7.5 and 7.7). 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 6). 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 $\mathrm{Gap}_{O(\ell)}$DMMSA under *any* samplable distribution.

Theorem 1.9. We have shown that solving $\mathrm{Gap}_{O(\ell)}$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 7.3.

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 $\mathrm{Gap}_{O(\ell)}$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 some preliminary notions for the formal arguments. In Section 4, we introduce Balanced Label-Cover and establish its NP-hardness as the foundational hardness result for our NP-hardness proof. In Section 5, we formally define GapDMMSA and present the reduction from Balanced Label-Cover to GapDMMSA, which completes the proof of Theorem 1.8. In Section 6, we review Hirahara's reduction [Hir22] from GapDMMSA to GapLearn, which, along with the reduction in Section 5, completes the proof of Theorem 1.3. In Section 7, 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 8, 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 Analysis of Boolean Functions

We review some notions required for the analysis of Boolean functions.

Definition 3.1 (Influence)**.** *For each function $f\colon \{0,1\}^n \to \mathbb{R}$, $i \in [n]$, and $\mu \in [0,1]$, we define the (μ-biased) influence of the i-th coordinate on f as*

$$\mathrm{Inf}_i^\mu(f) := \mathbb{E}_{x_1,\ldots,x_{i-1},x_{i+1},x_n}\left[\mathrm{Var}_{x_i}[f(x_1,\ldots,x_n)]\right],$$

where each x_j is selected according to $\mathrm{Ber}(\mu)$.

Definition 3.2 (Noise operator)**.** *For $n \in \mathbb{N}$ and $\rho, \mu \in [0,1]$, we define N_ρ^μ as a mapping from $x \in \{0,1\}^n$ to a distribution of $y \in \{0,1\}^n$, where each y_i is selected as*

$$\begin{cases} y_i = x_i & \text{with probability } \rho \\ y_i \sim \mathrm{Ber}(\mu) & \text{with probability } 1-\rho. \end{cases}$$

We define the μ-biased noise operator T_ρ^μ (with parameter ρ) as the linear operator on $f\colon \{0,1\}^n \to \mathbb{R}$ defined as

$$\mathrm{T}_\rho^\mu f(x) := \mathbb{E}_{y\sim \mathrm{N}_\rho^\mu(x)}\left[f(y)\right].$$

Lemma 3.3 ([cf. GL22, Lemma 8.4])**.** *For every function $f\colon \{0,1\}^n \to [0,1]$, and every $\mu, \eta, \tau \in (0,1)$, it holds that*

$$\left|\{i \in [n] : \mathrm{Inf}_i^\mu(\mathrm{T}_{1-\eta}^\mu f) \geq \tau\}\right| \leq (\eta\tau)^{-1}.$$

Definition 3.4 ([GL22, Definition 8.1], restated)**.** *For $\rho, \mu \in [0,1]$, $n, k \in \mathbb{N}$, and $x \in \{0,1\}^n$, we define a distribution $\tilde{\mathrm{N}}_\rho^{\otimes k}(x)$ over $(\{0,1\}^n)^k$ as a distribution of $(x^{(1)},\ldots,x^{(k)})$ selected according to the following procedure: for each $i \in [n]$, (i) let $b_i \sim \mathrm{Ber}(\rho)$, and (ii) select $(x_i^{(1)},\ldots,x_i^{(k)})$ as*

$$(x_i^{(1)},\ldots,x_i^{(k)}) = \begin{cases} (x_i,\ldots,x_i) & \text{if } b_i = 1 \\ \mathrm{Ber}(\mu)^{\otimes k} & \text{if } b_i = 0 \end{cases}$$

3.2 Hitters

We will use a useful tool called hitters. For the survey on this subject, refer to [Gol11]. In this work, we particularly use the pairwise-independent hitter, whose sample complexity m is far from optimal, but it has pairwise independence which is useful for our purpose.

Lemma 3.5 ([see Gol11, Appendix C.2])**.** *There exists a polynomial-time deterministic algorithm H^{pw} satisfying the following for every $\ell, \varepsilon^{-1}, \delta^{-1} \in \mathbb{N}$:*

- *$H^{pw}(r, 1^{\varepsilon^{-1}}, 1^{\delta^{-1}})$, where $r \in \{0,1\}^{2\ell}$, outputs $m := 1/(\varepsilon\delta)$ strings $x_1, \ldots, x_m \in \{0,1\}^{\ell}$;*
- *For every $S \subseteq \{0,1\}^{\ell}$ with $|S| \geq \varepsilon 2^{\ell}$,*

$$\Pr_r \left[H^{pw}(r, 1^{\varepsilon^{-1}}, 1^{\delta^{-1}}) \text{ produces some } x_i \text{ such that } x_i \in S \right] \geq 1 - \delta;$$

- *The strings produced by $H^{pw}(r, 1^{\varepsilon^{-1}}, 1^{\delta^{-1}})$ is pairwise independent, i.e., for each $i, j \in [m]$ with $i \neq j$ and $v_i, v_j \in \{0,1\}^{\ell}$,*

$$\Pr_r [x_i = v_i] = 2^{-\ell} \text{ and } \Pr_r [(x_i, x_j) = (v_i, v_j)] = 2^{-2\ell}.$$

3.3 Cryptography and Secret Sharing

We review some notations in cryptography.

Definition 3.6 (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 3.7 (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 3.8 ([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 3.9 (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 $s_1, \dots, s_n \in \{0,1\}^*$;*
- Rec *takes n strings and outputs a bit;*
- *(Completeness) For every $b \in \{0,1\}$, every $S \in \mathcal{A}$, and every $s'_1, \dots, s'_n \in \{0,1\}^*$,*

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

  where $(s_1, \dots, s_n) \leftarrow \mathsf{Share}(b)$;
- *(Security) For every randomized function $f_1, \dots, f_n\colon \{0,1\}^* \to \{0,1\}^*$, every $b \in \{0,1\}$, and every $S \notin \mathcal{A}$,*

$$\Pr_{\{s_i\},\{f_i\},\{s'_i\}}[\mathsf{Rec}(f_1(s'_1), \dots, f_m(s'_n)) = b] = \frac{1}{2},$$

  where $(s_1, \dots, s_n) \leftarrow \mathsf{Share}(b)$ and

$$\begin{cases} s'_i = s_i \ (w.p.\ 1) & \text{if } i \in S \\ s'_i \sim \{0,1\}^{|s_i|} & \text{otherwise.} \end{cases}$$

We say that an authorized set $\mathcal{A}$ admits secret sharing of total size s if it holds that $|s_1| + \cdots + |s_n| \leq s$ 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 3.10 ([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., $\mathsf{F}[\ell] \subseteq \mathfrak{S}_\ell$.

3.4 Algorithmic Information

In this paper, we fix a prefix-free universal Turing machine U arbitrarily. For each $t \in \mathbb{N}$ and input Π to U, 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 3.11 (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 3.12 (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 3.13 ([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 3.14 ([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 3.15. *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 3.14. 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 3.14 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

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

□

3.5 Sampling with Advice

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

Definition 3.16 (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 $\alpha\colon \{0,1\}^{2^{cn}} \to \{0,1\}^\ell$ such that $\mathcal{D}$ is statistically identical to the distribution generated by $U(\alpha(r), r, n)$ for $r \sim \{0,1\}^{2^{cn}}$ conditioned on the event that $U(\alpha(r), r, n) \neq \bot$, and the probability that $U(\alpha(r), r, n) \neq \bot$ is at least $1 - 2^{-2^\ell}$, where $c := 100$.*

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

4 Balanced Label Cover

In this section, we introduce the problem Balanced-Label-Cover and show the NP-hardness from the PCP theorem developed in [MR10]. This is the source of our NP hardness result for DMMSA.

First, we introduce the original problem of Label-Cover.

Definition 4.1 (Label-Cover). *An instance of* LABEL-COVER *is composed of a bipartite multigraph $G = (V_L, V_R, E)$ and two finite alphabet sets Σ_L and Σ_R, where $|\Sigma_L| \geq |\Sigma_R|$. Every vertex in V_L is supposed to get a label in Σ_L, and every vertex in V_R is supposed to get a label in Σ_R. For each edge $e \in E$, there is a constraint defined as a* partial *function $\pi_e\colon \Sigma_L \to \Sigma_R$.*

Given a labeling to the vertices of the graph, i.e., functions $\alpha_L \colon V_L \to \Sigma_L$ and $\alpha_R \colon V_R \to \Sigma_R$, an edge $e = (v, w) \in E$ is said to be satisfied if $\pi_e(\alpha_L(v)) = \alpha_R(w)$ (note that the constraint is unsatisfied if $\pi_e(\alpha_L(v))$ is undefined).

The goal is to find a labeling that maximizes the number of satisfied edges. We say that γ fraction of the edges are satisfiable if there exists a labeling that satisfies γ fraction of the edges.

For each left vertex $e \in E$, we use the notation $\Sigma_{L|e}$ to represent

$$\Sigma_{L|e} := \{a \in \Sigma_L : \pi_e(a) \text{ is defined}\}.$$

We may call $a \in \Sigma_L$ valid on e if $a \in \Sigma_{L|e}$. Note that two edges e and e' that share the left vertex may have different valid alphabets, i.e., $\Sigma_{L|e} \neq \Sigma_{L|e'}$. For convenience, we regard the *partial* constraint function $\pi_e \colon \Sigma_L \to \Sigma_R$ as the *total* constraint function $\pi_e \colon \Sigma_{L|e} \to \Sigma_R$.

Next, we introduced the following additional structure on LABEL-COVER instances. This is a key in our reduction to DMMSA.

Definition 4.2 (Balanced)**.** *For finite sets S, T, we say that a function $\pi \colon S \to T$ is balanced if $|\pi^{-1}(a)| = |\pi^{-1}(b)| = |S|/|T|$ for every $a, b \in T$. Moreover, we say that a* partial *function $\pi \colon S \to T$ is balanced if it is balanced when regarded as a total function whose domain is the set on which π is defined.*

Definition 4.3 (Balanced Label-Cover)**.** *A* Balanced-LABEL-COVER *instance is the same as that of* LABEL-COVER *with the following additional properties: let $G = (V_L, V_R, E)$ be a constraint graph, let Σ_L, Σ_R be alphabet sets, and let $\{\pi_e\}_{e \in E}$ be constraints. Then,*

1. *$|\Sigma_{L|e}| = |\Sigma_{L|e'}|$ for any $e, e' \in E$;*
2. *$\pi_e \colon \Sigma_{L|e} \to \Sigma_R$ is balanced for any $e \in E$.*

We will show that the NP-hardness of approximation for Balanced-LABEL-COVER. The proof is based on the query reduction technique developed in [MR10] with additional observations on their bipartite Locally Decode/Reject Codes (LDRCs). Moreover, we can assume that the fraction of left vertices is larger enough than right vertices by the simple transformation shown below. The lemma is stated as follows.

Lemma 4.4. *For every $\varepsilon \colon \mathbb{N} \to (0, 1]$ with $\varepsilon(n) \geq 1/(\log\log\log n)^{O(1)}$, there exists a reduction from solving 3-SAT of instance size n to distinguishing the following two cases where (i) a given* Balanced-LABEL-COVER *instance is completely satisfiable or (ii) at most $\varepsilon(n)$ of its edges are satisfiable under the promise that the label cover instance has a constraint bipartite graph $G = (V_L, V_R, E)$ and alphabet set (Σ_L, Σ_R) and satisfies*

- *$|V_L| + |V_R| = \mathsf{poly}(n)$;*
- *G is biregular;*
- *$|\Sigma_L| = o(\log n)$ and $|\Sigma_R| = o(\log\log n)$;*
- *$\frac{|V_L|}{|V_L|+|V_R|} \geq 1 - \frac{1}{1+|\Sigma_{L|e}|/|\Sigma_R|}$ for every $e \in E$ (in fact, $|\Sigma_{L|e}|$ does not depend on e).*

The proof will be given in Section 4.1.

4.1 NP-hardness of Balanced-Label-Cover

First, we introduce some notions required for the proof of the NP-hardness.

Theorem 4.5 ([cf. Hås01, Theorem 2.24]). *There exists a constant $\varepsilon > 0$ such that it is* NP*-hard to distinguish, for a given 3-SAT instance satisfying*

- *each clause contains exactly three distinct variables as literals;*
- *each variable appears in clauses exactly five times,*

the following two cases: (i) the given instance is satisfiable and (ii) at most $1-\varepsilon$ fraction of clauses in the given instance is satisfiable.

We also review bipartite LDRCs.

Definition 4.6 ((Edge Reading) Bipartite LDRC [MR10]). *Let*

$$\langle x_{1,1}, \dots, x_{1,k} \rangle, \dots, \langle x_{N,1}, \dots, x_{N,k} \rangle \in [n]^k$$

be a list of k-tuples. A (Edge Reading) Bipartite LDRC for the k-tuples is

$$\mathcal{G} = ((A, B, E), \Sigma_A, \Sigma_B, \{\pi_e\}_{e \in E}, \{\mathrm{valid}_v\}_{v \in A}, \{\tau_e\}_{e \in E}, \{\rho_e\}_{e \in E}),$$

where

- *$G := (A, B, E)$ is a reguler bipartite graph;*
- *$\mathcal{G}' := (G, \Sigma_A, \Sigma_B, \{\pi'_e\}_{e \in E})$ is an instance of* LABEL-COVER*, where for each $e = (v, w) \in E$, $\pi'_e(a) = \pi_e(a)$ if $\mathrm{valid}(a) = 1$; otherwise (i.e., $\mathrm{valid}(a) = 0$), $\pi'_e(a)$ is undefined;*
- *every edge $e \in E$ carries a k-tuple $\tau_e \in [n]^k$ from the list and an evaluation function $\rho_e \colon \Sigma_A \to \{0,1\}^k$;*
- *for each $j \in [N]$, the tuple $\langle x_{j,1}, \dots, x_{j,k} \rangle$ appears on the same number of edges.*

Given labeling $f \colon A \to \Sigma_A$ and $g \colon B \to \Sigma_B$ to the vertices (A, B), an edge e is said to be "satisfied" if it is satisfied in $\mathcal{G}'$. For a massage $x \in \{0,1\}^n$, the edge $e = (a, b)$ is said to "decode" x if $\rho_e(f(a)) = (x_{i_1}, \dots, x_{i_k})$, where $\tau_e = (i_1, \dots, i_k)$.

Let $\delta_0 \in (0, 1)$, and let $L \colon (0, 1) \to \mathbb{R}_{\geq 0}$ be a decreasing function. The LDRC is said to be (δ_0, L)-bipartite LDRC if it satisfies the following conditions:

- *(Completeness) For every $x \in \{0,1\}^n$, one can efficiently compute assignments $f \colon A \to \Sigma_A$ and $g \colon B \to \Sigma_B$ such that all edges are satisfied and decode x.*
- *(Soundness) For every right assignment $g \colon B \to \Sigma_B$, every $\delta \in [\delta_0, 1)$, there exists $l \leq L(\delta)$ messages $x_1, \dots, x_l \in \{0,1\}^n$ such that the following holds for any left assignment $f \colon A \to \Sigma_A$: when picking uniformly at random an edge $e \in E$, the probability that e is satisfied but does not encode any one of $x_1, \dots, x_l$ is at most $O(\delta)$.*

Theorem 4.7 ([MR10, Theorem 14]). *There exists a constant $\alpha \in (0, 1/2)$ such that the following holds: Let $k := k(n) \leq (\log n)^\alpha$, let $\varepsilon := \varepsilon(n)$ with $(\log n)^{-\alpha} \leq \varepsilon < 1$. Then, there is an efficient algorithm that given a collection of size N of distinct k-tuples from a set $[n]$, outputs a $(\varepsilon, L(\delta) = \delta^{-O(1)})$-bipartite LDRC for these tuples. Moreover, its size is $(N + n) \cdot n^{o(1)}$, the alphabets satisfy $\log |\Sigma_A| \leq k \cdot \mathsf{poly}(1/\varepsilon)$ and $\log |\Sigma_B| \leq \log(1/\varepsilon)$, the degree of the A (i.e., left) vertices is $\varepsilon^{-O(k)}$, and the degree of the B (i.e., right) vertices is $\varepsilon^{-O(1)}$.*

We can further observe the additional property for the construction of bipartite LCRC proposed in [MR10].

Lemma 4.8. *Let $\mathcal{G} = ((A, B, E), \Sigma_A, \Sigma_B, \{\pi_e\}_{e\in E}, \{\mathrm{valid}_v\}_{v\in A}, \{\tau_e\}_{e\in E}, \{\rho_e\}_{e\in E})$ be the constructed bipartite LDRC as in Theorem 4.7. Then, for every $e = (v, w) \in E$, every massage $x \in \{0,1\}^n$, and every $b \in \Sigma_B$, the number of $a \in \Sigma_A$ such that (a, b) satisfies e (i.e., $\mathrm{valid}_v(a) = 1$ and $\pi_e(a) = b$) and make e decode x (i.e., $\rho_e(a) = (x_{i_1} \ldots, x_{i_k})$, where $\tau_e = (i_1, \ldots, i_k)$) is the same (regardless of e, x, and b).*

We observe the lemma above in Appendix A. Now, we show the NP-hardness of *Balanced*-LABEL-COVER based on the query reduction technique presented in [MR10].

Proof of Lemma 4.4. Let $\varepsilon\colon \mathbb{N} \to (0, 1]$ be an arbitrary function satisfying $\varepsilon(n) \geq 1/(\log\log\log n)^{O(1)}$. Let $L(\cdot)$ be the function in Theorem 4.7 when it is applied for $k(n) = (\log\log\log n)^{\log\log\log n}$ and $\varepsilon(n)$. We define $\delta_0(n) := \varepsilon(n)/L(\varepsilon(n)) \geq \mathsf{poly}(\varepsilon(n)) \geq 1/(\log\log\log n)^{O(1)}$.

By Theorem 4.5, we can map a 3-SAT instance ϕ of instance size n to another 3-SAT instance $\varphi := \varphi_\phi = (C_1, \ldots, C_m)$, where each C_i represents a clause on exact 3 distinct variables, each variable appears in clauses exactly 5 times, and the instance size is polynomially bounded in n. Without loss of generality, we assume $m \geq \log n$; otherwise, $\mathsf{P} = \mathsf{NP}$ and we can construct a trivial reduction that solves the given instance and maps it to a canonical instance according to the answer. Let ε_0 be the constant in Theorem 4.5, i.e., if the original instance is unsatisfiable, then for every assignment a to φ, at least ε_0-fraction of clauses are unsatisfied by a.

We invokes the pairwise-independent hitter H^{pw} on $\ell = \log m$, ε_0, and $\delta_0(n)/2$ and construct a new PCP system as follows: Its proof is an assignment a to φ. The PCP verifier V uses a $2\ell = O(\log n)$-bit random string r to obtain $i_1, \ldots, i_q \in \{0,1\}^\ell$ from $H^{pw}(r, 1^{\varepsilon_0^{-1}}, 1^{2\delta_0(n)^{-1}})$. Let us identify each i_j as the corresponding index in $[m]$ (recall that $\ell = \log m$). Then, V locally checks whether a satisfies

$$\begin{cases} C_{i_1} \wedge \cdots \wedge C_{i_q} & \text{if all of variables in } C_{i_1}, \ldots, C_{i_q} \text{ are distinct} \\ C_{k_1} \wedge \cdots \wedge C_{k_q} & \text{otherwise} \end{cases}$$

by querying $3q$ points on a, where $k_1, \ldots, k_q \in [m]$ are arbitrarily chosen indices so that all of variables in $C_{k_1}, \ldots, C_{k_q}$ are distinct. Namely, all constraints in the new PCP system perform 3-CNF formula of q clauses over $3q$ distinct variables, and thus, the number of assignment satisfying each CNF-constraint is the same. We further claim that this is the $3q$-query PCP system for 3-SAT as follows:

Claim 4.9. *If the given 3-SAT instance ϕ is satisfiable, then there exists a proof that V accepts with probability 1. By contrast, if ϕ is unsatisfiable, then V rejects any proof with probability at least $1 - \delta_0(n)$.*

We defer the proof of Claim 4.9 and continue the proof.

For each random string $r \in \{0,1\}^{2\ell}$ to V, let $i_1^r, \ldots, i_{3q}^r \in [m]$ be the query points. We assume that $k(n) \geq 3q(n)$ since $q(n) = 1/\varepsilon_0^{-1}\delta_0(n)^{-1} = (\log\log\log n)^{O(1)}$ and $k(n) = (\log\log\log n)^{\omega(1)}$. We construct a (ε, L)-bipartite LDRC of Theorem 4.7 and Lemma 4.8 for the list $\{i_1^r, \ldots, i_{3q}^r\}_{r\in\{0,1\}^{2\ell}}$ of $k(n)$-tuples, where we add $k(n) - 3q(n)$ distinct indices different from $\{i_1^r, \ldots, i_{3q}^r\}$. Let

$$\mathcal{G} = ((A, B, E), \Sigma_A, \Sigma_B, \{\pi_e\}_{e\in E}, \{\mathrm{valid}_v\}_{v\in A}, \{\tau_e\}_{e\in E}, \{\rho_e\}_{e\in E})$$

be the constructed bipartite LDRC. By Theorem 4.7, we have

$$O(|\Sigma_A|) = O(2^{k\mathsf{poly}(\varepsilon(n)^{-1})}) = 2^{(\log\log\log n)^{O(\log\log\log n)}} = 2^{o(\log\log n)} = o(\log n).$$

We construct a LABEL-COVER instance $\mathcal{G}'$ as

$$\mathcal{G}' = \left((A, B, E), \Sigma_A, \Sigma_B, \{\pi'_e\}_{e\in E}\right),$$

where, for each $e = (v, w) \in E$, we define the partial function $\pi'_e \colon \Sigma_A \to \Sigma_B$ so that the domain $\Sigma_{A|e}$ of π'_e is

$$\Sigma_{A|e} := \{a \in \Sigma_A : \mathrm{valid}_v(a) = 1 \text{ and } \rho_e(a) \text{ is verified by } V \text{ on randomness } r\},$$

where r is a random string such that $\tau_e = (i^r_1, \ldots, i^r_{3q})$, and $\pi'_e(a) = \pi_e(a)$ for each $a \in \Sigma_{A|e}$.

Note that (A, B, E) is biregular by Definition 4.6 and Theorem 4.7. In addition, we can observe that $\mathcal{G}'$ is balanced (i.e., $\mathcal{G}'$ is an instance of Balanced-LABEL-COVER) as follows: Since the number of assignment satisfying each CNF-constraint of φ is the same, the number of accepted massages decoded by each ρ_e is the same. By Lemma 4.8, for each edge e, each assignment α satisfying a CNF-constraint, and each $b \in \Sigma_B$, the number of A-alphabets a that satisfy π'_e and $\rho_e(a) = \alpha$ is the same. Thus, all the constraints are balanced. Furthermore, $\Sigma_{A|e} = \Sigma_{A|e'}$ for any pair (e, e') of edges; otherwise, there must exist two messages that violate Lemma 4.8.

