Electronic Colloquium on Computational Complexity, Report No. 106 (2021)

The Space Complexity of Sampling

Eshan Chattopadhyay*
Cornell University
eshanc@cornell.edu

Jesse Goodman*
Cornell University
jpmgoodman@cs.cornell.edu

David Zuckerman†
University of Texas at Austin
diz@cs.utexas.edu

July 22, 2021

Abstract

Recently, there has been exciting progress in understanding the complexity of distributions. Here, the goal is to quantify the resources required to generate (or sample) a distribution. Proving lower bounds in this new setting is more challenging than in the classical setting, and has yielded interesting new techniques and surprising applications. In this work, we initiate a study of the complexity of sampling with *limited memory*, and prove small-space analogs of several results previously known only for sampling in AC^0.

1. We exhibit an explicit boolean function $b : \{0,1\}^n \to \{0,1\}$ that cannot be computed by width $2^{\Omega(n)}$ read-once branching programs (ROBPs), even on average, but such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be exactly sampled by ROBPs of width $O(n)$.
2. We exhibit an explicit boolean function $b : \{0,1\}^n \to \{0,1\}$ such that any distribution sampled by an ROBP of width $2^{\Omega(n)}$ has statistical distance $\frac{1}{2} - 2^{-\Omega(n)}$ from $(\mathbf{U}_n, b(\mathbf{U}_n))$. We show that any such b witnesses exponentially small correlation bounds against ROBPs, and we extend these results to hold for the unknown-order setting.
3. We show that any distribution sampled by an ROBP of width $2^{\Omega(n)}$ has statistical distance $1 - 2^{-\Omega(n)}$ from any distribution that is uniform over a good code. More generally, we obtain sampling lower bounds for any list decodable code, which are nearly tight. Using a known connection, we also obtain data structure lower bounds for storing codewords.

Along the way, we prove a direct product theorem and several equivalence theorems. These tools offer a generic method to construct distributions with strong sampling lower bounds, and translate these lower bounds into correlation bounds against ROBPs. As an application of our direct product theorem, we show a strong complexity separation between sampling with AC^0 circuits and sampling with ROBPs.

*Supported by NSF CAREER Award 2045576.
†Supported by NSF Grants CCF-1705028 and CCF-2008076 and a Simons Investigator Award (#409864).

ISSN 1433-8092

1 Introduction

A central goal in complexity theory is to quantify the resources required to perform certain tasks. Traditionally, complexity theory has focused on the task of *computing*: here, one fixes a function $f : \{0,1\}^m \to \{0,1\}^n$ and computational model $\mathcal{C}$ (e.g., low-depth circuits), and asks for lower bounds on the size of any $F \in \mathcal{C}$ that computes f.

Recently, a growing body of work has sought to understand the power of these same computational models for the task of *sampling*. Here, instead of fixing a function f, one picks a target distribution $\mathbf{Q} \sim \{0,1\}^n$. Then, one asks for lower bounds on the size of any $F \in \mathcal{C}$ that generates (samples) $\mathbf{Q}$, when supplied with uniformly random bits.

The complexity of sampling can be traced back to the '80s [JVV86], and has recently seen an exciting new wave of interest [ASTS+03, GGN10, Vio12a, Vio12b, LV12, BIL12, DW12, JSWZ13, Aar14, Wat14, Vio14, BCS16, Wat16, Vio16, Wat20, Vio20, GW20]. Despite this significant progress, results are still only known for a few computational models like AC^0 and communication protocols. In particular, nothing is known about the complexity of sampling with *limited memory*, while this remains a fundamental model in other areas of complexity theory.

In this work, we aim to fill this gap, and initiate a study of the *space* complexity of sampling. Our model corresponds to the streaming model of computation, an active area of research. Before we dive into our model and present our results, we briefly survey sampling in AC^0, and motivate some key questions about sampling with limited memory.

Sampling can be easier than computing A motivating paradigm in the complexity of sampling is the (perhaps surprising) fact that a fixed computational model $\mathcal{C}$ may be more powerful at sampling than computing. In particular, consider fixing a function $f : \{0,1\}^n \to \{0,1\}^m$ and comparing the tasks of computing f on every input x, with sampling $f(\mathbf{U}_n)$ [GGN10]. Intuitively, the latter task should seem easier: any $F \in \mathcal{C}$ that computes f must also have $F(\mathbf{U}_n) = f(\mathbf{U}_n)$. Furthermore, setting f^{-1} to be a one-way permutation makes $f(x)$ very hard to compute, but $f(\mathbf{U}_n)$ very easy to sample [Vio12a].