We define $k := k(n)$ as the minimum value satisfying

$$k \cdot |A| \geq \left(1 + \frac{|\Sigma_A|}{|\Sigma_B|}\right) \cdot |B|.$$

Note that $k(n) = O(|\Sigma_A||B|) \leq \mathsf{poly}(n)$. Moreover, it is easy to observe that

$$\begin{aligned} k|A| - \left(1 - \frac{1}{1 + |\Sigma_A|/|\Sigma_B|}\right) k|A| &= \frac{1}{1 + |\Sigma_A|/|\Sigma_B|} \cdot k|A| \\ &\geq |B| \\ &\geq \left(1 - \frac{1}{1 + |\Sigma_A|/|\Sigma_B|}\right) |B|. \end{aligned}$$

Thus, for each $e \in E$,

$$\frac{k|A|}{k|A| + |B|} \geq 1 - \frac{1}{1 + |\Sigma_A|/|\Sigma_B|} \geq 1 - \frac{1}{1 + |\Sigma_{A|e}|/|\Sigma_B|}.$$

Finally, we construct the Balanced-LABEL-COVER instance $\mathcal{G}''$ satisfying the properties of Lemma 4.4 by duplicating the left vertices of $\mathcal{G}'$. In particular, we copy A vertices k times as $A^{(1)}, \ldots, A^{(k)}$, and copy the edge set E as $E^{(1)}, \ldots, E^{(k)}$, where $(a^{(i)}, b) \in E^{(i)}$ iff $(a, b) \in E$ for each i, $a^{(i)} \in A^{(i)}$ copied from $a \in A$, and $b \in B$. Let $A'' = \cup_i A^{(i)}$ and $E'' = \cup_i E^{(i)}$. We also copy the constraints $\{\pi'_e\}_{e\in E}$ in the same manner and obtain $\{\pi''_e\}_{e\in E''}$. Then we construct $\mathcal{G}'$ as

$$\mathcal{G}' = \left((A'', B, E''), \Sigma_A, \Sigma_B, \{\pi''_e\}_{e\in E''}\right).$$

Now we verify each property.

Completeness. If the original instance ϕ is satisfiable, then so is φ by Claim 4.9. In this case, by following the assignment for the bipartite LDRC that encodes the satisfying assignment to φ, it is easy to observe that all constraints are satisfied by the completeness of the bipartite LDRC.

Soundness. We verify the soundness for $\mathcal{G}'$, i.e., there is no assignment to A and B that satisfies greater than $C\varepsilon(n)$-fraction of edge constraints in $\mathcal{G}'$ for some absolute constant C. This implies the same soundness for $\mathcal{G}''$ because if there exists such an assignments to $A'' = \cup_i A^{(i)}$ and B, then

it must satisfy greater than $C\varepsilon(n)$-fraction of edge constraints in $(A^{(i)}, B, E^{(i)})$ for some i, which is identical to $\mathcal{G}'$.

We observe the soundness for $\mathcal{G}'$, which follows from the same argument as the proof of [MR10, Theorem 15]. Suppose that the original instance ϕ is unsatisfiable. Then, the PCP verifier V accept any proof (i.e., the assignment to φ) with probability at most $\delta_0(n) = \varepsilon(n)/L(\varepsilon(n))$ by Claim 4.9. Consider arbitrary assignments to A and B. By the soundness of the bipartite LDRC $\mathcal{G}$, there exists at most $l \leq L(\varepsilon(n))$ proofs $\pi_1, \ldots, \pi_l$ such that the probability that the edge $e \in E$ is satisfied in $\mathcal{G}$, but does not decode any of $\pi_1, \ldots, \pi_l$, is at most $O(\varepsilon(n))$. Notice that each proof π_i can satisfy at most $\varepsilon(n)/L(\varepsilon(n))$-fraction of constraints in the PCP system. Thus, the fraction of constraints (of the PCP system) satisfied by *some* proof in $\pi_1, \ldots, \pi_l$ is at most $L(\varepsilon(n)) \cdot \varepsilon(n)/L(\varepsilon(n)) = \varepsilon(n)$. Therefore, the fraction of the constraints in $\mathcal{G}'$ is at most $O(\varepsilon(n)) + \varepsilon(n) \leq C \cdot \varepsilon(n)$ for some absolute constant C.

Balanced constraints. It holds since all constraint functions in $\mathcal{G}'$ are balanced, and $\mathcal{G}''$ is constructed just by duplicating vertices and constraints of $\mathcal{G}'$.

The first item. It is verified as follows:

$$|A''| + |B| \leq k(n) \cdot (|A| + |B|) \leq \mathsf{poly}(n) \cdot (2^{2\ell} + m) \cdot m^{o(1)} \leq \mathsf{poly}(n).$$

The second item. (A'', B, E'') is biregular since (A, B, E) is biregular.

The third item. We have already shown that $|\Sigma_A| = o(\log n)$. By Theorem 4.7, we also have

$$|\Sigma_B| \leq 1/\varepsilon(n) = o(\log \log n).$$

The fourth item. For every $e \in E''$,

$$\frac{|A''|}{|A''| + |B|} = \frac{k|A|}{k|A| + |B|} \geq 1 - \frac{1}{1 + |\Sigma_{A|e}|/|\Sigma_B|}.$$

We complete the proof by showing Claim 4.9.

Proof of Claim 4.9. It is easy to verify the completeness since $\varphi := \varphi_\phi$ is satisfiable when ϕ is satisfiable. Thus, it suffices to show the soundness.

Suppose that ϕ is unsatisfiable. Fix an assignment a to φ arbitrarily. Then, there exists a subset S_a of m clauses in φ such that $|S_a| \geq \varepsilon_0 \cdot m$ and $C(a) = 0$ (i.e., unsatisfied) for all $C \in S_a$. Recall that the PCP verifier V uses its randomness r to select q indices $i_1, \ldots, i_q \in [m]$ according to $H^{pw}(r, 1^{\varepsilon_0^{-1}}, 1^{2\delta_0(n)^{-1}})$ and checks whether $C_{i_1} \wedge \cdots \wedge C_{i_q}$ is satisfied if no variable is shared among $C_{i_1}, \ldots, C_{i_q}$. Namely it suffices to show that

$$\Pr_r \left[\text{all variables in } C_{i_1} \wedge \cdots \wedge C_{i_q} \text{ are distinct} \wedge \exists j \in [q] \text{ s.t. } C_{i_j} \in S_a\right] \geq 1 - \delta_0(n),$$

where $(i_1, \ldots, i_q) \leftarrow H^{pw}(r, 1^{\varepsilon_0^{-1}}, 1^{2\delta_0(n)^{-1}})$. Let E denote the event above.

By the property of the hitter and the fact that $|S_a| \geq \varepsilon_0 \cdot m$, it holds that

$$\Pr_r \left[\forall j \in [q]\ C_{i_j} \notin S_a\right] \leq \delta_0(n)/2.$$

In addition, by the pairwise independence of H^{pw}, we have

$$\begin{aligned}
&\Pr_r \left[\exists j, j' \in [q] \text{ s.t. } j \neq j' \text{ and } C_{i_j} \text{ and } C_{i_{j'}} \text{ share a variable}\right] \\
&\quad \leq q^2 \cdot \Pr_{i,i' \sim [m]}[C_i \text{ and } C_{i'} \text{ share a variable}] \\
&\quad \leq q^2 \cdot \frac{3^2}{m} \leq q^2 \cdot \frac{9}{\log n} \leq (2\varepsilon_0\delta_0(n))^{-2} \frac{9}{\log n} \leq \frac{(\log \log \log n)^{O(1)}}{\log n} = o\left(\frac{1}{\log \log n}\right),
\end{aligned}$$

where the first inequality follows from the union bound and the pairwise independence, and the second inequality holds by the union bound (over the 3^2 pairs of variables in C_i and $C_{i'}$) and Theorem 4.5, i.e., the fact that all the variable appear in the exactly same number of clauses in φ.

Thus, for all large enough $n \in \mathbb{N}$,

$$\Pr_r\left[\exists j, j' \in [q] \text{ s.t. } j \neq j' \text{ and } C_{i_j} \text{ and } C_{i_{j'}} \text{ share a variable}\right] \leq \delta_0(n)/2,$$

and by the union bound,

$$\Pr[\neg E] \leq \delta_0(n)/2 + \delta_0(n)/2 = \delta_0(n).$$

Thus, we obtain that $\Pr[E] \geq 1 - \delta_0(n)$, which completes the proof. ◇

□

5 NP-Hardness of Distributional Minimum Monotone Satisfying Assignment

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

5.1 Distributional Minimum Monotone Satisfying Assignment

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

Definition 5.1 (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 $n, s \in \mathbb{N}$ 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 the weight function w satisfies $w(x) \geq 1/\mathsf{poly}(n)$, where n is the number of variables, for each $x \in X$, and w is represented in polynomial size in n.

The main theorem in this section is stated as follows.

Theorem 5.2. *There exist functions $\ell := \ell(n) = \omega(1)$ and $\gamma := \gamma(n) = o(1)$ such that solving 3SAT of size n is reducible to $\mathrm{Gap}_{\ell/\mathsf{polylog}\ell}^{\mathsf{negl}(\ell),\gamma}\mathsf{F}[\ell]$-DMMSA 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 5.3. $\mathrm{Gap}_{\sigma}^{\varepsilon,\gamma}\mathfrak{C}$-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 $m = \Theta(p(m)^2 \cdot n)$ 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 $\mathrm{Gap}_{\sigma}^{\varepsilon+1/p(n),\gamma+1/p(n)}\mathfrak{C}$-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

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

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 satisfying assignment such that

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

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

5.2 Block $\vee$-Noise Operator

In this section, we introduce some additional notions on the analysis of Boolean functions for the proof of Theorem 5.2.

Definition 5.4 (Block $\vee$-noise operator)**.** *For* $n \in \mathbb{N}$*, let* $\mathcal{P} = (S_1, \ldots, S_m)$ *be a disjoint set system of* $[n]$*, i.e.,* $S_i \subseteq [n]$ *and* $S_i \cap S_j = \emptyset$ *for every* $i, j \in [m]$ *with* $i \neq j$*. For* $\rho, \mu \in [0, 1]$*, we define* $\vee\text{-}\mathrm{N}_{\rho}^{\mathcal{P},\mu}$ *as a mapping from* $x \in \{0,1\}^n$ *to a distribution over* $\{0,1\}^n$ *sampled according to the following procedure: (i) select* $b_j \sim \mathrm{Ber}(\rho)$ *for each* $j \in [m]$*, (ii) sample* $y \in \{0,1\}^n$*, where each* y_i *is selected as*

$$\begin{cases} y_i = x_i & \text{if } i \in S_j \text{ and } b_j = 1 \\ y_i \sim \mathrm{Ber}(\mu'_j) & \text{if } i \in S_j \text{ and } b_j = 0 \\ 0 & \text{otherwise } (i.e.,\ i \notin \cup_j S_j), \end{cases}$$

where $\mu'_j := 1 - (1 - \mu)^{1/|S_j|}$*, i.e.,* $1 - (1 - \mu'_j)^{|S_j|} = \mu$*.*

We define the μ*-biased block* $\vee$*-noise operator* $\vee\text{-}\mathrm{T}_{\rho}^{\mathcal{P},\mu}$ *(with parameter* ρ*) as the linear operator on* $f\colon \{0,1\}^m \to \mathbb{R}$ *defined as*

$$\vee\text{-}\mathrm{T}_{\rho}^{\mathcal{P},\mu} f(x) := \mathbb{E}_{y\sim\vee\text{-}\mathrm{N}_{\rho}^{\mathcal{P},\mu}(x)}\left[f(y)\right].$$

Any partial function $\pi\colon [n] \to [m]$, where $n, m \in \mathbb{N}$, determines a set system $(\pi^{-1}(1), \ldots, \pi^{-1}(m))$ over $[n]$, and we abuse the notation π^{-1} to represent the set system induced by π. Note that π^{-1} is disjoint whenever π is surjective.

We introduce a notation $\vee \cdot \pi$.

Definition 5.5. *For any $n, m \in \mathbb{N}$, any string $x \in \{0,1\}^n$, and any partial function $\pi\colon [n] \to [m]$, we define $\vee \cdot \pi\colon \{0,1\}^n \to \{0,1\}^m$ as the mapping defined as for each $x \in \{0,1\}^n$ and each $j \in [m]$,*

$$\vee \cdot \pi(x)_j := \bigvee_{i \in \pi^{-1}(j)} x_i.$$

The following proposition is derived immediately from the definitions.

Proposition 5.6. *For $n, m \in \mathbb{N}$, a partial and surjective function $\pi\colon [n] \to [m]$, and $\tau, \rho \in [0,1]$, it holds that the distribution of $\vee \cdot \pi(\vee\text{-}\mathrm{N}_\rho^{\pi^{-1},\mu}(x))$ is statistically equivalent to $\mathrm{N}_\rho^\mu(\vee \cdot \pi(x))$ for all $x \in \{0,1\}^n$.*

Proof. In both cases where (i) $y \sim \vee \cdot \pi(\vee\text{-}\mathrm{N}_\rho^{\pi^{-1},\mu}(x))$ and (ii) $y \sim \mathrm{N}_\rho^\mu(\vee \cdot \pi(x))$, it is easily checked that the probability that $y_i = 1$ is exactly

$$\begin{cases} \rho + (1-\rho)\mu & \text{if } \vee \cdot \pi(x)_i = 1 \\ (1-\rho)\mu & \text{if } \vee \cdot \pi(x)_i = 0 \end{cases}$$

for each $i \in [m]$. □

Now we show the following key lemma.

Lemma 5.7. *For every $n, m \in \mathbb{N}$ with $n > m$, every partial function $\pi\colon [n] \to [m]$, every $\rho, \mu \in [0,1]$, every function $f\colon \{0,1\}^m \to \mathbb{R}$, and every $i \in [n]$ in the domain of π, if π is balanced, then*

$$\mathrm{Inf}_i^{\mu'}\left(\vee\text{-}\mathrm{T}_\rho^{\pi^{-1},\mu}\left(f \circ (\vee \cdot \pi)\right)\right) \le \mathrm{Inf}_{\pi(i)}^{\mu}\left(\mathrm{T}_\rho^\mu(f)\right),$$

where $\mu' := 1 - (1-\mu)^{m'/n}$, i.e., $1 - (1-\mu')^{n/m'} = \mu$ for $m' := |\{i \in [n] : \pi(i) \text{ is defined}\}|$.

Proof. For readability, we may omit the subscript and superscript from $\vee\text{-}\mathrm{T}_\rho^{\pi^{-1},\mu}$, T_ρ^μ, $\vee\text{-}\mathrm{N}_\rho^{\pi^{-1},\mu}$, and N_ρ^μ below.

For each $j \in [n]$, let x_j be a random variable selected according to $\mathrm{Ber}(\mu')$. Let $x = x_1 \circ \cdots \circ x_n$ and $x_{[n]\setminus i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. Then, we have that

$$\begin{aligned}
\mathrm{Inf}_i^{\mu'}\left(\vee\text{-}\mathrm{T}_\rho^{\pi^{-1},\mu}\left(f \circ (\vee \cdot \pi)\right)\right) &= \mathbb{E}_{x_{[n]\setminus i}}\left[\mathbb{E}_{x_i}\left[\vee\text{-}\mathrm{T}\left(f \circ (\vee \cdot \pi)\right)(x)^2\right] - \left(\mathbb{E}_{x_i}\left[\vee\text{-}\mathrm{T}\left(f \circ (\vee \cdot \pi)\right)(x)\right]\right)^2\right] \\
&= \mathbb{E}_x\left[\left(\mathbb{E}_{\bar{x}\sim\vee\text{-}\mathrm{N}(x)}\left[f(\vee \cdot \pi(\bar{x}))\right]\right)^2\right] - \mathbb{E}_{x_{[n]\setminus i}}\left[\left(\mathbb{E}_{x_i}\mathbb{E}_{\bar{x}\sim\vee\text{-}\mathrm{N}(x)}\left[f(\vee \cdot \pi(\bar{x}))\right]\right)^2\right] \\
&\le \mathbb{E}_x\left[\mathbb{E}_{\bar{x}\sim\vee\text{-}\mathrm{N}(x)}\left[f(\vee \cdot \pi(\bar{x}))^2\right]\right] - \mathbb{E}_{x_{[n]\setminus i}}\left[\left(\mathbb{E}_{x_i}\mathbb{E}_{\bar{x}\sim\vee\text{-}\mathrm{N}(x)}\left[f(\vee \cdot \pi(\bar{x}))\right]\right)^2\right] \\
&= \mathbb{E}_x\left[f(\vee \cdot \pi(x))^2\right] - \mathbb{E}_{x_{[n]\setminus i}}\left[\left(\mathbb{E}_{x_i}\mathbb{E}_{\bar{x}\sim\vee\text{-}\mathrm{N}(x)}\left[f(\vee \cdot \pi(\bar{x}))\right]\right)^2\right],
\end{aligned}$$

where the inequality follows from Jensen's inequality.

For each $j \in [m]$, let y_j be a random variable selected according to $\mathrm{Ber}(\mu)$. Let $y = y_1 \circ \cdots \circ y_n$ and $y_{[m]\setminus\pi(i)} = (y_1, \ldots, y_{\pi(i)-1}, y_{\pi(i)+1}, \ldots, y_m)$. Then, we have that

$$\begin{aligned}
\mathrm{Inf}_{\pi(i)}^{\mu}\left(\mathrm{T}_\rho^\mu(f)\right) &= \mathbb{E}_{y_{[m]\setminus\pi(i)}}\left[\mathbb{E}_{y_{\pi(i)}}\left[\mathrm{T}\left(f\right)(y)^2\right] - \left(\mathbb{E}_{y_{\pi(i)}}\left[\mathrm{T}\left(f\right)(y)\right]\right)^2\right] \\
&= \mathbb{E}_y\left[\mathbb{E}_{\bar{y}\sim\mathrm{N}(y)}\left[f(\bar{y})^2\right]\right] - \mathbb{E}_{y_{[m]\setminus\pi(i)}}\left[\left(\mathbb{E}_{y_{\pi(i)}}\mathbb{E}_{\bar{y}\sim\mathrm{N}(y)}\left[\mathrm{T}\left(f\right)(\bar{y})\right]\right)^2\right] \\
&= \mathbb{E}_y\left[f(y)^2\right] - \mathbb{E}_{y_{[m]\setminus\pi(i)}}\left[\left(\mathbb{E}_{y_{\pi(i)}}\mathbb{E}_{\bar{y}\sim\mathrm{N}(y)}\left[f(\bar{y})\right]\right)^2\right].
\end{aligned}$$

Since $1-(1-\mu')^{n/m'}=\mu$, it holds that $\Pr_x[(\vee\cdot\pi(x))_j=1]=\mu$ for each $j\in[m]$. Thus,

$$\mathbb{E}_x\left[f(\vee\cdot\pi(x))^2\right]=\mathbb{E}_y\left[f(y)^2\right].$$

Therefore,

$$\begin{aligned}
&\mathrm{Inf}^{\mu}_{\pi(i)}\left(\mathrm{T}^{\mu}_{\rho}(f)\right)-\mathrm{Inf}^{\mu'}_{i}\left(\vee\text{-}\mathrm{T}^{\pi^{-1},\mu}_{\rho}\left(f\circ(\vee\cdot\pi)\right)\right)\\
&\quad\geq\mathbb{E}_{x_{[n]\setminus i}}\left[\left(\mathbb{E}_{x_i}\mathbb{E}_{\bar{x}\sim\vee\text{-}\mathrm{N}(x)}\left[f(\vee\cdot\pi(\bar{x}))\right]\right)^2\right]-\mathbb{E}_{y_{[m]\setminus\pi(i)}}\left[\left(\mathbb{E}_{y_{\pi(i)}}\mathbb{E}_{\bar{y}\sim\mathrm{N}(y)}\left[f(\bar{y})\right]\right)^2\right]\\
&\quad=\mathbb{E}_{x_{[n]\setminus i}}\left[\left(\mathbb{E}_{x_i}\mathbb{E}_{\bar{y}\sim\mathrm{N}(\vee\cdot\pi(x))}\left[f(\bar{y})\right]\right)^2\right]-\mathbb{E}_{y_{[m]\setminus\pi(i)}}\left[\left(\mathbb{E}_{y_{\pi(i)}}\mathbb{E}_{\bar{y}\sim\mathrm{N}(y)}\left[f(\bar{y})\right]\right)^2\right],
\end{aligned}$$

where the last equality follows from Proposition 5.6.

Namely, it suffices to show that

$$\mathbb{E}_{x_{[n]\setminus i}}\left[\left(\mathbb{E}_{x_i}\mathbb{E}_{\bar{y}\sim\mathrm{N}(\vee\cdot\pi(x))}\left[f(\bar{y})\right]\right)^2\right]\geq\mathbb{E}_{y_{[m]\setminus\pi(i)}}\left[\left(\mathbb{E}_{y_{\pi(i)}}\mathbb{E}_{\bar{y}\sim\mathrm{N}(y)}\left[\mathrm{T}\left(f\right)(\bar{y})\right]\right)^2\right].$$

We introduce some notations. Without loss of generality, we assume that $\pi(i)=m$. Let $\{i,i_1,\dots,i_{k-1}\}=\pi^{-1}(m)$, where $k=n/m'$ since π is balanced. Let $x_{\pi^{-1}(m)\setminus i}=(x_{i_1},\dots,x_{i_{k-1}})$. Then, the expression above is observed as follows:

$$\begin{aligned}
\mathbb{E}_{x_{[n]\setminus i}}\left[\left(\mathbb{E}_{x_i}\mathbb{E}_{\bar{y}\sim\mathrm{N}(\vee\cdot\pi(x))}\left[f(\bar{y})\right]\right)^2\right]&=\mathbb{E}_{y_{[m]\setminus\pi(i)},x_{\pi^{-1}(m)\setminus i}}\left[\left(\mathbb{E}_{x_i}\mathbb{E}_{\bar{y}\sim\mathrm{N}\left(y_{[m]\setminus\pi(i)}\circ\left(x_i\vee x_{i_1}\vee\cdots\vee x_{i_{k-1}}\right)\right)}\left[f(\bar{y})\right]\right)^2\right]\\
&\geq\mathbb{E}_{y_{[m]\setminus\pi(i)}}\left[\left(\mathbb{E}_{x_{\pi^{-1}(m)\setminus i}}\mathbb{E}_{x_i}\mathbb{E}_{\bar{y}\sim\mathrm{N}\left(y_{[m]\setminus\pi(i)}\circ\left(x_i\vee x_{i_1}\vee\cdots\vee x_{i_{k-1}}\right)\right)}\left[f(\bar{y})\right]\right)^2\right]\\
&=\mathbb{E}_{y_{[m]\setminus\pi(i)}}\left[\left(\mathbb{E}_{y_{\pi(i)}}\mathbb{E}_{\bar{y}\sim\mathrm{N}(y)}\left[f(\bar{y})\right]\right)^2\right],
\end{aligned}$$

where the inequality holds since $\mathbb{E}[X^2]\geq\mathbb{E}[X]^2$ for any random variable X. □

5.3 Proof of Theorem 5.2

We present the reduction from 3SAT to GapDMMSA and show Theorem 5.2. Let n be the instance size of the original SAT instance. We first apply the reduction in Lemma 4.4 and obtain a BALANCED-LABEL-COVER instance satisfying the conditions of the lemma. Let $G=(V_L,V_R,E)$, (Σ_L,Σ_R), and $\{\pi_e\}_{e\in E}$ be a constraint biregular graph, alphabet sets, and (balanced) constraints, respectively.

Now, we describe the DMMSA instance $\mathcal{D}$, where we use the following parameters:

- $k(n)=\omega(1)$: a sufficiently small super constant we specify later.
- $\sigma=|\Sigma_{L|e}|/|\Sigma_R|$ for $e\in E$ (note that σ does not depend on the choice of e).
- $\mu_L=1/(2k(n))$: a bias parameter for left codes.
- $\mu_R=1-(1-\mu_L)^{\sigma}$: a bias parameter for right codes.
- $d=(\log k(n))^2$: a duplicate parameter.
- $\rho=1/(d^3\log(4d))$: a correlation parameter.

- $\tau = 1/(2d)$: a noise parameter.

The set of binary variables is

$$X = \{x_{v,S}\}_{v\in V_L, S\in\{0,1\}^{|\Sigma_L|}} \cup \{y_{w,T}\}_{w\in V_R, T\in\{0,1\}^{|\Sigma_R|}},$$

and the weight function w is defined as

$$w(z) = \begin{cases} \frac{1}{|V_L|+|V_R|} \cdot \mu_L^{\mathsf{wt}(S)}(1-\mu_L)^{|\Sigma_L|-\mathsf{wt}(S)} & \text{if } z = x_{v,S} \\ \frac{1}{|V_L|+|V_R|} \cdot \mu_R^{\mathsf{wt}(T)}(1-\mu_R)^{|\Sigma_R|-\mathsf{wt}(T)} & \text{if } z = y_{w,T}. \end{cases}$$

Then, it is easily verified that

$$\begin{aligned} \sum_{z\in X} w(z) &= \sum_{v\in V_L} \sum_{S\in\{0,1\}^{|\Sigma_L|}} \frac{1}{|V_L|+|V_R|} \cdot \mu_L^{\mathsf{wt}(S)}(1-\mu_L)^{|\Sigma_L|-\mathsf{wt}(S)} \\ &\qquad + \sum_{w\in V_R} \sum_{T\in\{0,1\}^{|\Sigma_R|}} \frac{1}{|V_L|+|V_R|} \cdot \mu_R^{\mathsf{wt}(T)}(1-\mu_R)^{|\Sigma_R|-\mathsf{wt}(T)} \\ &= \sum_{v\in V_L} \frac{1}{|V_L|+|V_R|} + \sum_{v\in V_R} \frac{1}{|V_L|+|V_R|} = 1. \end{aligned}$$

Below we may regard $S \in \{0,1\}^{|\Sigma_L|}$ (resp. $T \in \{0,1\}^{|\Sigma_R|}$) as a subset $S \subseteq \Sigma_L$ (resp. $T \subseteq \Sigma_R$) interchangeably.

Procedure 1: a DMMSA instance $\mathcal{D}$ over X

Output: a DNF formula φ over X
Let $M := 2\mu_L^{-1}\rho^{-1}\log^2 k(n)$;
Select $e = (v, w) \sim E$;
Initialize φ as an empty formula;
repeat the following M times
  Select $S \in \{0,1\}^{|\Sigma_L|}$ according to $\mu_L^{\otimes|\Sigma_L|}$;
  Select $(T_1, \ldots, T_{d-1}) \in (\{0,1\}^{|\Sigma_L|})^{d-1}$ according to $\tilde{\mathrm{N}}_\rho^{\otimes(d-1)}(S)$;
  Add noise as $\bar{S} \sim \mathrm{N}_{1-\tau}^{\mu_L}(S)$ and $T_i' \sim \vee\text{-}\mathrm{N}_{1-\tau}^{\pi_e^{-1},\mu_R}(T_i)$ for each $i \in [d-1]$;
  Let $\bar{T}_i = \vee \cdot \pi_e(T_i') \in \{0,1\}^{|\Sigma_R|}$ for each $i \in [d-1]$;
  Update $\varphi := \varphi \vee (x_{v,\bar{S}} \wedge y_{w,\bar{T}_1} \wedge \cdots \wedge y_{w,\bar{T}_{d-1}})$;
end;
return φ;

It is not hard to verify that the leaf size $\ell(n)$ of $\mathcal{D}$ is at most

$$\ell(n) \le M \cdot d = 2\mu_L^{-1}\rho^{-1}d \cdot \log^2 k(n) = k(n) \cdot \mathsf{polylog}k(n).$$

We prove the completeness and soundness, which are stated as the following lemmas.

Lemma 5.8. *If the* Balanced-LABEL-COVER *instance is satisfiable with the promise in Lemma 4.4, then* $\mathsf{val}_{2^{-(\log k(n))^2}}(\mathcal{D}) \le 1/k(n)$.

Lemma 5.9. *Let* $\gamma = 1/\log k(n)$, *and let* $\varepsilon(n) = o(1)$ *be the function in Lemma 4.4. If* $\mathsf{val}_{3\gamma}(\mathcal{D}) \le \gamma/8$ *and* $k(n)$ *is a small enough super-constant function, then at least* $\varepsilon(n)$ *of edges of the* Balanced-LABEL-COVER *instance are satisfiable under the promise in Lemma 4.4.*

Theorem 5.2 is immediately derived from Lemmas 4.4, 5.8 and 5.9. Now, we prove Lemmas 5.8 and 5.9 to complete the proof.

Proof of Lemma 5.8. Since the Balanced-LABEL-COVER instance is satisfiable, there exists an assignment $\alpha_L \colon V_L \to \Sigma_L$ and $\alpha_R \colon V_R \to \Sigma_R$ such that for all $e = (u, v) \in E$, it holds that $\pi_e(\alpha_L(v)) = \alpha_R(w)$. For each $v \in V_L$ (resp. $w \in V_R$), let $\alpha_v := \alpha_L(v)$ (resp. $\alpha_w := \alpha_R(w)$).

Recall that the instance of Lemma 4.4 satisfies

$$\frac{|V_L|}{|V_L| + |V_R|} \geq 1 - \frac{1}{1 + |\Sigma_{L|e}|/|\Sigma_R|} = \frac{\sigma}{1+\sigma}.$$

This implies that

$$(1+\sigma) \cdot \frac{|V_L|}{|V_L| + |V_R|} \geq \sigma,$$

and

$$\frac{|V_L|}{|V_L| + |V_R|} \geq \sigma \left(1 - \frac{|V_L|}{|V_L| + |V_R|}\right) = \frac{|V_R|}{|V_L| + |V_R|}\sigma. \tag{1}$$

We translate (α_L, α_R) to an assignment $\beta \colon X \to \{0,1\}$ for DMMSA as follows:

$$\beta(z) = \begin{cases} \mathbb{1}\{\alpha_v \in S\} & \text{if } z = x_{v,S} \\ \mathbb{1}\{\alpha_w \in T\} & \text{if } z = y_{w,T}, \end{cases}$$

where we also regard β as the element in $\{0,1\}^{|X|}$ by letting $\beta_z := \beta(z)$. Note that β is composed of the valid long codes for assignments $\{\alpha_v\}_{v \in V_L \cup V_R}$.

The weight of β is evaluated as

$$\begin{aligned}
w(\beta) &= \sum_{v \in V_L} \frac{1}{|V_L| + |V_R|} \Pr_{S \sim \mu_L^{\otimes |\Sigma_L|}}[\alpha_v \in S] + \sum_{w \in V_R} \frac{1}{|V_L| + |V_R|} \Pr_{T \sim \mu_R^{\otimes |\Sigma_R|}}[\alpha_w \in T] \\
&= \frac{|V_L|}{|V_L| + |V_R|}\mu_L + \frac{|V_R|}{|V_L| + |V_R|}\mu_R \\
&= \frac{|V_L|}{|V_L| + |V_R|}\mu_L + \frac{|V_R|}{|V_L| + |V_R|}(1 - (1-\mu_L)^\sigma) \\
&\leq \frac{|V_L|}{|V_L| + |V_R|}\mu_L + \frac{|V_R|}{|V_L| + |V_R|}\sigma\mu_L \\
&\leq \frac{|V_L|}{|V_L| + |V_R|}\mu_L + \frac{|V_L|}{|V_L| + |V_R|}\mu_L \\
&= \frac{|V_L|}{|V_L| + |V_R|} \cdot 2\mu_L \\
&\leq 2\mu_L \\
&= 1/k(n),
\end{aligned}$$

where the first inequality follows from the union bound, and the second inequality follows from Equation (1).

Thus, it suffices to show that $\varphi(\beta) = 1$ with probability at least $1 - 2^{-\log^2 k(n)}$ over the choice $\varphi \sim \mathcal{D}$.

In the execution of $\mathcal{D}$, suppose that $e = (v, w) \in E$ is selected. We consider each sequential choice of $S, \{T_i\}, \bar{S}, \{T'_i\}$, and $\{\bar{T}_i\}$. If all the following events occur:

- E_1: $\alpha_v \in S$;
- E_2: $S_{\alpha_v} = (T_i)_{\alpha_v}$ for all $i \in [d-1]$;
- E_3: $S_{\alpha_v} = \bar{S}_{\alpha_v}$ and $(T_i)_{\alpha_v} = (T'_i)_{\alpha_v}$ for all $i \in [d-1]$,

then $\alpha_v \in \bar{S}$ and $\alpha_v \in T'_i$ for all i; thus, $\bar{T}_i = \vee \cdot \pi_e(\alpha_v)$ must contain $\pi_e(\alpha_v) = \alpha_w$. Therefore, if $E_{[3]} := E_1 \wedge E_2 \wedge E_3$ occurs at some step, the DNF φ must contain a term $(x_{v,\bar{S}} \wedge y_{w,\bar{T}_1} \wedge \cdots \wedge y_{w,\bar{T}_{d-1}})$ and thus $\varphi(\beta) = 1$ since $\alpha_v \in \bar{S}$ and $\alpha_w \in \bar{T}_i$ for all i. Since $\mathcal{D}$ repeats the sequential choice M times, we have

$$\Pr_{\varphi \sim \mathcal{D}}[\varphi(\beta) \neq 1] \leq (1 - \Pr[E_{[3]}])^M.$$

It is easily observed that $\Pr[E_1] = \mu_L$ and $\Pr[E_2] \geq \rho$ because $S \sim \mu_L^{\otimes |\Sigma_L|}$ and by the definition of $\tilde{\mathrm{N}}_\rho$. By the definitions of $\mathrm{N}_{1-\tau}^{\mu_L}$ and $\vee\text{-}\mathrm{N}_{1-\tau}^{\pi_e^{-1},\mu_R}$ and the union bound, we have $\Pr[\neg E_3] \leq d \cdot \tau = 1/2$. Therefore,

$$\Pr[E_{[3]}] \geq \mu_L \cdot \rho \cdot (1/2).$$

Thus, we conclude that

$$\Pr_{\varphi \sim \mathcal{D}}[\varphi(\beta) \neq 1] \leq (1 - \Pr[E_{[3]}])^M \leq (1 - \mu_L \rho/2)^{2\mu_L^{-1}\rho^{-1}\log^2 k(n)} \leq 2^{-\log^2 k(n)}.$$

□

Next, we show the soundness part. We use the following lemma.

Lemma 5.10 (following from [KS15]; see also [GL22, Appendix C])**.** *For every $\mu \in (0, 1/2)$, every $d \in \mathbb{N}$ with $d \geq 2$, and every $\rho \in [0,1]$ with $\rho \leq 1/(cd^2 \log(4d))$, let $t := t(\mu, d, \rho)$ be*

$$t(\mu, d, \rho) = d^{-2} \cdot \left(\frac{1}{d \cdot 4^d}\right)^{c' d 4^d (1-\rho)^{-1} \log(\rho^{-1}\mu^{-d}) \log(d4^d)},$$

where $c, c' > 0$ are universal constant. Then, the following holds: For $m \in \mathbb{N}$, let $f\colon \{0,1\}^m \to [0,1]$ and $g\colon \{0,1\}^m \to [0,1]$ be functions satisfying that

$$\max\{\mathbb{E}_x[f(x)], \mathbb{E}_x[g(x)]\} \leq 1/4.$$

If f and g satisfy

$$\{i : \mathrm{Inf}_i^\mu(f) > t\} \cap \{i : \mathrm{Inf}_i^\mu(g) > t\} = \emptyset,$$

then

$$\mathbb{E}_{x^{(0)},\ldots x^{(d-1)}}\left[f(x^{(0)}) \cdot \prod_{i \in [d-1]} g(x^{(i)})\right] \leq 2^{-d},$$

where $x^{(0)} \sim \mathrm{Ber}(\mu)^{\otimes m}$ and $(x^{(1)}, \ldots, x^{(d-1)}) \sim \tilde{\mathrm{N}}_\rho^{\otimes(d-1)}(x^{(0)})$.

Lemma 5.10 is proved based on the same idea as [GL22], building upon the work [Mos10; KS15]. We present the proof of Lemma 5.10 in Section 5.4 for completeness.

In our setting, we use Lemma 5.10 for $\mu = \mu_L = 1/k(n)$, $d = \log^2 k(n)$, and $\rho = 1/(d^3 \log(4d)) = 1/\mathsf{polylog}(k(n))$, which are consistent with the parameters for the reduction. Then, the function t is strictly decreasing in $k(n)$, and $t = O(1) > \varepsilon(n)$ when $k(n) = O(1)$ and n is enough large (recall

that $\varepsilon(n)$ is a function in Lemma 4.4). Now, we fix $k(n)$ to be a super-constant function small enough so that

$$t^2 \cdot \tau^2 \cdot \gamma = \frac{t^2}{4 \log^5 k(n)} > \varepsilon(n).$$

In this setting, we show the soundness.

Proof of Lemma 5.9. Suppose that $\mathsf{val}_{3\gamma}(\mathcal{D}) \leq \gamma/8$. Then there exists an assignment α to X such that $w(\alpha) \leq \gamma/8$ and $\Pr_{\varphi \sim \mathcal{D}}[\varphi(\alpha) = 1] \geq 3\gamma$. We may regard α as a mapping from X to $\{0,1\}$.

Let $V = V_L \cup V_R$. For each $v \in V$, we define $w_\alpha(v) \in [0,1]$ as

$$w_\alpha(v) = \begin{cases} \sum_{S \subseteq \Sigma_L : \alpha(x_{v,S})=1} \mu_L^{|S|}(1-\mu_L)^{|\Sigma_L|-|S|} & \text{if } v \in V_L \\ \sum_{T \subseteq \Sigma_R : \alpha(y_{v,T})=1} \mu_R^{|T|}(1-\mu_R)^{|\Sigma_R|-|T|} & \text{if } v \in V_R. \end{cases}$$

Then, we can observe that

$$\begin{aligned} &\mathbb{E}_{v \sim V}\left[w_\alpha(v)\right] \\ &= \frac{1}{|V_L| + |V_R|} \left(\sum_{v \in V_L} \sum_{\substack{S \subseteq \Sigma_L : \\ \alpha(x_{v,S})=1}} \mu_L^{|S|}(1-\mu_L)^{|\Sigma_L|-|S|} + \sum_{w \in V_R} \sum_{\substack{T \subseteq \Sigma_R : \\ \alpha(y_{w,T})=1}} \mu_R^{|T|}(1-\mu_R)^{|\Sigma_R|-|T|} \right) \\ &= \sum_{v \in V_L} \sum_{\substack{S \subseteq \Sigma_L : \\ \alpha(x_{v,S})=1}} w(x_{v,S}) + \sum_{w \in V_R} \sum_{\substack{T \subseteq \Sigma_R : \\ \alpha(y_{w,T})=1}} w(y_{w,T}) \\ &= w(\alpha). \end{aligned}$$

Thus, $\mathbb{E}_v\left[w_\alpha(v)\right] \leq \gamma/8$, and by Markov's inequality,

$$\Pr_{v \sim V}\left[w_\alpha(v) \leq 1/4\right] \geq 1 - \gamma/2.$$

Let $G_V := \{v \in V : w_\alpha(v) \leq 1/4\} \subseteq V$. Then we have $\Pr_{v \sim V}[v \in G_V] \geq 1 - \gamma/2$.

In addition,

$$\mathbb{E}_{e \sim E}\left[\Pr_{\varphi \sim \mathcal{D}}\left[\varphi(\alpha) = 1 \mid e\right]\right] = \Pr_{\varphi \sim \mathcal{D}}\left[\varphi(\alpha) = 1\right] \geq 3\gamma,$$

where we used the notation "$|e$" to represent the condition that the edge e is selected in the execution of $\mathcal{D}$. Thus, we observe that

$$\Pr_{e \sim E}\left[\Pr_{\varphi \sim \mathcal{D}}\left[\varphi(\alpha) = 1 \mid e\right] \geq \gamma\right] \geq 2\gamma;$$

otherwise, the expectation must be less than $1 \cdot 2\gamma + \gamma \cdot 1 = 3\gamma$.

Let $G_E := \{e \in E : \Pr_{\varphi \sim \mathcal{D}}\left[\varphi(\alpha) = 1 \mid e\right] \geq \gamma\} \subseteq E$. Then, we have $\Pr_{e \sim E}[e \in G_E] \geq 2\gamma$.

Since the constraint graph is biregular, a vertex $v' \in V$ selected as $(v,w) \sim E$ and $v' \sim \{v,w\}$ is distributed uniformly over V. Thus, we have

$$\Pr_{(v,w) \sim E}\left[v \notin G_V \vee w \notin G_V\right] \leq 2 \cdot \Pr_{\substack{(v,w) \sim E \\ v' \sim \{v,w\}}}\left[x \notin G_V\right] = 2 \cdot \Pr_{v' \sim V}\left[v' \notin G_V\right] \leq 2 \cdot (\gamma/2) = \gamma.$$

From above, we obtain that

$$\Pr_{e=(v,w) \sim E}\left[e \in G_E \wedge v \in G_V \wedge w \in G_V\right] \geq \gamma.$$

For each $v \in V_L$ and $w \in V_R$, we define functions $\alpha_v \colon \{0,1\}^{|V_L|} \to \{0,1\}$ and $\alpha_w \colon \{0,1\}^{|V_R|} \to \{0,1\}$ as

$$\alpha_v(S) = \alpha(x_{v,S}) \text{ and } \alpha_v(T) = \alpha(y_{w,T}).$$

Below, we fix $e = (v,w) \in E$ satisfying $e \in G_E \wedge v \in G_V \wedge w \in G_V$ arbitrarily and let $\pi := \pi_e$. By the construction of $\mathcal{D}$ and the union bound, we have

$$\begin{aligned}\Pr_{\varphi\sim\mathcal{D}}[\varphi(\alpha) = 1 \mid e] &\le M \cdot \mathbb{E}_{S,T_1,\dots,T_{d-1}}\left[\mathbb{E}_{\bar{S},\bar{T}_1,\dots,\bar{T}_{d-1}}\left[\alpha_v(\bar{S}) \cdot \prod_{i\in[d-1]} \alpha_w(\bar{T}_i)\right]\right] \\ &= M \cdot \mathbb{E}_{S,T_1,\dots,T_{d-1}}\left[\mathrm{T}^{\mu_L}_{1-\tau}\alpha_v(S) \cdot \prod_{i\in[d-1]} \vee\text{-}\mathrm{N}^{\pi^{-1},\mu_R}_{1-\tau}\alpha_w \circ (\vee\cdot\pi)(T_i)\right].\end{aligned}$$

Recall that $S \sim \mathrm{Ber}(\mu_L)^{\otimes|\Sigma_L|}$ and $(T_1,\dots,T_{d-1}) \sim \tilde{\mathrm{N}}^{\otimes(d-1)}_{\rho}(S)$.

Now, we define two functions $f, g \colon \{0,1\}^{|\Sigma_L|} \to [0,1]$ as

$$f(S) := \mathrm{T}^{\mu_L}_{1-\tau}\alpha_v(S) \text{ and } g(T) := \vee\text{-}\mathrm{N}^{\pi^{-1},\mu_R}_{1-\tau}\left(\alpha_w \circ (\vee\cdot\pi)\right)(T).$$

Then, $\Pr_{\varphi\sim\mathcal{D}}[\varphi(\alpha) = 1 \mid e] \le M \cdot \mathbb{E}[f(S) \cdot \prod_{i\in[d-1]} g(T_i)]$.

We evaluate the mean of f as follows:

$$\mathbb{E}_S[f(S)] = \mathbb{E}_S\mathbb{E}_{\bar{S}\sim\mathrm{N}^{\mu}_{1-\tau}(S)}[f(S)] = \mathbb{E}_S[f(S)] = \sum_{S:\alpha_{v,S}=1} \mu_L^{|S|}(1-\mu_L)^{|\Sigma_L|-|S|} = w_\alpha(v) \le 1/4,$$

where the inequality holds since $v \in G_V$. In the same way, we evaluate the mean of g as

$$\begin{aligned}\mathbb{E}_T[g(T)] &= \mathbb{E}_T\left[\vee\text{-}\mathrm{N}^{\pi^{-1},\mu_R}_{1-\tau}\alpha_w \circ (\vee\cdot\pi)(T)\right] \\ &= \mathbb{E}_T\left[\mathrm{N}^{\mu_R}_{1-\tau}\alpha_w(\vee\cdot\pi(T))\right] \\ &= \mathbb{E}_{\bar{T}\sim\mathrm{Ber}(\mu_R)^{\otimes|\Sigma_R|}}\left[\mathrm{N}^{\mu_R}_{1-\tau}\alpha_w(\bar{T})\right] \\ &= \mathbb{E}_{\bar{T}\sim\mathrm{Ber}(\mu_R)^{\otimes|\Sigma_R|}}\left[\alpha_w(\bar{T})\right] \\ &= w_\alpha(w) \le 1/4.\end{aligned}$$

where the second equality follows from Proposition 5.6, and the inequality holds because $w \in G_V$.

Therefore, we can apply Lemma 5.10 for f and g. Suppose that

$$\{i \in \Sigma_L : \mathrm{Inf}^{\mu_L}_i(f) > t\} \cap \{i \in \Sigma_L : \mathrm{Inf}^{\mu_L}_i(g) > t\} = \emptyset.$$

Then,

$$\mathbb{E}_{S,T_1,\dots,T_{d-1}}\left[f(S) \cdot \prod_{i\in[d-1]} g(T_i)\right] \le 2^{-d} = 2^{-\log^2 k(n)}.$$

Since $e \in G_E$, we have

$$\frac{1}{\log k(n)} = \gamma \le \Pr_{\varphi\sim\mathcal{D}}[\varphi(\alpha) = 1 \mid e] \le M \cdot \mathbb{E}[f(S) \cdot \prod_{i\in[d-1]} g(T_i)] \le \frac{k(n) \cdot \mathsf{polylog} k(n)}{2^{\log^2 k(n)}},$$

which is contradiction for any large enough n since $k(n) = \omega(1)$. Thus, there exists $i \in \Sigma_L$ such that $\mathrm{Inf}^{\mu_L}_i(f) > t$ and $\mathrm{Inf}^{\mu_L}_i(g) > t$. Notice that any coordinate $i \notin \Sigma_{L|e}$ is irrelevant for g by the definition of $\vee\cdot\pi$ (i.e., $\mathrm{Inf}^{\mu_L}_i(g) = 0$). Thus, we can assume that $i \in \Sigma_{L|e}$.