Amazingly, we also have examples of extremely simple *explicit* functions that demonstrate this separation. The canonical example is the function $f(x) = (x, \mathsf{parity}(x))$: the celebrated result of Håstad [Hås87] shows that f cannot be computed in AC^0, yet Babai [Bab87], Boppana and Lagarias [Kil88] give an extremely simple AC^0 circuit that samples $(\mathbf{U}_n, \mathsf{parity}(\mathbf{U}_n))$. Thus, obtaining sampling lower bounds is strictly more challenging (at least in AC^0), and their pursuit may unveil exciting new techniques and applications [Vio12a].

A natural first question, then, is to ask whether this motivation still holds in the limited memory setting:

Question 1. *Does there exist an explicit boolean function $b : \{0,1\}^n \to \{0,1\}$ such that $(x, b(x))$ is hard to compute with limited memory, but $(\mathbf{U}_n, b(\mathbf{U}_n))$ is easy to sample with limited memory?*

Sampling lower bounds for input-output pairs Given the above example of a boolean function $b : \{0,1\}^n \to \{0,1\}$ whose input-output pair $(x, b(x))$ is hard to compute but easy to sample in AC^0, perhaps the most natural first task is to exhibit a distribution of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$ that is *hard to sample* in AC^0. Recently, Viola proved a very strong result of this form [Vio20], giving an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any AC^0 circuit $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$, it holds that $|F(\mathbf{U}_\ell) - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{-n^{\Omega(1)}}$, where $|\cdot|$ denotes statistical distance.

This motivates us to ask the following question:

Question 2. *Does there exist an explicit boolean function $b : \{0,1\}^n \to \{0,1\}$ such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ is hard to sample with limited memory?*

Sampling lower bounds for codes Finally, it is easy to see sampling lower bounds for distributions of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$ cannot exceed $1/2$ (since $(\mathbf{U}_n, 0)$ or $(\mathbf{U}_n, 1)$ will yield an upper bound of $1/2$, and both of these are trivial to sample). Thus, it is natural to wonder whether one can find any explicit distribution $\mathbf{Q} \sim \{0,1\}^n$ with much stronger sampling lower bounds, perhaps even approaching 1.

In 2012, Viola and Lovett demonstrated a distribution of exactly this type [LV12], setting $\mathbf{Q} \sim \{0,1\}^n$ to be uniform over an asymptotically good error-correcting code, i.e., one having constant relative distance and rate. They showed that for such a distribution $\mathbf{Q}$, any AC^0 circuit $F : \{0,1\}^\ell \to \{0,1\}^n$ has $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - \varepsilon$ for $\varepsilon = n^{-\Omega(1)}$. In a subsequent work [BIL12], Beck, Impagliazzo, and Lovett improved the statistical distance to $1 - \varepsilon$ for $\varepsilon = 2^{-n^{\Omega(1)}}$. Using an observation of Viola [Vio12a], both works also obtain interesting data structure lower bounds for storing codewords.

Given these strong sampling lower bounds against codes for AC^0, we would like to know:

Question 3. *Are good codes hard to sample with limited memory?*

In this work, we provide positive answers to the above three questions, and give a complexity separation between sampling in AC^0 and sampling with limited memory. Before presenting our results, we must discuss our model for sampling distributions with limited memory.

1.1 Sampling in small space using ROBPs

To model sampling with limited memory, we will use the classic model of *read-once branching programs* (ROBPs). The ROBP model corresponds to the streaming model of computation, and thus a better understanding of the power of ROBPs for sampling tasks may also help provide new insights and tools for streaming algorithms.

A first attempt to model sampling in limited memory uses the classic definition of an ROBP (Definition 7), and replaces its input with uniform bits. However, such an ROBP computes a function $f : \{0,1\}^\ell \to \{0,1\}$, and so the distribution $f(\mathbf{U}_\ell)$ it samples will be over $\{0,1\}$. To sample general distributions $\mathbf{Q} \sim \{0,1\}^n$, we need an ROBP that can output multiple bits.

Perhaps the most natural way to extend an ROBP to output multiple bits is to simply allow it to output a sequence of bits upon reading any input bit. More formally, we can assign each edge in the ROBP an additional label consisting of a string of output bits. Then, given an input $x \in \{0,1\}^\ell$, the ROBP traverses a path in the usual way, but now outputs all the output labels seen along the way. Indeed, this is exactly a "read-once" version of multi-output branching programs considered in previous works [BFK+79, BC82, Bea89].

It will also be convenient to make one simplifying assumption: just as the inputs in an ROBP are "layered", we will assume that the outputs are also layered. That is, we require that any two edges traversing between the same two layers are labeled with the same number of output bits. This is a natural way to guarantee that the ROBP will compute a function of the form $F : \{0,1\}^\ell \to \{0,1\}^n$, since all paths are guaranteed to output the same number of bits. This completes our definition of *multi-output ROBP* (see Definition 8 for a more formal definition).

Just as a standard ROBP models an algorithm that reads from an input stream, multi-output ROBPs also allow the algorithm to *write* to an output stream (since it may write an arbitrary number of bits at each time step, without storing any of them in its memory). As it turns out, we will also prove that sampling using this model is equivalent (up to a small loss in parameters) to sampling using a different natural model from the

field of randomness extractors [KRVZ11]. Furthermore, note that for functions with one bit of output, our definition is equivalent to the classic single-bit-output ROBP definition. Thus, we will henceforth refer to multi-output ROBPs simply as ROBPs.

1.2 Summary of our results

We are now ready to formally state our results. At a high level, we prove the following four results for sampling with ROBPs:

1. Sampling is easier than computing.
2. Sampling lower bounds against input-output pairs.
3. Sampling lower bounds against codes.
4. A direct product theorem, which yields a strong complexity separation between sampling with AC^0 and sampling with ROBPs.

Before we begin, we remind the reader that we are interested in the sampling power of ROBPs when they have unlimited access to randomness, but have a bounded width w (as this parameter corresponds to their memory). Thus, whenever we specify an ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ below, we emphasize that there is no restriction on ℓ. We note that ℓ may even be exponentially large (or bigger still!) for our lower bounds, but for our upper bounds we can always take $\ell \leq \mathrm{poly}(n)$.

1.2.1 Sampling is easier than computing for ROBPs

In our first main theorem, we answer Question 1, and show that ROBPs are more powerful at sampling than computing. In particular, we give a function $b : \{0,1\}^n \to \{0,1\}$ such that ROBPs cannot compute $(x, b(x))$ (even with exponential width), but *can* sample $(\mathbf{U}_n, b(\mathbf{U}_n))$ (using just linear width). In fact, for the function we give, ROBPs will actually fail to compute the correct answer on $\approx 1/4$ fraction of the input.

Theorem 1 (Sampling is easier than computing)**.** *For any fixed $\varepsilon > 0$, there exist constants $C, c > 0$ and an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that the following holds. For every ROBP $F : \{0,1\}^n \to \{0,1\}$ of width $w \leq 2^{cn}$,*

$$\Pr_x[F(x) \neq b(x)] > \frac{1}{4} - \varepsilon,$$

but there exists an ROBP $G : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $w \leq Cn$ such that

$$G(\mathbf{U}_\ell) = (\mathbf{U}_n, b(\mathbf{U}_n)).$$

Thus, just like with AC^0, we see that sampling with ROBPs is strictly easier than computing with ROBPs. As a result, sampling lower bounds for ROBPs will be more challenging to obtain than classical lower bounds for ROBPs.

1.2.2 Sampling lower bounds against input-output pairs for ROBPs

Given the above result, it is natural to ask whether we can obtain sampling lower bounds for distributions of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$. We obtain such bounds in our second main theorem, answering Question 2.

Theorem 2 (Sampling lower bounds against input-output pairs)**.** *There is a universal constant $c > 0$ and an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $w \leq 2^{cn}$,*

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{-cn}.$$

We remark that Theorem 2 is tight up to the constant c, since statistical distance $\leq \frac{1}{2}$ can be achieved by an ROBP of width 1, and statistical distance $\leq \frac{1}{2} - 2^{-n}$ can be achieved by an ROBP of width 2. It turns out that we can take b as the $\mathbb{F}_2$-inner product function IP. Surprisingly, $(\mathbf{U}_n, \mathsf{IP}(\mathbf{U}_n))$ can be sampled in AC^0 [IN96], which gives us the following complexity separation.