Recall that

$$t < \mathrm{Inf}_i^{\mu_L}(f) = \mathrm{Inf}_i^{\mu_L}\left(\mathrm{T}_{1-\tau}^{\mu_L}\alpha_v\right)$$

In addition, by Lemma 5.7, we have

$$t < \mathrm{Inf}_i^{\mu_L}(g) = \mathrm{Inf}_i^{\mu_L}\left(\vee\text{-}\mathrm{N}_{1-\tau}^{\pi^{-1},\mu_R}\left(\alpha_w \circ (\vee \cdot \pi)\right)\right) \le \mathrm{Inf}_{\pi(i)}^{\mu_R}\left(\mathrm{T}_{1-\tau}^{\mu_R}\alpha_w\right).$$

Now, we construct a random assignment for the original LABEL-COVER instance. For each $v \in V_L$ and $w \in V_R$, we define subsets $A_v \subseteq \Sigma_L$ and $A_w \subseteq \Sigma_R$ of alphabets as

$$A_v := \left\{i \in \Sigma_L : \mathrm{Inf}_i^{\mu_L}\left(\mathrm{T}_{1-\tau}^{\mu_L}\alpha_v\right) > t\right\}$$
$$A_w := \left\{j \in \Sigma_R : \mathrm{Inf}_j^{\mu_L}\left(\mathrm{T}_{1-\tau}^{\mu_R}\alpha_w\right) > t\right\}.$$

By Lemma 3.3, we have $|A_v| < t^{-1}\tau^{-1}$ and $|A_w| < t^{-1}\tau^{-1}$ for any v and w.

We consider the random assignment where we assign each $v \in V_L$ (resp. $w \in V_R$) an alphabet selected uniformly at random from A_v (resp. A_w). We have shown that for every $e = (v, w) \in E$ satisfying $e \in G_E$ and $v, w \in G_V$, there exists $i \in \Sigma_{L|e}$ such that $i \in A_v$ and $\pi_e(i) \in A_w$. Recall that such a *good* edge $e = (v, w)$ is selected with probability at least γ over $e \sim E$, and the random assignment to (v, w) coincides with $(i, \pi_e(i))$ with probability at least $1/(|A_v| \cdot |A_w|) \ge t^2\tau^2$.

Therefore, the expected fraction of constraints of the LABEL-COVER instance satisfied by the random assignment assignment is at least

$$\gamma \cdot t^2 \cdot \tau^2 > \varepsilon(n).$$

Thus, there exists an assignment to $V_L \cup V_R$ that satisfies at least $\varepsilon(n)$ fraction of constraints. □

5.4 Proof of Lemma 5.10

Lemma 5.10 is an immediate corollary of the theorems presented by [KS15], where they utilized the invariance principle developed by Mossel [Mos10]. The proof of Lemma 5.10 is almost same as the proof presented in [GL22, Appendix C]. For completeness, we derive Lemma 5.10 from the previous work [KS15] in this section.

First, we review key notions and results presented in [KS15].

Gaussian Stability

Definition 5.11 (Gaussian Stability)**.** *Let $\Phi\colon \mathbb{R} \to [0,1]$ be the cumulative distribution function of the standard Gaussian. For a parameter $\rho \in [0,1]$, we define*

$$\Gamma_\rho(\mu, \nu) := \Pr\left[X \le \Phi^{-1}(\mu) \wedge Y \le \Phi^{-1}(\nu)\right],$$

where (X, Y) are two standard Gaussian random variables with covariance matrix $\left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$. For $d \ge 3$, $(\rho_1, \ldots, \rho_{d-1}) \in [0,1]^{d-1}$, and $(\mu_1, \ldots, \mu_d) \in [0,1]^d$, we extend the definition above inductively as

$$\Gamma_{\rho_1,\ldots,\rho_{d-1}}(\mu_1, \ldots, \mu_d) := \Gamma_{\rho_1}(\mu_1, \Gamma_{\rho_2,\ldots,\rho_{d-1}}(\mu_2, \ldots, \mu_d)).$$

Lemma 5.12 ([KS15, Lemma 2.4])**.** *For any $d \ge 2$ and $T \ge 2$ such that $1 \ge \mu_i \ge 1/T$ for each $i \in [d]$, there exists a universal constant $C > 0$ such that for any $\varepsilon \in (0, 1/2]$, if*

$$\rho \le \frac{\varepsilon}{C(d-1)(\log T/\varepsilon)},$$

then

$$\Gamma_{\bar{\rho}_{d-1}}(\mu_1,\ldots,\mu_d) \le (1+\varepsilon)^{d-1}\prod_{i=1}^{d}\mu_i,$$

where $\bar{\rho}_{d-1}$ *is a* $(d-1)$*-tuple with each entry* ρ*.*

Correlation and Invariance Principle

In this work, we only consider finite probability spaces.

Definition 5.13 (Correlation of Probability Spaces)**.** *Let* $(\Omega^1\times\Omega^2,\mu)$ *is a finite correlated probability space with the marginal probability spaces* (Ω^1,μ^1) *and* (Ω^2,μ^2)*. The correlation between these spaces is defined as*

$$\rho\left(\Omega^1,\Omega^2;\mu\right) := \sup\left\{|\mathbb{E}_\mu[fg]| : f\in L^2(\Omega^1,\mu^1), g\in L^2(\Omega^2,\mu^2)\, s.t.\, \mathbb{E}[f]=\mathbb{E}[g]=0, \mathbb{E}[f^2]=\mathbb{E}[g^2]\le 1\right\}.$$

For $d\ge 3$*, the correlation of* d *correlated spaces* $(\Omega^1\times\cdots\times\Omega^d,\mu)$ *is defined as*

$$\rho\left(\Omega^1,\ldots,\Omega^d;\mu\right) := \max_{i\in[d]}\rho\left(\prod_{j\neq i}\Omega^j,\Omega^i;\mu\right)$$

Based on the invariance principle developed in [Mos10], Khot and Saket [KS15] presented the following useful multi-linear Gaussian stability bound.

Theorem 5.14 ([KS15, Theorem 2.10])**.** *Let* $\{(\prod_{j=1}^{d}\Omega_i^j,\mu_i)\}_{i\in[m]}$ *be a sequence of correlated spaces such that for each* $i\in[m]$*, the probability of any atom in* $(\prod_{j=1}^{d}\Omega_i^j,\mu_i)$ *is at least* $\alpha\le 1/2$ *and* $\rho(\Omega^1,\ldots,\Omega_i^d)\le\rho$*. Then there exists a universal constant* $C>0$ *such that, for every* $\nu>0$*, if we take*

$$\tau = \frac{1}{d^2}\cdot\left(\frac{\nu}{d}\right)^{C\frac{d\log(1/\alpha)\log(d/\nu)}{\nu(1-\rho)}},$$

and d *functions* $\{f_j:\prod_{i=1}^{m}\Omega_i^j\to[0,1]\}_{j\in[d]}$ *satisfy that for all* $j,j'\in[d]$ *with* $j\neq j'$

$$\{i\in[m]:\mathrm{Inf}_i(f_j)>\tau\}\cap\{i\in[m]:\mathrm{Inf}_i(f_{j'})>\tau\}=\emptyset,$$

then

$$\mathbb{E}\left[\prod_{j\in[d]}f_j\right]\le\Gamma_{\bar{\rho}_{d-1}}\left(\mathbb{E}[f_1],\ldots,\mathbb{E}[f_d]\right)+\nu,$$

where $\bar{\rho}_{d-1}$ *is a* $(d-1)$*-tuple with each entry* ρ*.*

Proof of the Key Lemma

Now, we derive Lemma 5.10 from Lemma 5.12 and Theorem 5.14.

Proof of Lemma 5.10. Let c,c' be the universal constants of Lemma 5.12 and Theorem 5.14, respectively. Recall that for $\mu\in(0,1/2)$ and $d\in\mathbb{N}$ with $d\ge 2$, the value $\rho\in[0,1]$ satisfies $\rho\le 1/(cd^2\log(4d))$ and

$$t := t(\mu,d,\rho) = d^{-2}\cdot\left(\frac{1}{d\cdot 4^d}\right)^{c'd4^d(1-\rho)^{-1}\log(\rho^{-1}\mu^{-d})\log(d4^d)}.$$

For each $i \in [m]$, it is easy to observe that $(x_i^{(0)}, \dots, x_i^{(d-1)})$, where $x_i^{(0)} \sim \mathrm{Ber}_\mu$ and $(x_i^{(1)}, \dots, x_i^{(d-1)}) \sim \tilde{\mathrm{N}}_\rho^{\otimes(d-1)}(x_i^{(0)})$, is random variables defined on the d correlated probability space with correlation at most ρ and the probability of any atom is at least $\rho\mu^d$ ($< 1/2$).

Let $f\colon \{0,1\}^m \to [0,1]$ and $g\colon \{0,1\}^m \to [0,1]$ be functions satisfying that

$$\max\{\mathbb{E}_x[f(x)], \mathbb{E}_x[g(x)]\} \le 1/4$$

and

$$\{i : \mathrm{Inf}_i^\mu(f) > t\} \cap \{i : \mathrm{Inf}_i^\mu(g) > t\} = \emptyset.$$

Define $f_1, f_2, \dots, f_d\colon \{0,1\}^m \to [0,1]$ as $f_1 \equiv f$ and $f_i \equiv g$ for all $i \neq 1$. Observe that τ defined in Theorem 5.14 with respect to $\alpha = \rho\mu^d$ and $\nu = 4^{-d}$ is equal to t. Thus, by applying Theorem 5.14 for $f_1, f_2, \dots, f_d$, we have

$$\mathbb{E}\left[f(x^{(0)}) \cdot \prod_{i\in[d-1]} g(x^{(i)})\right] \le \Gamma_{\bar{\rho}_{d-1}}\left(\mathbb{E}[f], \mathbb{E}[g], \dots, \mathbb{E}[g]\right) + 4^{-d} \le \Gamma_{\bar{\rho}_{d-1}}\left(1/4, \dots, 1/4\right) + 4^{-d}.$$

Since $\rho \le 1/(cd^2\log(4d))$, we apply Lemma 5.12 for $\varepsilon = 1/d$ and obtain that

$$\Gamma_{\bar{\rho}_{d-1}}\left(1/4, \dots, 1/4\right) \le \left(1 + \frac{1}{d}\right)^{d-1} \cdot 4^{-d} \le e \cdot 4^{-d}.$$

Thus, we conclude that

$$\mathbb{E}\left[f(x^{(0)}) \cdot \prod_{i\in[d-1]} g(x^{(i)})\right] \le (1+e)4^{-d} \le 2^{-d},$$

where we used $d \ge 2$. □

6 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 5.2.

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 6.1. *Let $\ell := \ell(m)$, $\gamma := \gamma(m)$, and $\varepsilon := \varepsilon(m)$. The problem $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma(\ell)}\mathfrak{S}_\ell$-DMMSA, where m is the instance size, is reducible to a learning problem as follows: There exists a polynomial-time algorithm R such that, for given $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma(\ell)}\mathfrak{S}_\ell$-DMMSA-instance $z = (\mathcal{D}, w, 1^n, s)$ and randomness $z' \sim \{0,1\}^\lambda$, where $\lambda := \lambda(|z|)$, R produces a distribution $\mathcal{E}_{z,z'}$ such that*

- *$\{\mathcal{E}_{z,z'}\}_{z,z'}$ is samplable with $\ell(|z|) + o(\ell(|z|))$ advice.*
- *(Completeness) If z is an yes instance, then for every $z' \in \{0,1\}^\lambda$, there exists a $\mathsf{poly}(n)$-time program h of size $(s+1)\cdot\lambda$ such that*

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

- (*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 $0.99\sigma(\ell(|z|)) \cdot s \cdot \lambda$ such that*

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

Proof. Let $z = (\mathcal{D}, w, 1^n, s)$ be an instance of $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma(\ell)}\mathfrak{S}_\ell$-DMMSA. Here, $n, s \in \mathbb{N}$, $\mathcal{D}$ is a circuit representing a sampler for a distribution on n-variate monotone formulas whose authorized sets admit secret sharing of total length $\ell := \ell(|z|)$, and $w\colon [n] \to [0,1]$ be 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 $i \in [n]$.

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 $n' := \ell + \rho + \lambda$.

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 $j_i \in [n]$ the index of the variable to which s_i is distributed. Let $z_j \sim \{0,1\}^{w(j)\lambda}$ for each $j \in [n]$, and $c_i := s_i \oplus \mathsf{DP}_1(f_{j_i}; z_{j_i})_{w(j_i)\lambda+1}$ 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 $r, b, \{z_i\}$, 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 at most $\ell := \ell(|z|)$ advice. Let S be a uniform algorithm that takes $(1^{\langle|z|,|z'|\rangle}, a, r')$, where $r' \in \{0,1\}^{\rho+\ell\lambda+\mathsf{poly}(|n'|)}$ and $a \in \{0,1\}^\ell$, and outputs $(r'_{[\rho+\lambda]} \circ a, r'_{\rho+\lambda+1})$. We consider the advice function $\alpha_{z,z'}$ that takes the same random seed r', has embedded nonuniform advice z and $z' = \{f_i\}_{i\in[n]}$, and outputs the description of $S(1^{\langle|z|,|z'|\rangle}, c_1 \circ \cdots \circ c_\ell, \text{-})$, where $c_1 \circ \cdots \circ c_\ell$ is an ℓ-bit string following the definition of $\mathcal{E}_{z,z'}$. Namely, $r := r'_{[\rho]}$, $\varphi := \mathcal{D}(r)$, $z_j := r'_{[\rho+(w(j)+\cdots+w(j-1))\lambda+1:\rho+(w(1)+\cdots+w(j))\lambda]}$ for each $j \in [n]$, $b := r'_{\rho+\lambda+1}$, $s := \mathsf{Share}(b;\varphi)$, where Share uses fresh randomness from the suffix of r', j_i is the same as in $\mathcal{E}$, $c_i = s_i \oplus \mathsf{DP}(f_{j_i}; z_{j_i})_{w(j_i)\lambda+1}$ for each $i \in [\ell]$. Then, for every z and $z' = \{f_i\}_i$, the distribution $\mathcal{E}_{z,z'}$ is statistically identical to that of $U(S(1^{\langle|z|,|z'|\rangle}, c_1 \circ \cdots \circ c_\ell, \text{-}), r') = U(\alpha_{z,z'}(r'), r')$ over r'. Thus, $\mathcal{R}$ produces a distribution (over samples) samplable with advice complexity at most $\ell + o(\ell)$.

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' is the polynomials specified below.

Claim 6.2 (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 polynomial-time program h of size at most $(s+1)\cdot\lambda$ $(\leq s')$ such that*

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

Claim 6.3 (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 $0.99\sigma(\ell(|z|)) \cdot s \cdot \lambda$ such that*

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

Proof of Claim 6.2. 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

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

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 $r \in \{0,1\}^n$, $z_i \in \{0,1\}^{w(i)\lambda}$ for each $i \in [n]$, 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 $\mathsf{DP}_1(f_{j_i}; z_{j_i})$ 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 $c = c_1 \cdots \circ c_\ell$. 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

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

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

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

The claim holds by selecting the polynomial p large enough so that every $\lambda \geq p(|z|)$ satisfies

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

$\diamond$

Proof of Claim 6.3. 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 $0.99\sigma(\ell(|z|)) \cdot s \cdot \lambda$ that maps n' bits to 1 bit. We will show that

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

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 $(Z_1Y_1, \ldots, Z_nY_n)$, where Z_i (resp. Y_i) is a random variable taking values in $\{0,1\}^\lambda$ (resp. $\{0,1\}$), computes (i) $\varphi = \mathcal{D}(r)$,

(ii) $(s_1, \ldots, s_\ell) = \mathsf{Share}(b; \varphi)$ with randomness r', (iii) $c_i = Y_{j_i} \oplus s_i$ for each $i \in [\ell]$ (recall that j_i is the index of the i-th share), and then outputs 1 if and only if

$$h(r \circ (Z_1)_{[w(1)\lambda]} \circ \cdots \circ (Z_n)_{[w(n)\lambda]} \circ c_1 \circ \cdots \circ c_\ell) = b.$$

It is easy to verify that if $Z_i Y_i = \mathsf{DP}_1(\tilde{f}_i; Z_i)$ 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 3.12), there exists a subset $B \subseteq [n]$ such that

$$\mathrm{K}(\tilde{f}_B | D) \leq |B| \cdot (n + a + O(\log n\lambda a \gamma(|z|)^{-1})) \leq \mathsf{poly}(|z|) \cdot \log \lambda, \tag{2}$$

and

$$\left| \Pr_{\alpha, \{Z_i Y_i\}} [D(\alpha, Z_1 Y_1, \ldots, Z_n Y_n) = 1] - \Pr_{\alpha, \{Z_i Y_i'\}} \left[D(\alpha, Z_1 Y_1', \ldots, Z_n Y_n') = 1\right] \right| \leq \gamma(|z|) \tag{3}$$

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

We have that

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

where the second inequality holds because D is constructed from h. Notice that f_B is a random string of length $w(B)\lambda$, where $w(B) = \sum_{i \in B} w(i)$. Since $w(B) \geq \min_i w(i) \geq 1/\mathsf{poly}(|z|)$ (where we assume $B \neq \emptyset$, otherwise it contradicts Eq. (2)), there exists a polynomial p_1' such that for every $\lambda \geq p_1'(|z|)$, it holds that $w(B)\lambda \geq 3|z|$ (note that p_1' is independent of B since it holds as long as $B \neq \emptyset$). By the standard counting argument, in this case, $\mathrm{K}(f_B) \geq w(B)\lambda - |z|$ with probability $1 - \mathsf{negl}(|z|)$ over the choices of $\{f_i\}$. Under this event, along with Eq. (2), we have

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

for some universal polynomial p_0.

Since $w(B) \geq \min_i w(i) \geq 1/\mathsf{poly}(|z|)$ (as long as $B \neq \emptyset$) there exists a universal polynomial p_2' such that for every $\lambda \geq p_2'(|z|)$,

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

Thus, by selecting p' as $p'(|z|) = \max\{p_1'(|z|), p_2'(|z|)\}$, for every $\lambda \geq p'(|z|)$, the following holds with probability at least $1 - \mathsf{negl}(|z|)$:

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

Below, we consider the case where the event above occurs.

Since $|h| \leq 0.99\sigma(\ell(|z|)) \cdot s \cdot \lambda$, we have

$$w(B) \leq \frac{0.99}{0.995} \sigma(\ell(|z|))s < \sigma(\ell(|z|))s.$$

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

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

Now, consider the execution of $D(\alpha, Z_1Y'_1, \ldots, Z_nY'_n)$. If the randomness r (which is a part of α) yields such φ, the input to h does not contain any information on the authorized subset of shares because $c_i = s_i \oplus Y'_{j_i}$ and $Y'_{j_i} \sim \{0,1\}$ when $j_i \notin B$. In this case, by the security of the secret sharing scheme, the probability that D outputs 1 (i.e., h correctly outputs b) is $1/2$. Thus,

$$\Pr_{\alpha,\{Z_iY'_i\}}\left[D(\alpha, Z_1Y'_1, \ldots, Z_nY'_n) = 1\right] \leq \frac{1}{2} + \gamma(|z|).$$

Therefore, by Eq. (3)

$$\begin{aligned}\Pr_{\alpha,\{Z_iY_i\}}[D(\alpha, Z_1Y_1, \ldots, Z_nY_n) = 1] &\leq \Pr_{\alpha,\{Z_iY'_i\}}\left[D(\alpha, Z_1Y'_1, \ldots, Z_nY'_n) = 1\right] + \gamma(|z|) \\ &\leq \frac{1}{2} + 2\gamma(|z|).\end{aligned}$$

By contrast,

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

From the two inequalities above, we conclude that

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

◇

□

The lemma above implies Theorem 1.3 along with Theorem 5.2.

Corollary 6.4 (Theorem 1.3)**.** *There exist functions $\ell(n) = \omega(1)$, $\varepsilon(n) = o(1)$, and $\gamma(n) = o(1)$ such that, the following problem is* NP*-hard under a* BPP *reduction: For given pair of a distribution $\mathcal{D}$ over $\{0,1\}^n \times \{0,1\}$ (described as a circuit) samplable with advice complexity $\ell(n)$ and 1^s, where $s \in \mathbb{N}$ with $s \leq \mathsf{poly}(n)$, distinguish the following two cases:*

Yes cases: There exists a $\mathsf{poly}(n)$-time program h of size s such that $\Pr_{(x,b)\sim\mathcal{D}}[h(x) = b] \geq 1 - \varepsilon(n)$.

No cases: For all programs h of size at most $s \cdot \frac{\ell(n)}{\mathsf{polylog}l(n)}$, $\Pr_{(x,b)\sim\mathcal{D}}[h(x) = b] \leq \frac{1}{2} + \frac{\gamma(n)}{2}$.

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). Corollary 6.4 demonstrates that our result holds in a stronger setting, where the learner also has access to the description of the underlying distribution for learning.

7 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 7.1. *For every $\ell(n) = \omega(1)$, every $\varepsilon(n) = o(1)$, and every constant $\gamma \in [0, 1/4)$, there exists $C > 0$ such that $\mathrm{Gap}^{\varepsilon(n),\gamma}_{C\cdot\ell(n)}\mathfrak{S}_{\ell(n)}$-DMMSA is in* coAM*, where n is the instance size.*

Note that Theorem 7.1 implies Theorem 1.9 as special cases since (i) $\mathsf{F}[\ell] \subseteq \mathfrak{S}_\ell$ (Theorem 3.10) and (ii) $\mathsf{Junta}[\ell] \subseteq \mathfrak{S}_{1.5^{\ell+o(\ell)}}$ [AN21].

Theorem 7.1 is proved across Sections 7.1 to 7.3, outlined as follows: In Section 7.1, 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 7.3, 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 7.1. In Section 7.5, 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 7.2 (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 all $\rho \in \{0,1\}^*$, i.e.,*

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

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 7.1 and 7.2.

Lemma 7.3. *Let $\ell(n) = \omega(1)$, $\varepsilon(n) = o(1)$, and let $\gamma \in [0, 1/4)$ be a constant. Then, there exists $C \geq 1$ such that $\mathrm{Gap}^{\varepsilon(n),\gamma}_{C \cdot \ell(n)}\mathfrak{S}_{\ell(n)}$-DMMSA, where n is the instance size, 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 7.1.

Theorem 7.4. *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 7.4 in Section 7.3.

7.1 Inductive Inference: From Learning to Universal Extrapolation

Towards proving Lemma 7.3, we first reduce learning to extrapolation under the time-bounded universal distribution.

Lemma 7.5. *Let $\mathcal{D} = \{\mathcal{D}_{z,z'}\}_{z\in Z, z'\in\{0,1\}^{r(z)}}$, where $Z \subseteq \{0,1\}^*$ and $r(z) \le \mathsf{poly}(|z|)$, be a samplable distribution satisfying that there exist $n, s\colon Z \to \mathbb{N}$, $\gamma(z) = o(1)$, 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 $n(z) \le \mathsf{poly}(|z|)$;*
- *there exists a $p(\cdot)$-time program h of description size $s(z,z')$ such that*

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

Then, for every $\varepsilon \in (0, 1/2)$, there exist a constant $c := c_{\varepsilon,\delta} > 0$ and a polynomial $t(|z|)$ such that for every long enough $z \in Z$ and for $m \ge m(z,z') = c\cdot(s(z,z') + \log|z|)$,

$$\Pr_{z'}\left[\Pr_{(x^1,b^1),\dots,(x^m,b^m),i,\mathsf{Next}_1}\left[\mathsf{Next}_1\left(xb_{<i}; \mathrm{Q}_{|z}^{t(|z|)}\right) = b_i\right] \ge 1-\varepsilon\right] \ge 1 - o_{|z|}(1),$$

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

Proof. Let $\varepsilon > 0$ and $c := 6c_0^2\varepsilon^{-2}$, 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. Let $t := t(|z|) = m(|z|)\cdot\tau(|z|)$, where τ is a large enough polynomial (particularly for p and the time-complexity of sampling according to $\mathcal{D}$). Fix a long enough $z \in Z$ arbitrarily. Let $n := n(z) \le \mathsf{poly}(|z|)$,and $\gamma := \gamma(z)$.

Fix $z' \in \{0,1\}^{r(z)}$. Let $s := s(z,z')$. Let $X_1,\dots,X_m,B_1,\dots,B_m$ be a random variable representing the values of $x_1,\dots,x_m,b_1,\dots,b_m$ respectively. 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 7.6. *For each z' and each $(x,b)\sim(X,B)$,*

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

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, for large enough polynomial τ and $t = m(|z|) \cdot \tau(|z|) \geq \tau(|z|)$,

$$\begin{aligned}
\mathrm{KL}\Big(B \,\Big|\, X \,\Big\|\, (\mathrm{Q}^t_{|z})_{[mn+1:mn+m]} \,\Big|\, (\mathrm{Q}^t_{|z})_{[mn]}\Big) &\leq \mathbb{E}_{(x,b)\sim(X,B)}\left[\log \frac{1}{\Pr[\mathrm{Q}^t_{[mn+m]} = xb | Q^t_{[mn]} = x]}\right] \\
&\leq \mathbb{E}_{(x,b)}\left[c_0(s + \mathrm{cd}^t(x,b|z) + \gamma^{0.75}_{x,b} m + \log|z|)\right] \\
&= c_0 s + c_0 \mathbb{E}_{(x,b)}[\mathrm{cd}^t(x,b|z)] + c_0 \log|z| + c_0 \mathbb{E}_{x,b}\left[\gamma^{0.75}_{x,b}\right] \cdot m \\
&\leq c_0 s + 6c_0^2 \log|z| + c_0 \mathbb{E}_{x,b}\left[\gamma^{0.75}_{x,b}\right] \cdot m \\
&\leq c_0 s + 6c_0^2 \log|z| + c_0 \mathbb{E}_{x,b}\left[\gamma_{x,b}\right]^{0.75} \cdot m \\
&= c_0 s + 6c_0^2 \log|z| + c_0 \gamma^{0.75} \cdot m,
\end{aligned}$$

where the third inequality follows from Lemma 3.15, and the last inequality follows from Jensen's inequality.

By the chain rule for the KL divergence,

$$\begin{aligned}
\frac{1}{m}\sum_{i\in[m]} \mathrm{KL}\Big(B_i \,\Big|\, XB_{<i} \,\Big\|\, (\mathrm{Q}^t_{|z})_{[mn+i]} \,\Big|\, (\mathrm{Q}^t_{|z})_{[mn+i-1]}\Big) &\leq \frac{c_0 s}{m} + \frac{6c_0^2 \log|z|}{m} + c_0\gamma^{0.75} \\
&\leq \frac{\varepsilon^2}{6} + \varepsilon^2 + \gamma^{0.75}.
\end{aligned}$$

Since $\gamma := \gamma(|z|) = o(1)$, 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,

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

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,

$$\begin{aligned}
&\Pr_{\mathsf{Next}_1}\left[\mathsf{Next}_1(xb_{<i}; \mathrm{Q}^t_{|z}) \neq h(x_i)\right] \\
&\quad\leq \Pr_{B_i}\left[B_i \neq h(x_i) | XB_{<i} = xb_{<i}\right] + \Delta_{\mathrm{tv}}\left((B_i|XB_{<i} = xb_{<i}), ((\mathrm{Q}^t_{|z})_{[mn+i]} | (\mathrm{Q}^t_{|z})_{[mn+i-1]} = xb_{<i})\right).
\end{aligned}$$

By taking the expectation over i and $XB_{<i}$, we have

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

By the union bound,

$$\begin{aligned}\Pr_{i,XB_{<i},\mathsf{Next}_1}\left[\mathsf{Next}_1(xb_{<i};\mathrm{Q}^t_{|z})\neq B_i\right] &\le \Pr_{i,XB_{<i},\mathsf{Next}_1}\left[\mathsf{Next}_1(xb_{<i};\mathrm{Q}^t_{|z})\neq h(x_i)\right] + \Pr_{i,XB_{<i},B_i}\left[B_i\neq h(x_i)\right]\\ &\le 2\gamma+\varepsilon,\end{aligned}$$

which is less than 2ε for long enough z since $\gamma = o(1)$. 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 7.6. Remember that $t = \tau(|z|)\times m(|z|)$. First, we show that

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

when τ is large enough. 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 truncated 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 take 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 $\mathsf{wt}(x^i) = \gamma_{x,b}m$ and by Fact 3.13, it is reconstructed from $O(\gamma_{x,b}^{0.75}m+\log|z|)$-size program when τ is large enough. Thus, the description size of Π' is at most

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

Thus, we have

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

where the summation is taken over t-time programs Π that are given z and produce x as a prefix.

By contrast, we have

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

Thus, we conclude

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

by selecting large enough $c_0 > 0$. ◇

□

7.2 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 7.3.

Lemma 7.7. *Let $\mathcal{D} = \{\mathcal{D}_{z,z'}\}_{z\in Z, z'\in\{0,1\}^{r(z)}}$, where $Z \subseteq \{0,1\}^*$ and $r(z) \le \mathsf{poly}(|z|)$, be a samplable distribution satisfying the conditions in Lemma 7.5.*

For every $\varepsilon, \delta \in (0, 1/2)$, there exist a polynomial-time nonadaptive randomized oracle machine h, an auxiliary-input function $\{f_z\}_{z\in Z}$, a constant $c := c_{\varepsilon,\delta} > 0$ such that for every long enough $z \in Z$, for every oracle $\mathcal{I}$ that inverts f_z (i), and for $m \ge m(z,z') = c\cdot(s(z,z') + \log|z|)$,

$$\Pr_{z'}\left[\Pr_{S=\{(x^1,b^1),\ldots,(x^m,b^m)\}}\left[\Pr_{h,(x,b)}\left[h^{\mathcal{I}}(S,x,z) = b\right] \ge 1-\varepsilon\right] \ge 1-\delta\right] \ge 1 - o_{|z|}(1),$$

where $(x^1,b^1),\ldots,(x^m,b^m),(x,b) \sim \mathcal{D}_{z,z'}$.

To show the lemma, we use the following result.

Theorem 7.8 ([IL90; HN23])**.** *For every samplable distribution $\mathcal{D} = \{\mathcal{D}_z\}_{z\in Z}$, where $Z \subseteq \{0,1\}^*$, every constant c, and every large enough polynomial τ, 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,*

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

Proof of Lemma 7.7. We define a samplable distribution $\mathcal{D}' = \{\mathcal{D}'_z\}_{z\in Z}$ as a distribution of $x^1\circ\cdots\circ x^m\circ b^1\circ\cdots\circ b^m$, where $z' \sim \{0,1\}^{r(z)}$ and $(x^1,b^1),\ldots,(x^m,b^m) \sim \mathcal{D}_{z,z'}$ (recall that $m = c(s(z,z') + \log|z|)$) for each $z \in Z$. We apply Theorem 7.8 for $\mathcal{D}'$ 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,

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

where τ is a large enough polynomial, and y represents $x^1\circ\cdots\circ x^m\circ b^1\circ\cdots\circ b^m$.

By Markov's inequality,

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

where $xb_{<i} = x^1\circ\cdots\circ x^m\circ b^1\circ\cdots\circ b^{i-1}$.

By Lemma 7.5 and the union bound, it holds that with probability $1 - o_{|z|}(1)$ over the choice of z',

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

and

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

Below we only consider such z'.

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

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

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

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

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

where $i \sim [m]$, and

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

By Markov's inequality,

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

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

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

By Markov's inequality,

$$\Pr_{S}\left[\Pr_{x^*}\left[\textit{Equation (4)} \text{ holds}\right] \ge \frac{\varepsilon}{2}\right] \ge 1 - \delta.$$

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

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

Therefore, we obtain

$$\Pr_{z'}\left[\Pr_{S}\left[\Pr_{h,(x^*,b^*)}\left[h^{\mathcal{I}}(S, x^*, z) = b^*\right] \ge 1 - \varepsilon\right] \ge 1 - \delta\right] \ge 1 - o_{|z|}(1).$$

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

Now, we prove Lemma 7.3.

Proof of Lemma 7.3. For a large enough constant C, we will show that $\mathrm{Gap}^{\varepsilon(\cdot),\gamma}_{C\cdot\ell(\cdot)}\mathfrak{S}_{\ell(\cdot)}$-DMMSA is reducible to inverting an auxiliary-input function $f = \{f_z\}$ via a description-restricted context-sensitive FAIN reduction (recall that $\ell(\cdot) = \omega(1)$, $\varepsilon(\cdot) = o(1)$, and $\gamma \in [0, 1/4)$).

Let $z = (\mathcal{D}, w, 1^n, s)$ be an instance of $\mathrm{Gap}^{\varepsilon(\cdot),\gamma}_{C\cdot\ell(\cdot)}\mathfrak{S}_{\ell(\cdot)}$-DMMSA. Let R be the reduction in Lemma 6.1, 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. Recall that $\{\mathcal{E}_{z,z'}\}$ is samplable with ℓ advice. Thus, for each (z,z'), there exists a function $\alpha_{z,z'}$ such that (i) $\alpha_{z,z'}(r)$ outputs a ℓ-bit string, and (ii) the distribution of $U(\alpha_{z,z'}(r),r)$ is statistically identical to $\mathcal{E}_{z,z'}$ for $r \sim \{0,1\}^{\mathsf{poly}(|z|)}$ within negligible error. Below, we omit the argument on this negligible error. Recall that the advice $\alpha_{z,z'}(r)$ in the reduction R of Lemma 6.1 consists of ℓ bits produced by the secret sharing scheme. Consequently, $\alpha_{z,z'}$ is polynomial-time computable given (z,z'), and $U(\alpha_{z,z'}(\text{-}),\text{-})$ also halts in polynomial time and never outputs $\perp$.

We use Lemma 7.7 for the description size $s' = (\lambda+1)s$, $\varepsilon = 1/4-\gamma$, and $\delta = 1/32$. Let $h, m, \{f_z\}$, and $\mathcal{I}$ be as in Lemma 7.7. We also define a randomized oracle machine H given access to $\mathcal{I}$ as follows:

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

Here $\bar{r}$ is composed of m random seeds $r^1,\dots,r^m$ for $U(\alpha_{z,z'}(\text{-}),\text{-})$, $S = \{(x^1,b^1),\dots,(x^m,b^m)\}$, where $(x^i,b^i) = U(\alpha_{z,z'}(r^i),r^i)$ for each $i \in [m]$, and m is chosen as in Lemma 7.7 for s', i.e.,

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

For each $z' \in \{0,1\}^\lambda$, $\bar{r} \in \{0,1\}^{m\cdot\mathsf{poly}(|z|)}$, and $\rho \in \{0,1\}^{l(|z|)}$ (where l 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
>
> $$\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}[H^{\mathcal{I}(\text{-},z;\rho)}(x;\bar{r},z) = b]$$
>
> with additive accuracy error $\pm(1/16-\gamma/4)$ 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 $\tilde{p} \geq 5/8 + 3\gamma/2$.

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 $\mathrm{Gap}^{\varepsilon(\cdot),\gamma}_{C\cdot\ell(\cdot)}\mathfrak{S}_{\ell(\cdot)}$-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 7.9. *For every oracle $\mathcal{O}$, computable function l, and for every long enough instance z, if z is an yes instance, and $\mathcal{O}(\text{-},z;\rho)$ inverts f_z for all $\rho \in \{0,1\}^{l(|z|)}$, then*

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

Claim 7.10. *For each Turing machine I, computable function l, 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 $\rho \in \{0,1\}^{l(|z|)}$, then*

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

Proof of Claim 7.9. 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 6.1 shows that for every z', there exists a polynomial-time program h^* of size $s' = (s+1)\lambda$ such that

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

Thus, as long as z is an yes instance, $\mathcal{E}_{z,z'}$ satisfies the conditions of Lemma 7.5. For all $\rho \in \{0,1\}^{l(|z|)}$, Lemma 7.7 (recall that it was applied for $\varepsilon = 1/4 - \gamma$ and $\delta = 1/32$) shows that if $\mathcal{O}(\text{-}, z; \rho)$ inverts f_z, then

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

for long enough z. Namely, it holds that

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

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

$$\tilde{p} \geq \Pr[H^{\mathcal{O}(\text{-},z;\rho)}(x;\bar{r},z)] - (1/16 - \gamma/4) \geq 11/16 + 5\gamma/4 > 5/8 + 3\gamma/2,$$

where the last inequality follows from $\gamma < 1/4$.

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

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

We claim that if z is long enough, then

$$\Pr_{A,\rho}[A^{\mathcal{O}_I(\text{-},z;\rho)}(z) = 1] \geq 1/4 \implies \Pr_{\bar{r},\rho}[T_z(\bar{r},\rho) = 1] \geq 1/16. \tag{5}$$

First, we assume (5) 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 $\Pr_{A,\rho}[A^{\mathcal{O}_I(\text{-},z;\rho)}(z) = 1] \geq 1/4$. Thus, by (5), we have $\Pr_{\bar{r},\rho}[T_z(\bar{r},\rho) = 1] \geq 1/16$.

Let $G_z = \{(\bar{r},\rho) : T_z(\bar{r},\rho) = 1\}$. Since $\Pr_{\bar{r},\rho}[(\bar{r},\rho) \in G_z] = \Pr_{\bar{r},\rho}[T_z(\bar{r},\rho) = 1] \geq 1/16 > 0$, it holds that $G_z \neq \emptyset$. In addition, we observe that the set G_z is enumerable given z by checking whether $T_z(\bar{r},\rho)$ holds, as the number of possible z' and (x,b) is finite, and $\alpha_{z,z'}$ is computable. Thus, by letting $(\bar{r}^*,\rho^*)$ be the lexicographically first element in G_z, we have

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

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

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

Remember that

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

where $S = \{(x^1,b^1),\ldots,(x^m,b^m)\}$, and $(x^i,b^i) = U(\alpha_{z,z'}(r^i),r^i)$ for each $i \in [m]$. Furthermore, $\alpha_{z,z'}(\cdot)$ is just an ℓ-bit string for each i. Therefore, for each z', the description size of $H^{\mathcal{O}_I(\cdot,z;\rho^*)}(\cdot;\bar{r}^*,z)$ (as a time-unbounded program) is at most

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

When $\lambda = \mathsf{poly}(|z|)$ is large enough, we have

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

for a large enough constant $C > 0$.

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

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

By Lemma 6.1, z is not a no instance of $\mathrm{Gap}^{\varepsilon(\cdot),\gamma}_{C\cdot\ell(\cdot)}\mathfrak{S}_{\ell(\cdot)}$-DMMSA.

In the remainder, we observe the implication (5), 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

$$\Pr_{z',\bar{r}}\left[\Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}_I(\cdot,z;\rho)}(x;\bar{r},z)\right] \geq \frac{1}{2}+2\gamma\right] \geq \frac{3}{4}. \tag{6}$$

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 $E^\rho_{z,z',\bar{r}}$,

$$\tilde{p} \leq \Pr_{(x,b)\sim\mathcal{E}_{z,z'}}\left[H^{\mathcal{O}_I(\cdot,z;\rho)}(x;\bar{r},z)\right] + (1/16-\gamma/4) < (1/2)+2\gamma+1/16-\gamma/4 = 9/16+7\gamma/4 < 5/8+3\gamma/2,$$

where the last inequality follows from $\gamma < 1/4$. 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 (6) implies

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

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. ◇

□

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

In this section, we present the proof of Theorem 7.4. 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 7.11 ([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*

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

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

  where $\mathcal{O}_M(\text{-},x;\rho)|_{R(x)} = (\mathcal{O}_M(q^1,x;\rho),\ldots,\mathcal{O}_M(q^k,x;\rho))$ 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

$$(1 - 1/p(|x|))|f_{z_x}^{-1}(y)| \leq \tilde{s}_y \leq (1 + 1/p(|x|))|f_{z_x}^{-1}(y)|. \tag{7}$$

  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

$$\tilde{\ell}_y = \lfloor \log |f_{z_y}^{-1}(y)| \rfloor \tag{8}$$

  for all τ-light queries y satisfying Equation (7) (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 (7) and (8). 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 7.4.

Proof of Theorem 7.4. 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 7.11 for R, respectively.

We construct an 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 7.11). 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 1).

First, we show the completeness. Suppose that the given x is long enough and a no instance. Then, by Lemma 7.11,

$$\Pr[\langle P, V\rangle(x) = 1] \geq \Pr_{R,\rho}[R^{O_M(\text{-},x;\rho)}(x) = 0] - \Delta_{\mathrm{tv}}\left(\langle P_0, V_0\rangle(x), \mathcal{O}_M(\text{-}, x; \rho)|_{R(x)}\right) \geq \frac{3}{4} - \frac{1}{|x|}. \tag{9}$$

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 $\Pr[\langle \tilde{P}_0, V\rangle(x) = \bot] < 1/3$ since V outputs 0 in such cases. Thus, by Lemma 7.11,

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

which is a contradiction. Therefore, we have

$$\Pr[\langle \tilde{P}, V\rangle(x) = 1] \leq 2/3. \tag{10}$$

From Equations (9) and (10), we conclude that $\Pi \in$ coAM.

□

7.4 CoAM bound for Problems Reducible to GapLearn

In this section, we prove Theorem 1.12 using an approach similar to that of Theorem 7.1. 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 $x \in \Pi_{\mathrm{yes}}$ (resp. $x \in \Pi_{\mathrm{no}}$), then $f(x) \in \Pi_{\mathrm{yes}}$ (resp. $f(x) \in \Pi_{\mathrm{no}}$).

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 7.4, it suffices to show the following lemma for Theorem 1.12.

Lemma 7.12. *For every constant $\gamma \in (0,1)$, there exists a constant C such that for all functions l and ε with $\omega(1) \le \ell(n) \le n^{O(1)}$ and $\varepsilon(n) = o(1)$, the following holds: If a paddable promise problem Π is reducible to $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ 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 $s \ge \lambda(|z|)$, 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. Since this is an instance of $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$, we can assume that each $\mathcal{E}_{z,z'}$ is samplable with ℓ bits of advice. Thus, for each pair (z,z'), there exists a function $\alpha_{z,z'}$ such that (i) $\alpha_{z,z'}(r)$ outputs an ℓ-bit string, and (ii) the distribution of $U(\alpha_{z,z'}(r), r)$ is statistically identical to $\mathcal{E}_{z,z'}$ for $r \sim \{0,1\}^{\mathsf{poly}(|z|)}$, as long as it does not output $\perp$. In the following, we consider only the case where U does not output $\perp$. Let $q := q(|z|)$ be the query complexity of R, and let $p(|z|) := 4q(|z|)$

By Lemmas 7.5 and 7.7, there exist a randomized oracle machine h, a polynomial-time computable function $f = \{f_z\}$, and a constant $c := c_\gamma \ge 1$ such that for every long enough $z \in \{0,1\}^*$, for every oracle $\mathcal{I}$ that inverts f_z and every $m \ge c \cdot (s^*(z') + \log|z|)$,

$$\Pr_{z'}\left[\Pr_{S=\{(x^1,b^1),\ldots,(x^m,b^m)\},h}\left[\Pr_{(x,b)}\left[h^{\mathcal{I}}(S,x,z) = b\right] \ge \frac{3}{4} + \frac{\gamma}{4}\right] \ge \frac{3}{4}\right] \ge 1 - \frac{1}{4p(|z|)}, \tag{11}$$

where $(x^1,b^1),\ldots,(x^m,b^m),(x,b) \sim \mathcal{D}_{z,z'}$, and $s^*(z')$ is the minimum size polynomial-time program h such that

$$\Pr_{(x,b)\sim\mathcal{D}_{z,z'}}[h(x) = b] \ge 1 - \epsilon(n_{z'}),$$

where n is the parameter of GapLearn instance generated by using z'.

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 $\mathrm{Gap}^{\varepsilon,\gamma}_{\sigma}\mathrm{Learn}[\ell]$ correctly for all z' satisfying the event in Equation (11). By the union bound, with probability at least $1 - q(|z|)/p(|z|) \ge 3/4$, 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 the parameters of the GapLearn instance generated from z'.

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

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

The algorithm A is given access to samples drawn from $\mathcal{D}_{z,z'}$ and a size parameter 1^s, sets m to be $c\cdot(s+\log|z|)$, 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 $\mathcal{D}_{z,z'}$. 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.

In the same proof as Claim 7.9, 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 (11).

By contrast, suppose that for given $(\mathcal{D}_{z,z'}, 1^s)$, 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 (11) and A outputs 1 for $\mathcal{D}_{z,z'}$ with probability at least 1/3.