Corollary 1. *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that the distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be sampled by AC^0 circuits, but cannot be sampled by ROBPs of width $2^{\Omega(n)}$.*

Furthermore, we provide an equivalence theorem which shows that sampling lower bounds against distributions of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$ immediately yield correlation bounds against b for ROBPs. By instantiating this equivalence theorem with Theorem 2, we obtain as a corollary an alternate proof of the known result that the $\mathbb{F}_2$*-inner product function* has exponentially small correlation with ROBPs of exponential width.[1]

We also show that Theorem 2 holds for ROBPs that may write their output in any order. By extending our equivalence theorem to hold for the unknown-order setting, we obtain as a corollary an alternate proof of the known result that good *extractors for interleaved sources* have exponentially small correlation with unknown-order ROBPs of exponential width.[2] Furthermore, these equivalence theorems provide a new route towards proving correlation bounds against ROBPs.

1.2.3 Sampling lower bounds against codes for ROBPs

In our third main theorem, we show that codes are very hard to sample in limited memory, even for ROBPs of exponential width, answering Question 3. Our most general result is the following, which holds for any list decodable code.

Theorem 3 (Sampling lower bounds against codes)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over a (ρ, L) list decodable code of dimension k. Then for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 8wL \cdot 2^{-\frac{\rho}{1+2\rho}k}.$$

We now point out a few key parameter settings for this general theorem. Using a standard fact relating list decodable codes to (n, k, d) codes (Fact 2), our general bound yields the following result for sampling (n, k, d) codes.

Corollary 2. *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over an (n, k, d) code of dimension k. Then for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 12w \cdot 2^{-\frac{kd}{4n}}.$$

[1] The known proof of this result follows directly from average-case communication complexity lower bounds against the inner product function.

[2] The known proof of this result follows directly from the explicit map $f : \{0,1\}^n \to \{0,1\}$ of Raz and Yehudayoff [RY11] that has exponentially small discrepancy for every partition of $[n]$ into two equally sized sets.

We prove that Corollary 2 is almost tight, in the sense that for almost all "valid" parameters n, k, d, there exists an (n, k, d) code that can be sampled by an ROBP of width $2^{\widetilde{O}(\frac{kd}{n})}$ (see Theorem 24 and Remark 3). Furthermore, the above corollary implies that any distribution sampled by an ROBP of exponential width has statistical distance exponentially close to 1 from a good code, answering Question 3:

Corollary 3. *If $\mathbf{Q}$ is a good code, then for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width $2^{\Omega(n)}$,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 2^{-\Omega(n)}.$$

This is tight up to the constant hidden in the $\Omega(\cdot)$, since statistical distance $\leq 1 - 2^{-n}$ is easily achieved by a width 1 ROBP that is constant over a single codeword.

We now turn to some applications of this result. Using one direction of one of our equivalence theorems (Theorem 11), we show that ROBPs of exponential width cannot test membership of a good code. In fact, we prove something stronger, and show the following covariance bounds (see Definition 4 for a definition of covariance).

Corollary 4. *Let $b : \{0,1\}^n \to \{0,1\}$ be the indicator function of a good code $Q \subseteq \{0,1\}^n$. Then for any ROBP $F : \{0,1\}^n \to \{0,1\}$ of width $2^{\Omega(n)}$, it holds that $|\operatorname{cov}(F, b)| \leq 2^{-\Omega(n)}$.*

Next, by applying a known connection between sampling lower bounds and data structure lower bounds [Vio12a], we immediately get tight data structure lower bounds for storing codewords succinctly and retrieving them using ROBPs.

Corollary 5. *Let $Q \subseteq \{0,1\}^n$ be a good code of dimension k. Suppose we can store codewords of Q using $k + r$ bits so that a codeword can be computed by an ROBP $F : \{0,1\}^{k+r} \to \{0,1\}^n$ of width $2^{\Omega(n)}$. Then $r \geq \Omega(n)$.*

In particular, the above corollary shows that if one wishes to store codewords that are retrievable by a width $2^{\Omega(n)}$ ROBP, they must use $\Omega(n)$ extra bits of redundancy. This is tight up to constant factors: (1) It is easy to store codewords that are retrievable by a width 2^n ROBP using 0 extra bits of redundancy; and (2) It is easy to store codewords that are retrievable by a width 1 ROBP using $n - k$ extra bits of redundancy.