Then, we have

$$\Pr_{z'\sim\mathcal{Z}',S^1,\ldots,S^N,h}\left[\exists i \in [N] \text{ s.t. } \Pr_{(x,b)}\left[h^M(S^i,x,z)=b\right] \geq \frac{1}{2}+\frac{\gamma}{2}\right] \geq 1 - \frac{1}{4p(|z|)}. \tag{12}$$

Let w be random seeds sufficiently long for generating N sample sets of each size m from $\mathcal{D}_{z,z'}$ (with only negligible failure probability) and executing h^M. We define a test for the random seed w as $T_z(w) = 1$ if and only if with probability at least $1 - 1/2p(|z|)$ over $z' \sim \mathcal{Z}'$, there exist $i \in [N]$ and a sequence of advice $\alpha_1, \ldots, \alpha_m \in \{0,1\}^{\ell(n)}$ such that for a sample set $S^i = \{(x^i_j, b^i_j)\}_{j\in[m]}$, where each (x^i_j, b^i_j) is generated using the corresponding seed in w and α_j, 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_z be a *good* seed set defined as $G_z = \{w : T_z(w) = 1\}$. Since $\mathcal{D}_{z,z'}$ is samplable with ℓ advice, Equation (12) and Markov's inequality imply that

$$\Pr_w[w \in G_z] \geq \frac{1}{2}.$$

Notice that T_z is recursively enumerable given z. Thus, the randomness $w^* \in G_z$ (to obtain N sample sets and execute h) initially found via universal search satisfies that (i) $\mathrm{K}(w^*) \leq O(\mathrm{K}(G_z)) = O(|z|)$, and (ii) with probability at least $1 - 1/2p(|z|)$ over $z' \sim \mathcal{Z}'$, it holds that there exist $i^* \in [N]$ and $\alpha_1, \ldots, \alpha_m \in \{0,1\}^{\ell(n)}$ such that

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

The description size of the hypothesis $h^M(S^{i^*}, \text{-}, z)$ is at most

$$\begin{aligned}\mathrm{K}(w^*) + \mathrm{K}(i^*) + |\alpha_1| + \cdots + |\alpha_m| + O(|z|) &\leq m \cdot \ell(n) + O(|z|)\\ &\leq c \cdot (s + \log|z|) \cdot \ell(n) + O(|z|)\\ &\leq (2c+1) \cdot s \cdot \ell(n),\end{aligned}$$

by selecting large enough λ (recall that $s \geq \lambda(|z|)$). Thus, these instances are not Yes instances for $\mathrm{Gap}^{\varepsilon,\gamma}_{C\cdot\ell}\mathrm{Learn}[\ell]$, where $C = 2c+1$ is a constant depending only on γ. 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 (11).

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. □

7.5 One-Way Functions from Average-Case Hardness

In this section, we prove Theorems 1.5 and 1.10.

Theorem 7.13 (Theorem 1.10 Item 1)**.** *For every $\ell(n) = \omega(1)$, every $\varepsilon(n) = o(1)$, and every constant $\gamma \in [0, 1/4)$, there exists $C \geq 1$ such that if $\mathrm{Gap}^{\varepsilon(n),\gamma}_{C\cdot\ell(n)}\mathsf{F}[\ell(n)]\text{-CMMSA} \notin \mathsf{BPP}$, where n is the instance size, then $(\Pi, \mathcal{D}) \notin \mathsf{Avg}_{1/\mathsf{poly}}\mathsf{BPP}$ for some $\Pi \in \mathsf{NP}$ and samplable distribution $\mathcal{D}$.*

Proof. By Lemma 7.3, there exist an auxiliary-input function $f = \{f_x\}$ and a description-restricted context-sensitive FAIN reduction R from $\Pi := \mathrm{Gap}^{\varepsilon(n),\gamma}_{C\cdot\ell(n)}\mathsf{F}[\ell(n)]\text{-CMMSA}, \mathcal{D})$ 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 3.8. □

Theorem 7.14 (Theorem 1.10 Item 2)**.** *For every $\ell(n) = \omega(1)$, every $\varepsilon(n) = o(1)$, and every constant $\gamma \in [0, 1/4)$, there exists $C \geq 1$ such that if $(\mathrm{Gap}^{\varepsilon(n),\gamma}_{C\cdot\ell(n)}\mathsf{F}[\ell(n)]\text{-CMMSA}, \mathcal{D}) \notin \mathsf{Avg}_{1/p}\mathsf{BPP}$ for some samplable distribution $\mathcal{D}$ and polynomial p, where n is the instance size, then there exists an inifinitely-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 $\Pi = \mathrm{Gap}^{\varepsilon(n),\gamma}_{C\cdot\ell(n)}\mathsf{F}[\ell(n)]\text{-CMMSA}$. Let p be an arbitrary polynomial.

By Lemma 7.3, 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. Then, there exists a polynomial-time randomized algorithm M such that for every $n \in \mathbb{N}$,

$$\Pr_{r,r_f}\left[\Pr_M[M(g_n(r, r_f)) \in g_n^{-1}(g_n(r, r_f))] \geq 1 - \mathsf{negl}(n)\right] \geq 1 - 1/8p(n).$$

By Markov's inequality,

$$\Pr_{x\sim\mathcal{D}_n}\left[\Pr_{M,r_f}[M(x, f_x(r_f)) \in \mathcal{D}_n^{-1}(x) \times f_x^{-1}(f_x(r_f))] \geq 3/4\right] \geq 1 - 1/p(n), \tag{13}$$

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, i.e., $M(x, y) = (x, M'(x, y))$.

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 $M'(x, f_x(r_f))$ inverts $f_x(r_f)$ over the choice of r_f within an additive accuracy error $\pm 1/16$ and confidence probability at least $1 - \mathsf{negl}(n)$.

If the estimated probability is at least $5/8$, then A executes $R^{\mathcal{O}_{M'}(\cdot,x;\rho)}(x)$ and outputs the same answer. Otherwise, A outputs $\perp$. Recall that since M' halts in polynomial time, A can simulate $R^{\mathcal{O}_{M'}(\cdot,x;\rho)}(x)$ in polynomial time.

Since the failure of the empirical estimation occurs only with negligible probability, we first ignore it. If x is passed the empirical test, it holds that

$$\Pr_{M,r_f}[M(x, f_x(r_f)) \in \mathcal{D}_n^{-1}(x) \times f_x^{-1}(f_x(r_f))] \geq \frac{5}{8} - \frac{1}{16} > \frac{1}{2}.$$

In this case,

$$\Pr_A[A(x) = \Pi(x)] = \Pr_{R,\rho}[R^{\mathcal{O}_{M'}(\cdot,x;\rho)}(x) = \Pi(x)] \geq \frac{3}{4}.$$

Furthermore, by Equation (13), at least with probability $1 - 1/p(n)$ over $x \sim \mathcal{D}_n$, the instance x must pass the test.

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

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

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 are due to the failure of the empirical estimation.

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

□

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 trivial from the definition, Thus, we will see only the remaining direction.

Theorem 7.15 (Theorem 1.5 Item 2 $\Rightarrow$ Item 1)**.** *If there is no infinitely-often one-way function, then for every $\gamma \in (0,1)$, there exists a constant C such that for every $\ell = \omega(1)$, every $\varepsilon = o(1)$, every samplable distribution $\mathscr{D} = \{\mathscr{D}_n\}_n$, where each $\mathscr{D}_n$ is over triples of 1^n, 1^s, and a distribution of samples in $\{0,1\}^n \times \{0,1\}$, there exists a randomized algorithm that solves $\mathrm{Gap}^{\varepsilon,\gamma}_{C\cdot\ell}\mathrm{Learn}[\ell]$ on average with probability at least $1 - 1/p(n)$ over $\mathscr{D}_n$ for every $n \in \mathbb{N}$.*

Proof. 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, for each $n \in \mathbb{N}$ and a random seed r for sampling from $\mathscr{D}_n$, the distribution $\mathcal{D}_{n,r}$ represents the distribution over samples in $\{0,1\}^n \times \{0,1\}$ drawn from $\mathscr{D}_n$ by using r. Recall that $\mathcal{D}_{n,r}$ is samplable with advice complexity $\ell := \ell(n)$.

By Lemmas 7.5 and 7.7, there exist a randomized oracle machine h, a polynomial-time computable function $f = \{f_n\}$, and a constant $c := c_\gamma \geq 1$ such that for every large enough $n \in \mathbb{N}$, for every oracle $\mathcal{I}$ that inverts f_n and every $m \geq c \cdot (s^*(r) + \log n)$,

$$\Pr_r\left[\Pr_{S=\{(x^1,b^1),\ldots,(x^m,b^m)\},h}\left[\Pr_{(x,b)}\left[h^{\mathcal{I}}(S,x,1^n) = b\right] \geq \frac{3}{4} + \frac{\gamma}{4}\right] \geq \frac{3}{4}\right] \geq 1 - \frac{1}{4p(n)},$$

where $(x^1, b^1), \ldots, (x^m, b^m), (x, b) \sim \mathcal{D}_{n,r}$, and $s^*(r)$ is the minimum size polynomial-time program h such that

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

Suppose that there is no one-way function. Then, there exists a polynomial-time randomized algorithm M such that for every $n \in \mathbb{N}$,

$$\Pr_{M,r}[M(1^n, f_n(r)) \in f_n^{-1}(f_n(r))] \geq 1/2.$$

Thus,

$$\Pr_r\left[\Pr_{S=\{(x^1,b^1),\ldots,(x^m,b^m)\},h}\left[\Pr_{(x,b)}\left[h^M(S, x, 1^n) = b\right] \geq \frac{3}{4} + \frac{\gamma}{4}\right] \geq \frac{3}{4}\right] \geq 1 - \frac{1}{4p(n)}. \tag{14}$$

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, 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 $c\cdot(s+\log n)$, 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.

In the same proof as Claim 7.9, 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 (14).

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 emprical 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 (14) and A outputs 1 for $\mathcal{D}_{n,r}$ with probability at least $1/3$.

Then, we have

$$\Pr_{r\sim\mathcal{R},S^1,\ldots,S^N,h}\left[\exists i \in [N] \text{ s.t. } \Pr_{(x,b)}\left[h^M(S^i, x, 1^n) = b\right] \geq \frac{1}{2} + \frac{\gamma}{2}\right] \geq 1 - \frac{1}{4p(n)}. \tag{15}$$

Let w be random seeds sufficiently long for generating N sample sets of each size m from $\mathcal{D}_{n,r}$ (with only negligible failure probability) and executing h^M. We define a test for the random seed w as $T_n(w) = 1$ if and only if with probability at least $1 - 1/2p(n)$ over $r \sim \mathcal{R}$, there exist $i \in [N]$ and a sequence of advice $\alpha_1, \ldots, \alpha_m \in \{0,1\}^{\ell(n)}$ such that for a sample set $S^i = \{(x_j^i, b_j^i)\}_{j \in [m]}$, where each (x_j^i, b_j^i) is generated using the corresponding seed in w and α_j, 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 be a *good* seed set defined as $G_n = \{w : T_n(w) = 1\}$. Since $\mathcal{D}_{n,r}$ is samplable with ℓ advice, Equation (15) and Markov's inequality imply that

$$\Pr_w[w \in G_n] \geq \frac{1}{2}.$$

Notice that T_n is recursively enumerable given n. Thus, the randomness $w^* \in G_n$ (to obtain N sample sets and execute h) initially found via universal search satisfies that (i) $\mathrm{K}(w^*) \leq O(\mathrm{K}(G_n)) = O(\log n)$, and (ii) with probability at least $1 - 1/2p(n)$ over $r \sim \mathcal{R}$, it holds that there exist $i^* \in [N]$ and $\alpha_1, \ldots, \alpha_m \in \{0,1\}^{\ell(n)}$ such that

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

The description size of the hypothesis $h^M(S^{i^*}, \text{-}, 1^n)$ is at most

$$\begin{aligned}\mathrm{K}(w^*) + \mathrm{K}(i^*) + |\alpha_1| + \cdots + |\alpha_m| + O(\log n) &\leq m \cdot \ell(n) + O(\log n) \\ &\leq c \cdot (s + \log n) \cdot \ell(n) + O(\log n) \\ &\leq (2c+1) \cdot s \cdot \ell(n),\end{aligned}$$

where the last inequality holds by assuming $\log n \leq s$; otherwise, we can efficiently find the hypothesis of size s by brute-force search. Thus, these instances are not Yes instances for $\mathrm{Gap}^{\varepsilon,\gamma}_{C \cdot \ell}\mathrm{Learn}[\ell]$, where $C = 2c + 1$ is a constant depending only on γ. 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 (14).

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. □

8 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 larger functions ℓ? In our case, ℓ is a function that grows very slowly (specifically, $\ell(n) \leq \log\log\log n$).

- 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

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. ISBN: 978-0-521-42426-4.

[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.

[Gol11] Oded Goldreich. "A Sample of Samplers: A Computational Perspective on Sampling". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. Springer, 2011, pp. 302–332. DOI: 10.1007/978-3-642-22670-0_24.

[Hås01] Johan Håstad. "Some optimal inapproximability results". In: *J. ACM* 48.4 (2001), pp. 798–859. DOI: 10.1145/502090.502098.

[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.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. "Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs?" In: *SIAM J. Comput.* 37.1 (2007), pp. 319–357. DOI: 10.1137/S0097539705447372.

[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.

[KS15] Subhash Khot and Rishi Saket. "Approximating CSPs Using LP Relaxation". In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. 2015, pp. 822–833. DOI: 10.1007/978-3-662-47672-7_67.

[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.

[Mos10] Elchanan Mossel. "Gaussian Bounds for Noise Correlation of Functions". In: *Geometric and Functional Analysis (GAFA)* 19.6 (2010), pp. 1713–1756. DOI: 10.1007/s00039-010-0047-x.

[MR10] Dana Moshkovitz and Ran Raz. "Two-query PCP with subconstant error". In: *J. ACM* 57.5 (2010), 29:1–29:29. DOI: 10.1145/1754399.1754402.

[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.

[ODo14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. ISBN: 978-1-10-703832-5.

[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.

[RST12] Prasad Raghavendra, David Steurer, and Madhur Tulsiani. "Reductions between Expansion Problems". In: *Proceedings of the Conference on Computational Complexity (CCC)*. 2012, pp. 64–73. DOI: 10.1109/CCC.2012.43.

[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 Observing Lemma 4.8

This section heavily relies on the construction presented in [MR10], and thus we refer the original paper for backgrounds.

A.1 Syntax and Balanced Properties

First, we review some variants of bipartite LDRC introduced [MR10] that appear in the intermediate composition steps, where we only mention some syntax relevant for observing the balanced property (for instance, we omit notions required only for proving correctness of the original construction and even requirements of LDRCs). In addition, we assume *uniformity* in some sense, i.e., we assume that the bipartite LDRC is constructed for each given targeted tuples in *some* uniform ways, and we do not even mention about the targeted k tuples in the definitions. For details on formal definitions and uniformity, see the original paper [MR10].

Definition A.1 (Bipartite LDRC, simplified syntax)**.** *Let R be a finite set of alphabets for the original message, and let $k \in \mathbb{N}$. A bipartite LDRC for k-tuples is composed of*

$$\langle G = (A, B, E), V, \Omega, \Sigma_A, \Sigma_B, \mathrm{valid}, \mathrm{label}, \pi, \mathrm{eval} \rangle,$$

where

- *$G = (A, B, E)$ is a bipartite multigraph;*
- *$V \in \{A, B\}$ is a set of evaluating vertices;*
- *Ω is a finite set of labels;*

- Σ_A *and* Σ_B *are finite sets of alphabets for* A *and* B*, respectively;*
- $\mathrm{valid}\colon A \times \Sigma_A \to \{0,1\}$ *is a function that specifies valid alphabets for each left vertex;*
- $\mathrm{label}\colon E \to \Omega$ *is a function that assigns a label for each edge;*
- $\pi\colon A \times \Sigma_A \times \Omega \to \Sigma_B$ *is a function that represents the projection rule for each edge;*
- $\mathrm{eval}\colon V \times \Sigma_V \to R^k$ *is a function that evaluates each vertex in* V *with its assignments.*

When $V = A$ (resp. $V = B$), we say the LDRC has a left (resp. right) reader.

Definition A.2 (Point-Variant of Bipartite LDRC, simplified syntax)**.** *Let* R *be a finite set of alphabets for the code, and let* $k \in \mathbb{N}$*. A point variant of bipartite LDRC for* k*-tuples in* X *is composed of*

$$\langle G = (A, B, E), V, \Omega, \Sigma_A, \Sigma_B, \mathrm{valid}, \mathrm{label}, \pi, \mathrm{eval}, \mathrm{evalp}\rangle,$$

where the syntax is the same as Definition A.1 except

- $\mathrm{evalp}\colon W \times \Sigma_W \to R$*, where* $W \in \{A, B\} \setminus V$*, is a function that evaluates the other vertex set.*

When $V = A$ (resp. $V = B$), we say the point-variant of LDRC has a left+point (resp. right+point) reader.

We define the balanced property for each definition above.

Definition A.3 (Balanced Property for Left Reader)**.** *Let* R *be a finite set of alphabets for the code, let* X *be a finite set of indices, and let* S *be a subset of* k*-tuples whose element is in* X*. A bipartite LDRC for* k*-tuples*

$$\mathcal{G} = \langle G = (A, B, E), A, \Omega, \Sigma_A, \Sigma_B, \mathrm{valid}, \mathrm{label}, \pi, \mathrm{eval}\rangle$$

is said to have the balanced property on S *if there exists* $M \in \mathbb{N}$ *such that for every (list of)* k*-tuples in* S*, every edge* $e = (v, w) \in E$*, every* $y \in R^k$*, and every* $\beta \in \Sigma_B$*, it holds that*

$$|\{\alpha \in \Sigma_A : \mathrm{valid}(v, \alpha) = 1, \pi(v, \alpha, \mathrm{label}(e)) = \beta, \mathrm{eval}(v, \alpha) = y\}| = M.$$

In addition, a point-variant of bipartite LDRC for k*-tuples*

$$\mathcal{G} = \langle G = (A, B, E), A, \Omega, \Sigma_A, \Sigma_B, \mathrm{valid}, \mathrm{label}, \pi, \mathrm{eval}, \mathrm{evalp}\rangle$$

is said to have the balanced property on S *if there exists* $M \in \mathbb{N}$ *such that for every (list of)* k*-tuples in* S*, every edge* $e = (v, w) \in E$*, every* $y \in R^k$*, and every* $z \in R$ *it holds that*

$$|\{(\alpha, \beta) \in \Sigma_A \times \Sigma_B : \mathrm{valid}(v, \alpha) = 1, \pi(v, \alpha, \mathrm{label}(e)) = \beta, \mathrm{eval}(v, \alpha) = y, \mathrm{evalp}(w, \beta) = z\}| = M.$$

Definition A.4 (Balanced Property for Right Reader)**.** *Let* R *be a finite set of alphabets for the code, let* X *be a finite set of indices, and let* S *be a subset of* k*-tuples whose element is in* X*. A bipartite LDRC for* k*-tuples*

$$\mathcal{G} = \langle G = (A, B, E), B, \Omega, \Sigma_A, \Sigma_B, \mathrm{valid}, \mathrm{label}, \pi, \mathrm{eval}\rangle$$

is said to have the balanced property on S *if there exists* $M \in \mathbb{N}$ *such that for every (list of)* k*-tuples in* S*, every edge* $e = (v, w) \in E$*, every* $y \in R^k$*, it holds that*

$$|\{(\alpha, \beta) \in \Sigma_A \times \Sigma_B : \mathrm{valid}(v, \alpha) = 1, \pi(v, \alpha, \mathrm{label}(e)) = \beta, \mathrm{eval}(w, \beta) = y\}| = M.$$

In addition, a point-variant of bipartite LDRC for k-tuples

$$\mathcal{G} = \langle G = (A, B, E), B, \Omega, \Sigma_A, \Sigma_B, \mathrm{valid}, \mathrm{label}, \pi, \mathrm{eval}, \mathrm{evalp} \rangle$$

is said to have the balanced property on S if there exists $M \in \mathbb{N}$ such that for every (list of) k-tuples in S, every edge $e = (v, w) \in E$, every $y \in R^k$, and every $z \in R$ it holds that

$$|\{(\alpha, \beta) \in \Sigma_A \times \Sigma_B : \mathrm{valid}(v, \alpha) = 1, \pi(v, \alpha, \mathrm{label}(e)) = \beta, \mathrm{eval}(w, \beta) = y, \mathrm{evalp}(v, \alpha) = z\}| = M.$$

When S is implicit, we let S be the whole set of k-tuples whose element is in X.

In the following sections, we will observe that every base LDRC has the balanced property and the composition steps preserves it in [MR10]. The construction consists of the following 15 steps, where we omit the term "bipartite" from bipartite LDRC.

1. Construct Reed–Muller-based LDRC with Right Reader (RM-RR)
   (a) Construct Reed–Muller-based LDRC with Light Reader (RM-LR);
   (b) Apply power reduction;
   (c) Apply right degree reduction;
   (d) Switch sides;
   (e) Apply right degree reduction.
2. Construct Reed–Muller-based LDRC with Right+Point Reader (RM-RPR)
   (a) Construct Reed–Muller-based LDRC with Light+Point Reader (RM-LPR);
   (b) Apply power reduction;
   (c) Apply right degree reduction;
   (d) Switch sides;
   (e) Apply right degree reduction.
3. Compose (outer) RM-RR with (inner) RM-RPR to construct new RM-RR
4. Construct Concatenated-Code-based LDRC with Light Reader (RM$\diamond$Had-LR)
   (a) Construct Hadamard-based LDRC with Light Reader (Had-LR);
   (b) Transform Had-LR to RM$\diamond$Had-LR;
   (c) Apply right degree reduction.
5. Compose (outer) RM-RR of Item 3 with (inner) RM$\diamond$Had-LR.

A.2 Base Cases

We observe the balanced properties of the three base cases.

A.2.1 Construct RM-LR (Item 1a)

In RM-LR, $R = \mathbb{F}$ and $X = \mathbb{F}^m$ for a finite field $\mathbb{F}$ and $m \in \mathbb{N}$. The A-alphabet Σ_A is a set of (descriptions of) degree-d 4-variate polynomials over $\mathbb{F}$, where it holds that $d > k+1$. The B-alphabet Σ_B is $\mathbb{F}$. The label set Ω is $\mathbb{F}^4$. The validity test always accepts, i.e., valid $\equiv 1$. The projection $\pi\colon A \times \Sigma_A \times \Omega \to \Sigma_B$ is defined as $\pi(a, \sigma_a, p) = \sigma_a(p)$ (recall that σ_a is a 4-variate polynomial). Recall that $V = A$. For each $a \in A$, the evaluation is defined as $\mathrm{eval}(a, \sigma_a) = (\sigma_a(p_1), \ldots, \sigma_a(p_k))$, where $p_1, \ldots, p_k \in \mathbb{F}^k$ and they are distinct. In addition, for each $e = (a, b) \in E$, it holds that $p_e := \mathrm{label}(e) \notin \{p_1, \ldots, p_k\}$.

Therefore, the condition of the balanced property is written as for each $y = (y_1, \ldots, y_k) \in R^k$ and $\sigma_b \in \Sigma_B$,

$$|\{\sigma_a : \sigma_a(p_1) = y_1 \wedge \cdots \sigma_a(p_k) = y_k \wedge \sigma_a(p_e) = \sigma_b\}|$$

is the same regardless of e. This is verified since the degree of σ_a is greater than $k+1$, and $p_1, \ldots, p_k, p_e$ are distinct.

A.2.2 Construct RM-LPR (Item 2a)

The balanced property is verified in the similar way as that of RM-LR.