1.2.4 A direct product theorem and complexity separation between sampling with AC^0 and ROBPs

For our final main result, we prove a direct product theorem. This gives a generic way to construct distributions with strong sampling lower bounds against ROBPs. Informally, this theorem shows that if a distribution $\mathbf{Q}$ is even a little hard to sample for ROBPs, then the distribution $\mathbf{Q}^{\otimes t}$ (defined as a sequence of t independent copies of $\mathbf{Q}$) is extremely hard to sample for ROBPs. More formally, we prove the following.

Theorem 4 (Direct product theorem)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be a distribution such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w, it holds that $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq \delta$. Then for any $t \in \mathbb{N}$ and ROBP $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^{nt}$ of width w, it holds that*

$$|F^*(\mathbf{U}_{\ell^*}) - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}.$$

In particular, our direct product theorem gives a way to boost statistical distance lower bounds of the form $\delta > 0$ (some tiny constant) to lower bounds of the form $1 - 2^{-\Omega(t)}$.

As the main application of our direct product theorem, we provide a strong complexity separation between sampling with ROBPs and sampling with AC^0 circuits. In particular, we give a simple family of

distributions $\mathbf{Q} \sim \{0,1\}^n$ that can be sampled in AC^0, but such that any distribution generated by a width $2^{\Omega(\sqrt{n})}$ ROBP has statistical distance $1 - 2^{-\Omega(\sqrt{n})}$ from $\mathbf{Q}$. More generally, we achieve the following tradeoff.

Corollary 6. *There is a universal constant $c > 0$ such that for all sufficiently large $n \in \mathbb{N}$, the following holds. There exists an AC^0 circuit $Q : \{0,1\}^\ell \to \{0,1\}^n$ such that for every ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width $w \leq 2^{c \cdot n / \log(1/\varepsilon)}$,*

$$|F(\mathbf{U}_\ell) - Q(\mathbf{U}_\ell)| \geq 1 - \varepsilon.$$

We remark that ε can be set to any function of n, with some interesting settings being $\varepsilon = 0.01$ and $\varepsilon = 2^{-\sqrt{n}}$. Furthermore, while this separation demonstrates a distribution $\mathbf{Q}$ samplable with AC^0 circuits but not samplable by ROBPs, it of course does not imply that AC^0 circuits are strictly more powerful at sampling than ROBPs. Indeed, an interesting future direction is to find a family of distributions samplable by ROBPs, but not samplable by AC^0 circuits. One idea is to construct an extractor for AC^0 sources (distributions generated by AC^0 circuits), which can be computed by a small width ROBP.

This concludes the overview of our main results.

Organization The rest of this paper will be structured as follows. We start by giving an overview of our techniques in Section 2. In Section 3, we provide some basic preliminaries that will be used throughout the paper. Then, in Section 4, we prove a variety of equivalence theorems, which help set up all our proofs, and which allow us to convert our sampling lower bounds into correlation bounds.

In Section 5, we show that sampling is easier than computing for ROBPs, proving Theorem 1. Next, in Section 6, we provide our sampling lower bounds against input-output pairs, thereby proving Theorem 2. We prove our sampling lower bounds against codes (Theorem 3) in Section 7, and in Section 8 we provide our direct product theorem (Theorem 4). Finally, we wrap up with some future directions in Section 9.

2 Overview of our techniques

In this section, we give a detailed overview of the techniques that go into proving our four main theorems:

1. Sampling is easier than computing: Theorem 1.
2. Sampling lower bounds against input-output pairs: Theorem 2.
3. Sampling lower bounds against codes: Theorem 3.
4. A direct product theorem: Theorem 4.

Before we sketch any of these proofs, we provide a common set up that they will all use.

2.1 Switching worlds

When all is said and done, our goal is to obtain theorems about sampling with ROBPs: that is, we wish to gain a deeper understanding of distributions of the form $F(\mathbf{U}_\ell)$, where $F : \{0,1\}^\ell \to \{0,1\}^n$ is a function computed by an ROBP. However, given the generality of ROBPs, this model of sampling can be a little cumbersome to work with formally.

In order to circumvent the need to work directly with this model, *we will actually prove all of our results using two intermediate models*, which we call complex samplers and simple samplers. These two types

of samplers are much easier to work with formally, as their definitions are simpler. Surprisingly, it turns out that proving results over these intermediate models is also *stronger* than proving results about sampling with ROBPs. Indeed, we provide a small collection of equivalence theorems that will allow us to convert our results about simple and complex samplers into results about sampling and computing with ROBPs.

We go into more detail below.

Complex samplers and their equivalence to sampling with ROBPs We call our first intermediate model a *complex sampler*. Such a model was originally introduced by Kamp, Rao, Vadhan, and Zuckerman under the name of *small-space sources* [KRVZ11]. It is defined as a certain type of branching program that receives no input. More formally, a complex sampler of width w and length n is a directed acyclic graph $G = (V, E)$ with layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$, each holding w vertices. For every $i \in [n]$, each $v \in V_{i-1}$ can have an *arbitrary* number of edges into the next layer. The vertex v assigns a probability distribution p_v over its outgoing edges, and each of them also receive a label of 0 or 1. There is a distinguished start vertex $v_{\mathsf{start}} \in V_0$, and the complex sampler generates a distribution $\mathbf{X} \sim \{0,1\}^n$ by taking a random walk from v_{start} according to the edge probabilities $\{p_v\}$, outputting all bits seen along the way.

Here, the main equivalence theorem we will prove is that a distribution $\mathbf{Q} \sim \{0,1\}^n$ is samplable by a complex sampler of width w if and only if it is samplable by an ROBP of width w (ignoring a small loss in parameters: see Theorem 6). The more challenging direction of this proof is that complex samplers $\implies$ ROBP samplers. Here, the difficulty is with simulating the multiple outgoing edges from each vertex, and the arbitrary probabilities it may assign over them. However, we show that such a simulation is possible by combining a lemma of [KRVZ11] (to make the edge probabilities of the complex sampler "granular") with a construction of a certain type of ROBP that sorts boolean strings into buckets of various sizes. In other words, the ROBP computes a type of "multi-thresholding" function.

Given this equivalence theorem: (1) Lower bounds against complex samplers imply lower bounds against sampling with ROBPs; and (2) The existence of complex samplers for a distribution implies that such a distribution can be sampled with ROBPs. Thus, it allows us to focus on proving results about complex samplers, instead of about sampling with ROBPs.

Simple samplers and their equivalence to computing with ROBPs We call the second intermediate model we use a *simple sampler*. Such a model is identical to the complex sampler, except it has the additional restriction that *each vertex may only have two outgoing edges*, labeled 0 and 1, respectively.

Previously, we discussed that complex *samplers* are equivalent to *sampling* with ROBPs. It may therefore be surprising to learn that, in some sense, simple *samplers* are equivalent to *computing* with ROBPs. In more detail, we prove an equivalence theorem that says the following: for any $b : \{0,1\}^n \to \{0,1\}$, there exists a simple sampler of width w that generates $(\mathbf{U}_n, b(\mathbf{U}_n))$ if and only if there exists an ROBP of width w that computes b. The proof is not difficult, and we refer the reader to Section 4.2 for more detail.

The main way we will use this theorem is in the direction of ROBP for b $\implies$ simple sampler for $(\mathbf{U}_n, b(\mathbf{U}_n))$. This will allow us to convert simple sampler lower bounds against $(\mathbf{U}_n, b(\mathbf{U}_n))$ into hardness of computing for ROBPs. Moreover, we will actually give an average-case version of this direction (Theorem 8), which will allow us to obtain average-case hardness against ROBPs. The proof is again not difficult, and we refer the reader to Section 4.2.1 for more detail.

Finally, we prove a few more small equivalence theorems that we use to get some bonus results that we listed in the introduction. But we have already collected everything we need to give a sketch of our main results, so we omit them from this overview (see Section 4 for our full set of equivalences).

At last, we are ready to start sketching our main theorems. With our equivalence theorems in hand, we will see that we can just focus on proving results about simple and complex samplers.

2.2 Sampling is easier than computing

We now begin the sketch of our first main theorem, Theorem 1. Recall that it roughly says that there is an explicit $b : \{0,1\}^n \to \{0,1\}$ such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be sampled by width $w = O(n)$ ROBPs, but b cannot be computed even by ROBPs of width $w = 2^{\Omega(n)}$. In fact, it actually says that such ROBPs fail to compute b on $\approx 1/4$ fraction of possible inputs.