In RM-LPR, (as in RM-LR) $R = \mathbb{F}$ and $X = \mathbb{F}^m$ for a finite field $\mathbb{F}$ and $m \in \mathbb{N}$. The A-alphabet Σ_A is a set of (descriptions of) degree-d 4-variate polynomials over $\mathbb{F}$, where it holds that $d > k+1$. The B-alphabet Σ_B is $\mathbb{F}$. The label set Ω is $\mathbb{F}^4$. The validity test always accepts, i.e., valid $\equiv 1$. The projection $\pi\colon A \times \Sigma_A \times \Omega \to \Sigma_B$ is defined as $\pi(a, \sigma_a, p) = \sigma_a(p)$ (recall that σ_a is a 4-variate polynomial). Recall that $V = A$. For each $a \in A$, the evaluation is defined as $\mathrm{eval}(a, \sigma_a) = (\sigma_a(p_1), \ldots, \sigma_a(p_k))$, where $p_1, \ldots, p_k \in \mathbb{F}^k$ and they are distinct. For each $b \in B$, the point evaluation is defined as $\mathrm{evalp}(b, \sigma_b) = \sigma_b$. In addition, for each $e = (a, b) \in E$, it holds that $p_e := \mathrm{label}(e) \notin \{p_1, \ldots, p_k\}$.

Namely, the condition of the balanced property is written as for each $y = (y_1, \ldots, y_k) \in R^k$ and each $z \in R$,

$$|\{\sigma_a : \sigma_a(p_1) = y_1 \wedge \cdots \sigma_a(p_k) = y_k \wedge \sigma_a(p_e) = z\}|$$

is the same regardless of e. This is verified since the degree of σ_a is greater than $k+1$, and $p_1, \ldots, p_k, p_e$ are distinct.

A.2.3 Construct Had-LR (Item 4a)

In Had-LR, $R = \mathbb{F}$ and $X = \mathbb{F}^m$ for a finite field $\mathbb{F}$ and $m \in \mathbb{N}$. We define a set $S \subseteq (\mathbb{F}^m)^k$ as

$$S = \left\{(x_1, \ldots, x_k) \in (\mathbb{F}^m)^k : x_1, \ldots, x_k \text{ are linearly independent}\right\},$$

and consider the balanced property on S. Below, we consider an arbitrary k-tuple in S.

The A-alphabet Σ_A is a set of (descriptions of) $(k+2)$-variate linear functions over $\mathbb{F}$. The B-alphabet Σ_B is $\mathbb{F}$. The label set Ω is $\mathbb{F}^{k+2}$. The validity test always accepts, i.e., valid $\equiv 1$. The projection $\pi\colon A \times \Sigma_A \times \Omega \to \Sigma_B$ is defined as $\pi(a, \sigma_a, p) = t_a^{-1} \cdot \sigma_a(p)$, where $t_a \in \mathbb{F} \setminus \{0\}$ (recall that σ_a is a $(k+2)$-variate linear function). Recall that $V = A$. For each $a \in A$, the evaluation is defined as $\mathrm{eval}(a, \sigma_a) = (\sigma_a(p_1), \ldots, \sigma_a(p_k))$, where $p_1, \ldots, p_k \in \mathbb{F}^k$. In addition, for each $e = (a, b) \in E$, it holds that $p_e := \mathrm{label}(e)$ and $p_1, \ldots, p_k$ are linearly independent.

Therefore, the condition of the balanced property (on S) is written as for each $y = (y_1, \ldots, y_k) \in R^k$ and $\sigma_b \in \Sigma_B$,

$$|\{\sigma_a : \sigma_a(p_1) = y_1 \wedge \cdots \sigma_a(p_k) = y_k \wedge \sigma_a(p_e) = t_a \cdot \sigma_b\}|$$

is the same regardless of $e = (a, b)$. This is verified since σ_a is $(k+2)$-variate, and $p_1, \ldots, p_k, p_e$ are linearly independent.

Note that when the targeted tuple is not in S, i.e., it contains linear dependent vectors, the corresponding evaluated points $p_1, \ldots, p_k$ can be linear dependent.

A.3 Manipulations

We observe that each manipulation preserves the balanced property.

A.3.1 Power Reduction (Items 1b and 2b)

Consider RM-LR in Item 1a or RM-LPR in Item 2a. Recall that they satisfy the balanced property. The purpose of power reduction is to reduce (the valid encodings of) the degree of A alphabets logarithmically.

Let A-alphabets are 4-variates polynomials of degree d over $\mathbb{F}$. A new A-alphabets are $4b$-variate polynomials of degree d' over $\mathbb{F}$, where $b, d' \in \mathbb{N}$ are determined by the parameters of the original code, and it holds that $d' \geq k+1$ (which follows from the choice of parameters of [MR10, Section 9.6]). The constraint graph and components are the same as the originals except the following: When a 4-variate polynomial p (of degree d) is evaluated on a point $p \in \mathbb{F}^4$ in the original code, we instead evaluate the corresponding $4b$-variate polynomial p' (of degree d') on $\phi(p) \in \mathbb{F}^{4d'}$, where

$$\phi(p_1, p_2, p_3, p_4) = \left(p_1^{2^0}, p_1^{2^1}, p_1^{2^2}, \ldots, p_1^{2^b}, \ldots, p_4^{2^0}, p_4^{2^1}, p_4^{2^2}, \ldots, p_4^{2^b}\right).$$

The original code evaluates a polynomial on distinct points, and so does the new code since $\phi(p) \neq \phi(p')$ for $p \neq p'$. Thus, by the same argument as Appendices A.2.1 and A.2.2, the new code has the balanced property.

A.3.2 Right Degree Reduction (Items 1c, 1e, 2c, 2e, and 4c)

Let $G = (A, B, E)$ be the original constraint (regular) bipartite graph. The purpose of this manipulation is to reduce the right degree. The idea is the following. We copy each B-vertex as many as the original right degree d_r and then connect them to d_r incoming edges according to a bipartite expander graph of size d_r without changing the constraints, labels, and so on. Since it does not change anything other than the graph structure, the manipulation preserves the balanced property.

A.3.3 Switching Sides (Items 1d and 2d)

The purpose of this step to transform RM-LR and RM-LPR into RM-RR and RM-RPR, respectively. Namely, we switch the evaluation side V from A to B.

First, we see the case of RM-LR. Let

$$\mathcal{G} = \langle G = (A, B, E), A, \Omega, \Sigma_A, \Sigma_B, 1, \mathrm{label}, \pi, \mathrm{eval} \rangle$$

be the original RM-LR. Then we transform $\mathcal{G}'$ to RM-RR

$$\mathcal{G}' = \langle G' = (A', B', E'), B, \Omega', \Sigma_{A'}, (\Sigma_{B'} :=) \Sigma_A, \mathrm{valid}', \mathrm{label}', \pi', \mathrm{eval}' \rangle,$$

where $A' = B$, $B' = A$, and $E' = \{(b, a) : (a, b) \in E\}$, i.e., G' is obtained by switching A and B.

Let d_r be the right degree of G. The label set Ω' is $[d_r]$, and for each $e = (b, a) \in E'$, the label $\mathrm{label}'(e) = i$ if (a, b) is the i-th edge coming into b in G. The A'-alphabet set is a set of mappings

from Ω' to Σ_A (i.e., d_r polynomials over $\mathbb{F}$). The projection rule is $\pi(b, \sigma_a, \xi) = \sigma_a(\xi) \in \Sigma_A$ $(= \Sigma_{B'})$. The evaluation eval' is the same as the original left evaluator, i.e., $\mathrm{eval}'(a, \sigma_a) = \mathrm{eval}(a, \sigma_a)$.

The validity test is defined as follows: For each $b \in A' (= B)$ and $\xi \in \Omega' (= [d_r])$, we can uniquely determine $a \in A$ such that (a, b) is ξ-th edge coming into b in G, and let $p_{b,\xi} := \mathrm{label}((a, b)) \in \mathbb{F}^4$. The validity test valid' accepts (b, σ_b) iff $\sigma_b(\xi)(p_{b,\xi})$ is consistent (i.e., the same) for all $\xi \in \Omega'$ (recall that $\sigma_b(\xi)$ is a 4-variate polynomial).

Remember that for every $(a, b) \in E$, the evaluated points in $\mathrm{eval}(a, \text{-})$ and $\mathrm{label}((a, b))$ are distinct. Thus, for each $(b, a) \in E'$ and $\xi \in \Omega$, the evaluated points in $\mathrm{eval}'(a, \sigma_b(\xi))$ and $p_{b,\xi}$ are distinct. Namely, for each $(b, a) \in E'$ and $\xi \in \Omega'$, the number of $(\sigma_b(\xi), \sigma_a) \in \Sigma_A \times \Sigma_A$ with $\pi(b, \xi) = \sigma_b(\xi) = \sigma_a$ that returns $y \in R^{k+1}$ on the above $k + 1$ evaluated points is the same regardless of y. Since the same holds for other $d_r - 1$ labels $\xi' \in \Omega' \setminus \{\xi\}$, there are the same number of polynomials $\sigma_b(\xi')$ that are consistent with $\sigma_b(\xi)(p_{b,\xi})$. Thus, the balanced property holds.

The construction of RM-RPR is the same except the point evaluation defined as

$$\mathrm{evalp}'(b, \sigma_b) = \mathrm{evalp}(b, \sigma_p)$$

where evalp is the point evaluation function of the original RM-LPR. Thus, the balanced property is observed in the same way as RM-RR.

A.3.4 Transfoming Had-LR to RM⋄Had-LR (Item 4b)

First, we review the concatenated code RM⋄Had constructed from (inner) Reed-Muller code and (outer) Hadamard code.

Let $\mathbb{F}$ be a finite field for alphabets of Reed-Muller code, and let $\mathbb{L} \leq \mathbb{F}$ be a prime subfield for alphabets of Reed-Muller code. Let τ be the extension degree $[\mathbb{F} : \mathbb{L}]$. Let $m \in \mathbb{N}$ be dimension. We select the degree d of polynomials for Reed-Muller code so that $d > k + 1$. Let M be the number of degree-d m-variate monomials. Then, we consider the Reed-Muller encoding $\mathbb{F}^m \to \mathbb{F}^{|\mathbb{F}^m|}$ and the Hadamard encoding $\mathbb{L}^\tau \to \mathbb{L}$.

The concatenation of the two encoding is a linear function over $\mathbb{L}$, where an original message p in $\mathbb{F}^M$ (regarded as $\mathbb{L}^{M\tau}$) is encoded by the Reed-Muller encoding resulting in the codeword in $\mathbb{F}^{|\mathbb{F}^m|}$ and then each symbol in $\mathbb{F}$ regarded as $\mathbb{L}^\tau$ is further encoded by the Hadamard encoding. Namely, each symbol in the codeword is indexed by $(x, y) \in \mathbb{F}^m \times \mathbb{L}^\tau$ and it is obtained as $\langle p(x), y \rangle_{\mathbb{L}}$, where p is regarded as the degree-d polynomial over $\mathbb{F}$, the value $p(x)$ is regarded as the element in $\mathbb{L}^\tau$, and $\langle , \rangle_{\mathbb{L}}$ is the inner product over $\mathbb{L}$. By the linearity, for each $(x, y) \in \mathbb{F}^m \times \mathbb{L}^\tau$, there is the coefficient vector $e_{x,y} \in \mathbb{L}^{M\tau}$ such that the codeword at position (x, y) for the original message $p \in \mathbb{L}^{M\tau}$ is $\langle p, e_{x,y} \rangle$.

Now, we present how the base Had-LR (whose alphabet set is $R = \mathbb{L}$ and index set is $X = \mathbb{L}^{M\tau}$) is transformed into an LDRC based on the concatenation codes RM⋄Had-LR. The alphabet set is $R' = \mathbb{L}$, and the new index set is $X' = \mathbb{F}^m \times \mathbb{L}^\tau$. For each k-tuple $((x_1, y_1), \ldots, (x_k, y_k)) \in X'^k$ in the list, we construct a k-tuple $(e_{x_1,y_1}, \ldots, e_{x_k,y_k}) \in X^k$ and invokes the construction of Had-LR for the resulting list of X^k. The resulting LDRC $\mathcal{G}$ is the outcome of the construction of RM⋄Had-LR.

Define $S \subseteq X'^k$ as

$$S := \{((x_1, y_1), \ldots, (x_k, y_k)) : x_1, \ldots, x_k \text{ are distinct}\}.$$

Then we observe the balanced property of RM⋄Had-LR on S. Since Had-LR has the balanced property on k-tuples that are linearly independent (see Appendix A.2.3), it suffices to show that $e_{x_1,y_1}, \ldots, e_{x_k,y_k}$ are linearly independent as long as $x_1, \ldots, x_k$ are distinct.

Let $(x_1, y_1), \ldots, (x_k, y_k) \in X'$, and suppose that $x_1, \ldots, x_k$ are distinct. Since the degree d for the Reed-Muller code is greater than $k + 1$, for every $v = (v_1, \ldots, v_k) \in \mathbb{F}^k$, there exists a message

$p \in \mathbb{L}^{M\tau}$ (regarded as a degree-d polynomial over $\mathbb{F}$) such that $(p(x_1), \dots p(x_k)) = v$. Namely, by selecting p appropriately, we can let each $p(x_i)$ be an arbitrary linear function over $\mathbb{L}$, where p is regarded as the corresponding polynomial over $\mathbb{F}$. Thus, for each $i \in [k]$, we can select $p_i \in \mathbb{L}^{M\tau}$ so that

$$(\langle e_{x_1,y_1}, p_i\rangle_{\mathbb{L}^{M\tau}}, \dots, \langle e_{x_k,y_k}, p_i\rangle_{\mathbb{L}^{M\tau}})^T = (\langle y_1, p_i(x_1)\rangle_{\mathbb{L}}, \dots, \langle y_k, p_i(x_k)\rangle_{\mathbb{L}})^T = e_i,$$

where $e_i \in (\mathbb{L}^{M\tau})^k$ is the unit vector whose i-th element is 1.

Therefore, by letting $E = (e_{x_1,y_1}, \dots, e_{x_k,y_k}) \in (\mathbb{L}^{M\tau})^{k\times k}$ and $P = (p_1, \dots, p_k) \in (\mathbb{L}^{M\tau})^{k\times k}$, we have $P^T E = I_k$, where $I_k \in (\mathbb{L}^{M\tau})^{k\times k}$ is the identity matrix. Namely, E has the inverse matrix P^T and full rank. Thus, $e_{x_1,y_1}, \dots, e_{x_k,y_k}$ are linearly independent, as desired.

A.4 Compositions

Finally, we observe each composition also preserves the balanced property, which completes the proof of Lemma 4.8.

A.4.1 Composing (outer) RM-RR with (inner) RM-RPR (Item 3)

In the composed RM-RR, $R = \mathbb{F}$ and $X = \mathbb{F}^m$ for a finite field $\mathbb{F}$ and $m \in \mathbb{N}$. The construction algorithm performs as follows:

On the given list $\mathcal{L} = \{(x_{i,1}, \dots, x_{i,k})\}_{i\in[N]}$ of k-tuples, it first invokes the construction of the (outer) RM-RR for $\mathcal{L}$ in Item 1e and obtain

$$\mathcal{G}_{out} = \langle G_{out} = (A_{out}, B_{out}, E_{out}), B_{out}, \Omega_{out}, \Sigma_{A_{out}}, \Sigma_{B_{out}}, \mathrm{valid}_{out}, \mathrm{label}_{out}, \pi_{out}, \mathrm{eval}_{out}\rangle.$$

Note that G_{out} is regular. Let d_l and d_r be the left and right degree in G_{out}, respectively. Recall that $\Sigma_{B_{out}}$ is a set of degree-d w-variate polynomials over $\mathbb{F}$ for some $d, w \in \mathbb{N}$.

For each $b_{out} \in B_{out}$, we define two types of query points in $\mathbb{F}^w$ for each $b_{out} \in B_{out}$.

- Vertex queried points (k points). Recall that $\mathrm{eval}_{out}(b_{out}, \text{-})$ evaluates $\sigma_{b_{out}} \in \Sigma_{B_{out}}$ on k distinct points $p_1, \dots, p_k \in \mathbb{F}^w$. We call them vertex queried points.
- Edge queried points (d_r points). For each $a_{out} \in A_{out}$ with $e_{out} = (a_{out}, b_{out}) \in E_{out}$ and for $\sigma_{a_{out}} \in \Sigma_{A_{out}}$, the validity test valid_{out} evaluates $\sigma_{a_{out}}(\mathrm{label}_{out}(e_{out}))$ (which is supposed to be $\sigma_{b_{out}}$ by the projection rule) on a point $p_{a_{out}} \in \mathbb{F}_w$. We call it an edge queried point. Since there are d_r many a_{out}'s, there are d_r edge queried points.

For each $b_{out} \in B_{out}$, the construction algorithm invokes the (inner) RM-RPR for the $k + d_r$ query points above (as a $(k + d_r)$-tuple) and obtain

$$\mathcal{G}_{in}^{b_{out}} = \langle G_{in}^{b_{out}} = (A_{in}, B_{in}, E_{in}^{b_{out}}), B_{in}, \Omega_{in}, \Sigma_{A_{in}}, \Sigma_{B_{in}}, \mathrm{valid}_{in}^{b_{out}}, \mathrm{label}_{in}^{b_{out}}, \pi_{in}^{b_{out}}, \mathrm{eval}_{in}^{b_{out}}, \mathrm{evalp}_{in}^{b_{out}}\rangle,$$

where we used the fact that the vertex sets, labels, and alphabets are universal regardless of the given tuple in the inner construction. Then it constructs a new RM-RR

$$\mathcal{G} = \langle G = (A, B, E), B, \Omega, \Sigma_A, \Sigma_B, \mathrm{valid}, \mathrm{label}, \pi, \mathrm{eval}\rangle$$

as follows:

The vertex sets are $A = A_{out} \times A_{in}$ and $B = B_{out} \times B_{in}$. The edge set is

$$E = \left\{(\langle a_{out}, a_{in}\rangle, \langle b_{out}, b_{in}\rangle) \in A \times B : (a_{out}, b_{out}) \in E_{out} \text{ and } (a_{in}, b_{in}) \in E_{in}^{b_{out}}\right\}.$$

The label set is $\Omega = [d_l] \times \Omega_{in}$. For each edge $e = (\langle a_{out}, a_{in}\rangle, \langle b_{out}, b_{in}\rangle) \in E$, the label is assigned as $\mathrm{label}(e) = (i, \mathrm{label}_{in}^{b_{out}}(a_{in}, b_{in}))$, where $i \in [d_l]$ such that (a_{out}, b_{out}) is the i-th outgoing edge from a_{out} in G_{out}.

The B-alphabet set is $\Sigma_B = \Sigma_{B_{in}}$, which is a set of degree-d' μ-variate polynomials over $\mathbb{F}$ for some $d', \mu \in \mathbb{N}$. In the actual construction, it holds that $d' > k + 2$ (which follows from the choice of parameters of [MR10, Section 9.6]). The A-alphabet set is a set of mappings from Ω to Σ_B. The projection rule is defined as $\pi(a, \sigma_a, \xi) = \sigma_a(\xi)$ for $a \in A$, $\sigma_a \in \Sigma_A$, and $\xi \in \Omega$.

Let $b = \langle b_{out}, b_{in}\rangle \in B$. For b_{in} and $\sigma_b \in \Sigma_B$ $(= \Sigma_{B_{in}})$, the inner RM-RR evaluates σ_b on distinct $k + d_l$ points $p_1^{b_{in}}, \ldots, p_k^{b_{in}}$ (which correspond to k vertex queried points) and $q_1^{b_{in}}, \ldots, q_{d_l}^{b_{in}}$ (which correspond to d_l edge points). Then, $\mathrm{eval}(b, \sigma_b) = (\sigma_b(q_1^{b_{in}}), \ldots, \sigma_b(q_k^{b_{in}}))$ (i.e., the first k values in $\mathrm{eval}_{in}^{b_{out}}(b_{in}, \sigma_b)$).

Finally, we specify the validity test. Let $a = \langle a_{out}, a_{in}\rangle \in A$ and $\sigma_a \in \Sigma_A$. We consider the following depth-2 tree. There are $|\Omega_{out}|$ depth-1 internal nodes, which are indexed by $\xi_{out} \in \Omega_{out}$, and each node indexed by $\xi_{out} \in \Omega_{out}$ has leaves (as descendants) indexed by $\Omega^{\xi_{out}} \subseteq \Omega$, where

$$\Omega^{\xi_{out}} = \{(i, \xi_{in}) : \text{the } i\text{-th outgoing edge } e \text{ from } a_{out} \text{ in } G_{out} \text{ satisfies } \mathrm{label}_{out}(e) = \xi_{out}\}.$$

Namely, $\{\Omega^{\xi_{out}}\}_{\xi_{out} \in \Omega_{out}}$ is a partition of Ω and thus the leaves can be identified with Ω. In fact, the number of descendants are the same for each level regardless of the choice of a (see [MR10, Section 10]). Each leaf $\xi \in \Omega$ has two ancestors $root, \xi_{out}$ (such that $(root, \xi_{out})$ and (ξ_{out}, ξ) are edges) and assigns evaluation points $p_{root}^{\xi}, p_{\xi_{out}}^{\xi} \in \mathbb{F}^{\mu}$ to $root, \xi_{out}$, respectively as follows: Let $\xi = (i, \xi_{in})$. Then i specifies the neighborhood b_{out} in G_{out}. Then the edge queried points on b_{out} must contain the corresponding query point. Let p_{root}^{ξ} be the corresponding evaluated point for $\sigma_a(\xi)$ where $\sigma_a \in \Sigma_A$. We also define $p_{\xi_{out}}^{\xi}$ as the evaluation point that $\mathrm{valid}_{in}^{b_{out}}(a_{in}, \sigma_{a_{in}})$ performs for $\sigma_{a_{in}}(\xi_{in})$ (where $\sigma_{in} \in \Sigma_{A_{in}}$, and $\sigma_{a_{in}}(\xi_{in})$ is a μ-variate polynomial). For $\sigma_a \in \Sigma_A$, the test $\mathrm{valid}(a, \sigma_a)$ first evaluates $\sigma_a(\xi)(p_{root}^{\xi})$ and $\sigma_a(\xi)(p_{\xi_{out}}^{\xi})$ for each $\xi \in \Omega$ and its ancestors $root, \xi_{out}$ and outputs 1 if and only if these values (in $\mathbb{F}$) are consistent for all points in the tree.

Now, we observe the balanced property. Let $e = (a, b) \in E$, where $a = \langle a_{out}, a_{in}\rangle$ and $b = \langle b_{out}, b_{in}\rangle$, and $\xi = \mathrm{label}(e)$. For $(\sigma_a, \sigma_b) \in \Sigma_A \times \Sigma_B$ satisfying the projection rule, i.e., $\sigma_a(\xi) = \sigma_b$, the evaluation $\mathrm{eval}(b, \sigma_b)$ and the validity test $\mathrm{valid}(a, \sigma_a)$ evaluates $\sigma_a(\xi)$ on $k + 2$ different points $p_1^{b_{in}}, \ldots, p_k^{b_{in}}, p_{root}^{\xi}, p_{\xi_{in}}^{\xi}$. Since the degree of σ_a is larger than $d+2$, for each $v = (v_1, \ldots, v_{k+2}) \in \mathbb{F}^{k+2}$, the number of $(\sigma_a(\xi), \sigma_b)$ that takes the values v on the evaluated points are the same regardless of v. For each v and other $|\Omega| - 1$ labels $\xi' \in \Omega \setminus \{\xi\}$, the number of $\sigma_a(\xi')$ that are consistent with $\sigma_a(\xi)$ in the validity test is the same (where $\sigma_a(\xi')$ is evaluated on two distinct points). Since the structure of the tree in the validity test is the same regardless of a, the balanced property holds.

A.4.2 Composing (outer) RM-RR with (inner) RM◇Had-LR (Item 5)

This step yields the final construction of LDRC in Theorem 4.7. The alphabet set and index set is the same as RM◇Had-LR, i.e., $R = \mathbb{L}$ and $X = \mathbb{F}^m \times \mathbb{L}$, where $\mathbb{F}$ is a finite field, $\mathbb{L} \leq \mathbb{F}$ with $\tau = [\mathbb{F} : \mathbb{L}]$, and $m \in \mathbb{N}$.

For a given list $\{(\langle x_{i,1}, y_{i,1}\rangle, \ldots, \langle x_{i,k}, y_{i,k}\rangle)\}_{i \in [N]}$, where $\langle x_{i,j}, y_{i,j}\rangle \in X$ for each i, j, the construction algorithm first invokes that of the outer RM-RR in Appendix A.4.1 for $\{(x_{i,1}, \ldots, x_{i,k})\}_{i \in [N]}$ and obtain

$$\mathcal{G}_{out} = \langle G_{out} = (A_{out}, B_{out}, E_{out}), B_{out}, \Omega_{out}, \Sigma_{A_{out}}, \Sigma_{B_{out}}, \mathrm{valid}_{out}, \mathrm{label}_{out}, \pi_{out}, \mathrm{eval}_{out}\rangle.$$

Let d_l be the left degree of G_{out}. Recall that $\Sigma_{B_{out}}$ is a set of degree-d w-variate polynomials over $\mathbb{F}$ for some $d, w \in \mathbb{N}$.

For each $I = (a_{out}, \xi) \in A_{out} \times \Omega_{out}$ and each $y \in \mathbb{L}^\tau$ (regarded as an index), we define two types of evaluation points as follows:

- Vertex queried points (kd_l points). For each b_{out} such that $(a_{out}, b_{out}) \in E_{out}$ (there are d_l such points), $\mathrm{eval}_{out}(b_{out}, \sigma_{b_{out}})$ evaluates $\sigma_{b_{out}}$ on k distinct points $p_1, \ldots, p_k \in \mathbb{F}^w$ to obtain the codeword at positions $x_{i,1}, \ldots, x_{i,k}$ for some $i \in [N]$. We call k points $\langle p_1, y_{i,1}\rangle, \ldots, \langle p_k, y_{i,k}\rangle \in \mathbb{F}^w \times \mathbb{L}^\tau$ vertex queried points.
- Path queried points (2 points). Recall that $\mathrm{valid}_{out}(a_{out}, \sigma_{a_{out}})$ evaluates $\sigma_{a_{out}}(\xi)$ (that is a w-variate polynomial) on two distinct points q, q' that are different from all of $p_1, \ldots, p_k$ above (regardless of b_{out}). We define path queried points as $\langle q, y\rangle$ and $\langle q', y\rangle$.

For each $I = (a_{out}, \xi) \in A_{out} \times \Omega_{out}$, the construction algorithm invokes that of the inner RM$\diamond$Had-LR for the list $\{z_{y,1}, \ldots, z_{y,kd_l+2}\}_{y \in \mathbb{L}^\tau}$ where $z_{y,1}, \ldots, z_{y,kd_l+2}$ are the above kd_l+2 queried points on I that are indexed by y. Then, it obtains

$$\mathcal{G}^I_{in} = \langle G^I_{in} = (A_{in}, B_{in}, E^I_{in}), A_{in}, \Omega_{in}, \Sigma_{A_{in}}, \Sigma_{B_{in}}, \mathrm{valid}^I_{in}, \mathrm{label}^I_{in}, \pi^I_{in}, \mathrm{eval}^I_{in}\rangle,$$

where we used the universality of vertices, labels, and alphabets, and then produces the edge reading LDRC

$$\mathcal{G} = \langle G = (A, B, E), \Sigma_A, \Sigma_B, \{\pi_e\}_{e \in E}, \{\mathrm{valid}_a\}_{a \in A}, \{\mathrm{eval}_e\}_{e \in E}\rangle$$

defined as follows[5]:

The vertex sets are $A = A_{out} \times A_{in}$ and $B = B_{out} \times B_{in}$. The edge set is

$$E = \left\{(\langle a_{out}, a_{in}\rangle, \langle b_{out}, b_{in}\rangle) \in A \times B : e_{out} = (a_{out}, b_{out}) \in E_{out} \text{ and } (a_{in}, b_{in}) \in E^{(a_{out}, \mathrm{label}_{out}(e_{out}))}_{in}\right\}.$$

For convenience, we introduce a label set $\Omega = \Omega_{out} \times \Omega_{in}$. Then, for each $e = (\langle a_{out}, a_{in}\rangle, \langle b_{out}, b_{in}\rangle) \in E$, its label is defined as $\mathrm{label}(e) = (\mathrm{label}_{out}(e_{out}), \mathrm{label}^I_{in}(e_{in}))$, where $e_{out} = (a_{out}, b_{out})$, $e_{in} = (a_{in}, b_{in})$, and $I = (a_{out}, \mathrm{label}_{out}(e_{out}))$.

The B-alphabet set is $\Sigma_B = \Sigma_{B_{in}}$ $(= \mathbb{L})$. The A-alphabet set Σ_A is a set of mappings from Ω_{out} to $\Sigma_{A_{in}}$, i.e., w'-variate linear functions over $\mathbb{L}$ for some $w' \in \mathbb{N}$. For each $e = (a, b) \in E$, the projection rule is defined as $\pi_e(\sigma_a) = \pi^I_{in}(\sigma_a(\xi_{out}), \xi_{in}) = \sigma_a(\xi_{out})(\xi_{in})$ for $\sigma_a \in \Sigma_A$, where $\mathrm{label}(e) = (\xi_{out}, \xi_{in})$, $a = \langle a_{out}, a_{in}\rangle$, and $I = (a_{out}, \xi_{out})$.

For each $I = (a_{out}, \xi_{out}) \in A_{out} \times \Omega_{out}$ and for each $a_{in} \in A_{in}$ and $\sigma_{a_{in}} \in \Sigma_{A_{in}}$ $(= \Sigma_B)$, the inner evaluation $\mathrm{eval}^I_{in}(a_{in}, \sigma_{a_{in}})$ evaluates $\sigma_{a_{in}}$ on $kd_l + 2$ points $p^{I,a_{in}}_{y_{a_{in}},1}, \ldots, p^{I,a_{in}}_{y_{a_{in}},k} \in \mathbb{L}^{w'}$ (corresponding to the k vertex queried points) and $q^{I,a_{in}}_{y_{a_{in}},1}, q^{I,a_{in}}_{y_{a_{in}},2} \in \mathbb{L}^{w'}$ (corresponding to the 2 path queried points) for some $y_{a_{in}} \in \mathbb{L}^\tau$ determined only by a_{in}. For each b_{out} with $(a_{out}, b_{out}) \in E_{out}$, there exist k points $p^{I,a_{in},b_{out}}_{y_{a_{in}},1}, \ldots, p^{I,a_{in},b_{out}}_{y_{a_{in}},k}$ corresponding the k vertex queried points associated to b_{out} in the former kd_l points. Then for each $e = (a, b) \in E$, where $a = \langle a_{out}, a_{in}\rangle$, $b = \langle b_{out}, b_{in}\rangle$, $\mathrm{label}(e) = \xi = (\xi_{out}, \xi_{in})$, and $I = (a_{out}, \xi_{out})$, and for $\sigma_a \in \Sigma_A$, the evaluation is defined as $\mathrm{eval}_e(\sigma_a) = (\sigma(\xi_{out})(p^{I,a_{in},b_{out}}_{y_{a_{in}},1}), \ldots, \sigma(\xi_{out})(p^{I,a_{in},b_{out}}_{y_{a_{in}},k}))$, i.e., the corresponding k values in $\mathrm{eval}^I_{in}(a_{in}, \sigma(\xi_{out}))$.

Finally, we specify the validity test. Remember that $\mathrm{valid}^I_{in} \equiv 1$, and $\mathrm{valid}_{out}(a_{out}, \sigma_{a_{out}})$ performs the consistency check among Ω_{out} polynomials $\{\sigma_{a_{out}}(\xi_{out})\}_{\xi_{out} \in \Omega_{out}}$ in $\Sigma_{B_{out}}$ according to the depth-2 evaluation tree $T_{a_{out}}$. For $a = \langle a_{out}, a_{in}\rangle \in A$ and $\sigma_a \in \Sigma_A$, the test $\mathrm{valid}(a, \sigma_a)$ performs the consistency check among Ω_{out} polynomials $\{\sigma_a(\xi_{out})\}_{\xi_{out} \in \Omega_{out}}$ in $\Sigma_{A_{in}}$ according to the depth-2 evaluation tree $T_{a_{out}}$, where the evaluation points are replaced with $q^{I,a_{in}}_{y_{a_{in}},1}, q^{I,a_{in}}_{y_{a_{in}},2}$ corresponding

[5]Here we omit the tuple function $\{\tau_e\}$ from Definition 4.6 since it is irrelevant for the balanced property.

path queried points on $I = (a_{out}, \xi_{out})$ indexed by $y_{a_{in}}$ for each $\xi_{out} \in \Omega_{out}$. As valid_{out}, the new test $\mathrm{valid}(a, \sigma_a)$ outputs 1 if all the evaluated values (in $\mathbb{L}$) are the same for each vertex in the tree.

Now we observe the balanced property. Remember that the evaluation tree $T_{a_{out}}$ has the same graph structure regardless of a_{out}. For each $e = (a, b) \in E$, where $a = \langle a_{out}, a_{in} \rangle$, $b = \langle b_{out}, b_{in} \rangle$, $\mathrm{label}(e) = (\xi_{out}, \xi_{in})$, $I = (a_{out}, \xi_{out})$, and for $\sigma_a \in \Sigma_A$, the evaluation function, the validity test, and the projection rule evaluate $\sigma_a(\xi_{out})$ on the following $k + 3$ points

$$p^{I,a_{in},b_{out}}_{y_{a_{in}},1}, \ldots, p^{I,a_{in},b_{out}}_{y_{a_{in}},k}, q^{I,a_{in}}_{y_{a_{in}},1}, q^{I,a_{in}}_{y_{a_{in}},2}, \xi_{in}.$$

Based on the same argument as that of Appendix A.4.1, it suffices to show that the number of linear functions $\sigma_a(\xi_{out})$ that takes $k+3$ values $v \in \mathbb{L}^{k+3}$ on the points above is the same regardless of the choice of v. Namely, it suffices to show that the $k + 3$ points above are linearly independent. Notice that $p^{I,a_{in},b_{out}}_{y_{a_{in}},1}, \ldots, p^{I,a_{in},b_{out}}_{y_{a_{in}},k}, q^{I,a_{in}}_{y_{a_{in}},1}, q^{I,a_{in}}_{y_{a_{in}},2}$ are evaluation points by $\mathrm{eval}^I_{in}(a_{in}, \text{-})$, and $\xi_{in} = \mathrm{label}^I_{in}((a_{in}, b_{in}))$. In Appendix A.2.3, we have observed that the linear independence holds when the corresponding elements in the given list for Had-LR are linearly independent. In Appendix A.3.4, we have shown that this occurs when the corresponding elements in $\mathbb{F}^w \times \mathbb{L}^t$ in the given list for RM$\diamond$Had-LR have distinct $\mathbb{F}^w$ elements. Notice that the $\mathbb{F}^w$-elements corresponding to $p^{I,a_{in},b_{out}}_{y_{a_{in}},1}, \ldots, p^{I,a_{in},b_{out}}_{y_{a_{in}},k}, q^{I,a_{in}}_{y_{a_{in}},1}, q^{I,a_{in}}_{y_{a_{in}},2}$ are k vertex queried points of I (for b_{out}) and 2 path queried points of I. Thus, their $\mathbb{F}^w$-elements are distinct (see the definitions of each queried point) and the balanced property holds for the resulting LDRC.

B An Approximation Algorithm for DNF-MMSA

We present an elementary approximation algorithm for DNF-MMSA based on LP relaxation.

Theorem B.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 $v_{IP} \leq s^*$.

We round α to a binary assignment $\tilde{\alpha} \in \{0,1\}^n$ as follows: for each $i \in [n]$,

$$\tilde{\alpha}_i = \begin{cases} 1 & \text{if } (\ell/(1-\eta)) \cdot \alpha_i \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Then, $w(\tilde{\alpha}) \leq (\ell/(1-\eta)) \cdot w(\alpha) \leq (\ell/(1-\eta))s^*$. 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 $\mathbb{E}_i[z_i] \geq 1-\epsilon$, 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 $y_{i,j} \geq (1-\eta)/\ell$ since $y_{i,j} \geq 0$. 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. □

C Advice Complexity of Sampling

We fix a prefix-free universal Turing machine U such that for every prefix-free Turing machine M, there exists a description d_M of M such that $U(d_M, x) = M(x)$ for every $x \in \{0,1\}^*$. Let $\mathrm{K}(x \mid y)$ denote the minimum length of $d \in \{0,1\}^*$ such that $U(d; y) = x$, where y is given to the prefix-free universal Turing machine on an auxiliary tape.

Definition C.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 $\alpha\colon \{0,1\}^{2^{cn}} \to \{0,1\}^\ell$ such that $\mathcal{D}$ is statistically identical to the distribution generated by $U(\alpha(r), r, n)$ for $r \sim \{0,1\}^{2^{cn}}$ conditioned on the event that $U(\alpha(r), r, n) \neq \bot$, and the probability that $U(\alpha(r), r, n) \neq \bot$ is at least $1 - 2^{-2^\ell}$, where $c := 100$.*

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. Note that we allow the sampler to output a special symbol "$\bot$" to indicate the failure of the sampling procedure.

We introduce two equivalent notions of advice complexity of sampling. One is a coding property:

Definition C.2. *We say that a distribution $\mathcal{D}$ over $\{0,1\}^n$ has a* coding property with error ℓ *if*

$$\mathrm{K}(x \mid n) \leq -\log \mathcal{D}(x) + \ell$$

for every $x \in \mathsf{Support}(\mathcal{D})$.

The other is the ∞-Rényi divergence to the universal distribution.

Definition C.3 (Rényi divergence)**.** *For two distributions P and Q over $\{0,1\}^n$ such that $\mathsf{Support}(P) \subseteq \mathsf{Support}(Q)$, the ∞-Rényi divergence between P and Q is defined to be*

$$\Delta_\infty(P \parallel Q) := \log \max \left\{ \frac{P(x)}{Q(x)} \,\middle|\, x \in \mathsf{Support}(Q) \right\}.$$

Definition C.4 (Universal distribution)**.** *The* universal distribution *over* $\{0,1\}^n$*, denoted by* m_n*, is defined to be the distribution such that* $\mathrm{m}_n(x) \propto 2^{-\mathrm{K}(x|n)}$ *for every* $x \in \{0,1\}^n$*. That is,* $\mathrm{m}_n(x) \coloneqq 2^{-\mathrm{K}(x|n)}/Z_n$*, where* $Z_n \coloneqq \sum_{y\in\{0,1\}^n} 2^{-\mathrm{K}(y|n)}$ *is a normalizing constant.*

Proposition C.5. *Let* $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ *be a family of distributions such that* $\mathsf{Support}(\mathcal{D}_n) \subseteq \{0,1\}^n$*. The following are equivalent for every function* $\ell\colon \mathbb{N} \to \mathbb{N}$ *with* $\omega(1) \le \ell(n) \le n$*.*

1. $\mathcal{D}_n$ *is samplable with advice complexity* $O(\ell(n))$ *for all large* $n \in \mathbb{N}$
2. $\Delta_\infty(\mathcal{D}_n \parallel \mathrm{m}_n) \le O(\ell(n))$ *for all large* $n \in \mathbb{N}$*.*
3. $\mathcal{D}_n$ *has a coding property with error* $O(\ell(n))$ *for all large* $n \in \mathbb{N}$*.*

Proof. We first observe the equivalence between Items 2 and 3. Item 2 is equivalent to

$$-\log \mathrm{m}_n(x) \le -\log \mathcal{D}_n(x) + O(\ell(n)) \tag{16}$$

for every $x \in \mathsf{Support}(\mathcal{D}_n)$. By definition, we have $-\log \mathrm{m}_n(x) = \mathrm{K}(x \mid n) + \log Z_n$ for the normalizing constant $Z_n = \sum_{y\in\{0,1\}^n} 2^{-\mathrm{K}(y|n)}$. Since $\mathrm{K}(y \mid n) \le n + O(1)$ for every $y \in \{0,1\}^n$, we have $Z_n \ge \Omega(1)$. We also have $Z_n \le 1$ by Kraft's inequality. It follows that $|-\log \mathrm{m}_n(x) - \mathrm{K}(x \mid n)| = |\log Z_n| = O(1)$. Thus, (16) is equivalent to Item 3, i.e.,

$$\mathrm{K}(x \mid n) \le -\log \mathcal{D}_n(x) + O(\ell(n)).$$

We prove Item 3 $\Rightarrow$ Item 1. The idea is to use rejection sampling. Assume that

$$\mathrm{K}(x \mid n) \le -\log \mathcal{D}_n(x) + \ell(n) \tag{17}$$

for every $x \in \mathsf{Support}(\mathcal{D}_n)$. Consider the following sampling procedure S that takes as input $n \in \mathbb{N}$, a coin flip sequence r, and an advice string $\alpha(r)$. It reads a parameter T (specified later) from the advice string $\alpha(r)$. Using the randomness r, it samples T strings $d^{(1)}, \cdots, d^{(T)} \sim \{0,1\}^{2n}$ and T real numbers $s^{(1)}, \cdots, s^{(T)} \sim [0,1]$ uniformly at random.[6] It reads an index $t \in [T]$ from the advice string $\alpha(r)$. If $t = 0$, then S outputs $\perp$ and halts. Otherwise, it "dovetails" the computations of the prefix-free universal Turing machine $U(n,y)$ for all the prefixes y of $d^{(t)}$ and lets x be its output. (Specifically, it runs $U(n,y)$ for all the prefixes y of $d^{(t)}$ in parallel. If U halts on some prefix y of $d^{(t)}$, then x is defined to be $U(n,y)$. By the prefix-free property, the output x is unique.) The output of S is defined to be x.

We now specify the advice string $\alpha(r) = (T,t)$. We define $T \coloneqq 2^{2\ell(n)}$. We define t to be the first index $t \in [T]$ such that $U(n,y)$ halts for some prefix y of $d^{(t)}$ and for $x \coloneqq U(n,y)$, it holds that $s^{(t)} \le \mathcal{D}_n(x)/\big(2^{\ell(n)} \cdot Q(x \mid n)\big)$, where we define

$$Q(x \mid n) \coloneqq \sum_{d\in\{0,1\}^{\le 2n}\colon U(n,d)=x} 2^{-|d|} \ge 2^{-\mathrm{K}(x|n)}.$$

If no such index exists, we define $t \coloneqq 0$. Note that, by Equation (17),

$$\frac{\mathcal{D}_n(x)}{2^{\ell(n)} \cdot Q(x \mid n)} \le \frac{\mathcal{D}_n(x)}{2^{\ell(n)-\mathrm{K}(x|n)}} \le 1.$$

The length of the advice string is at most $O(\log T + \log t) = O(\ell(n))$.

[6] Strictly speaking, it is impossible to choose a uniformly random real number. Instead, we select randomly from a binary decimal number with 2^n digits. This induces negligible errors, which we ignore for simplicity.

Let $x^{(t)}$ be the output of $U(n, y)$ for some prefix y of $d^{(t)}$ (which is uniquely defined because of the prefix-free property of U; if U does not halt on any prefix of $d^{(t)}$, let $x^{(t)}$ be undefined). Fix $x \in \{0,1\}^n$ and $t \in [T]$. The probability that $x^{(t)} = x$ is equal to $Q(x \mid n)$. Thus, the probability that $x^{(t)} = x$ and $s^{(t)} \leq \mathcal{D}_n(x)/\big(2^{\ell(n)} \cdot Q(x \mid n)\big)$ is equal to $\mathcal{D}_n(x)/2^{\ell(n)}$. For a fixed $t \in [T]$, the probability that $s^{(t)} \leq \mathcal{D}_n(x^{(t)})/\big(2^{\ell(n)} \cdot Q(x \mid n)\big)$ is

$$\sum_{x \in \{0,1\}^n} \frac{\mathcal{D}_n(x)}{2^{\ell(n)}} = 2^{-\ell(n)}.$$

Thus, conditioned on the event that $s^{(t)} \leq \mathcal{D}_n(x^{(t)})/\big(2^{\ell(n)} \cdot Q(x \mid n)\big)$, the probability that $x = x^{(t)}$ is equal to $\mathcal{D}_n(x)$; that is, the conditional distribution of $x^{(t)}$ is statistically identical to $\mathcal{D}_n$. Moreover, the probability that for every $t \in [T]$, the inequality $s^{(t)} \leq \mathcal{D}_n(x^{(t)})/\big(2^{\ell(n)} \cdot Q(x \mid n)\big)$ fails is at most $(1 - 2^{-\ell(n)})^T \leq \exp(-2^{-\ell(n)} \cdot T) = \exp(-2^{\ell(n)})$. Thus, the sampling procedure does not output $\perp$ with probability at least $1 - 2^{-2^{\ell(n)}} \geq 1 - \exp(-2^{O(\ell(n))})$.

We prove Item 1 $\Rightarrow$ Item 3. Assume that $\mathcal{D}_n$ is samplable with advice complexity $\ell(n)$. Consider the distribution generated by $U(w, r, n)$ for uniformly random $w \sim \{0,1\}^{\ell(n)}$ and $r \sim \{0,1\}^{2^{O(n)}}$. Fix $x \in \{0,1\}^n$. Then, we have

$$\Pr[U(w, r, n) = x] \geq 2^{-\ell(n)} \cdot \Pr[U(\alpha(r), r, n) = x]$$

because we have $w = \alpha(r)$ with probability at least $2^{-\ell(n)}$. We also have

$$\Pr[U(\alpha(r), r, n) = x] \geq \Pr[U(\alpha(r), r, n) \neq \perp \text{ and } U(\alpha(r), r, n) = x] \geq \frac{1}{2} \cdot \mathcal{D}_n(x).$$

The distribution $U(w, r, n)$ is an enumerable semi-probability distribution given n. It follows from the coding theorem (see, e.g., [Lee06; LV19]) that

$$\mathrm{K}(x \mid n) \leq -\log \Pr[U(w, r, n) = x] \leq -\log \mathcal{D}_n(x) + O(\ell(n)).$$

□

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il