Using the equivalence theorems sketched in Section 2.1, we can convert this theorem into a simple statement about simple and complex samplers, and prove that instead. The new version of this theorem (Theorem 13) says that there is an explicit $b : \{0,1\}^n \to \{0,1\}$ such that $(\mathbf{U}_n, b(\mathbf{U}_n)$ can be sampled by width $w = O(n)$ complex samplers, but every simple sampler of width $w = 2^{\Omega(n)}$ generates a distribution $\mathbf{X}$ with statistical distance $\approx 1/4$ from $(\mathbf{U}_n, b(\mathbf{U}_n))$. We sketch its proof below.

A warm-up using the address function Our first candidate function for b is the address function $\mathsf{address} : \{0,1\}^n \to \{0,1\}$, where $n = k + \log k$ for some k. This function parses its input $y \in \{0,1\}^n$ as $y = (x, \alpha)$, where $x \in \{0,1\}^k$ and $\alpha \in [k]$. Then, it outputs x_α. It is well-known that this function separates read-once and read-twice branching programs (see, e.g., [GV20]). We will see that it also separates simple and complex samplers, but that it will fail to give much more than a worst-case separation.

In more detail, we would like to show that width $w = O(n)$ complex samplers can generate $(\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n))$, but that width $w = 2^{\Omega(n)}$ simple samplers cannot. We sketch these one at a time.

First, to see why simple samplers cannot generate this distribution, consider any simple sampler $\mathcal{B}$, and suppose it has width $w = 2^k - 1$. Observe that in a simple sampler, there is a *single unique path* corresponding to any sequence of output bits it might produce. Thus, if we look at layer V_k in this simple sampler, there must be two *different* sequences of bits $x, y \in \{0,1\}^k$ that lead to the same vertex in $v \in V_k$. This is a big problem, because x and y must differ at some coordinate $\alpha \in [k]$, but the simple sampler cannot remember which of these strings it just outputted upon arriving at v. Thus, no matter what bit it decides to output at the very end, it will either disagree with x_α or y_α. In other words, the simple sampler fails to exactly generate $(\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n))$, and it has width $w = 2^k - 1 = 2^{\Omega(n)}$.

Now, to see why complex samplers can generate this distribution using width $w = O(n)$, we start by observing that $(\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n))$ is a convex combination of $2k$ distributions, each with a very nice structure. In particular, let $\mathbf{X}$ be uniform over $\{0,1\}^k$, let $\mathbf{A}$ be uniform over $[k]$, and note that $(\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n)) = (\mathbf{X}, \mathbf{A}, \mathbf{X}_\mathbf{A})$. Thus, for any fixing of $\mathbf{A}$ to a, and any fixing of $\mathbf{X}_a$ to $x_a \in \{0,1\}$, the distribution $(\mathbf{X}, \mathbf{A}, \mathbf{X}_\mathbf{A})$ simply becomes a sequence of independent bits: some of which are uniform, and some of which are constant. Call a distribution with this structure "nice." It is easy to verify that there are $2k$ such fixings, and each happens with probability $\frac{1}{2k}$.

Thus we see that $(\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n))$ is a convex combination of $2k$ nice distributions. Furthermore, it is easy to verify that any nice distribution can be sampled by a complex sampler of width $w = 1$. The final step now is the following observation: for any collection $\{\mathbf{Y}_i\}_{i \in [2k]}$ of distributions that can each be sampled in width 1 by a complex sampler, any convex combination of distributions from this set can be sampled in width $2k$ by a complex sampler.

To see why the above is true, start by drawing a new complex sampler $\mathcal{B}_i$ for each $\mathbf{Y}_i$. Then, draw a new node v^*, and connect it to the start node of each $\mathcal{B}_i$. Have v^* assign a probability distribution over its

outgoing edges, which corresponds to the convex combination desired. This new sampler indeed has width at most $2k$, and a random walk starting at v^* will indeed sample the desired distribution. But technically, our definition required each edge leaving v^* to be labeled with an output bit. But this is easy to fix: for each $\mathcal{B}_i$, let v^i_{start} be its start vertex. Suppose the edge from v^* to v^i_{start} is labeled with probability p_i. Then simply (1) remove the edge between $v^*, v^i_{\mathsf{start}}$, and (2) copy the edges leaving v^i_{start} onto v^*, multiplying their assigned probabilities by p_i.

Thus, we get a complex sampler of width $2k = O(n)$ that exactly samples $(\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n))$.

Obtaining average-case bounds through list-decodable codes So far, we've shown that width $O(n)$ complex samplers can generate $(\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n))$, but width $2^{\Omega(n)}$ simple samplers cannot. What we would really like is to show that such simple samplers cannot even come close to generating this distribution.

Going back to the impossibility proof for the simple sampler, recall that we argued that two different outputs $x, y \in \{0,1\}^k$ will lead to the same state in layer V_k. The simple sampler thus "forgot" what it output before that point, and since the strings x, y differ at some coordinate $\alpha \in [k]$, the simple sampler is guaranteed to make a mistake at least some of the time when outputting its last bit.

But if we want to show that the output $\mathbf{B}$ of the simple sampler is *far* from $(\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n))$, this isn't good enough: it could be the case that x, y differ at just *one* coordinate α, which shouldn't be getting selected as output with probability $\gg 1/k$. Thus, arguing that $|\mathbf{B} - (\mathbf{U}_n, \mathsf{address}(\mathbf{U}_n))| \geq \Omega(1)$ using this method does not seem promising.

The natural idea is to *force* x, y to differ at $\gg 1$ coordinate, and perhaps even $\Omega(k)$ coordinates. Towards this end, we define a more elaborate version of the address function, $\mathsf{address}^* : \{0,1\}^n \to \{0,1\}$, as follows (where n is just a little bigger than $k + \log k$). First, it parses its input $y \in \{0,1\}^n$ as $y = (x, \alpha)$ for $x \in \{0,1\}^k$ and $\alpha \in [\hat{k}]$, where $\hat{k}$ is a little bigger than k. Then, it encodes $x \in \{0,1\}^k$ into a codeword $\hat{x} \in \{0,1\}^{\hat{k}}$ using a good (ρ, L) list decodable code.[3] Finally, it outputs $\hat{x}_\alpha$.

The idea now is as follows. Consider any vertex v in layer V_k of the simple sampler. Let $x \in \{0,1\}^k$ be the simple sampler's "favorite" sequence of output bits leading to v, in the sense that it outputs $(x, \alpha, \hat{x}_\alpha)$ with higher probability than $(x, \alpha, \neg\hat{x}_\alpha)$ for the most number of indices α. Then by definition of list decodability, all but L of the other output strings y leading to v must have $\hat{y}$ differ from $\hat{x}$ at many coordinates ($\rho\hat{k}$). Then either one of two things could happen: (1) the simple sampler actually incorrectly outputs $(x, \alpha, \neg\hat{x}_\alpha)$ quite often for many α, meaning that it does the same for every other string leading to v (since x is the favorite); or (2) the simple sampler correctly outputs $(x, \alpha, \hat{x}_\alpha)$ for many α, including those where $\hat{x}, \hat{y}$ differ (meaning that it incorrectly outputs $(y, \alpha, \neg\hat{y}_\alpha)$ at such coordinates).

In the end, for all but wL of the strings $x \in \{0,1\}^k$, there are many indices $\alpha \in [\hat{k}]$ for which the simple sampler is likely to incorrectly output $\neg\hat{x}_\alpha$ as its final bit. This ultimately yields sampling lower bounds of roughly $\approx \frac{1}{2} \cdot \rho$ for exponential width simple samplers (Lemma 6), and plugging in a great list decodable code yields $\approx \frac{1}{4}$, as desired.

Thus, we have shown that any *simple sampler* $\mathbf{B}$ has roughly $|\mathbf{B} - (\mathbf{U}_n, \mathsf{address}^*(\mathbf{U}_n))| \geq 1/4$. But now we need to make sure that any *complex sampler* can still exactly sample $(\mathbf{U}_n, \mathsf{address}^*(\mathbf{U}_n))$. As with the original address function, we start by letting $\mathbf{X}$ be uniform over $\{0,1\}^k$, letting $\mathbf{A}$ be uniform over $[\hat{k}]$, and noting that $(\mathbf{U}_n, \mathsf{address}^*(\mathbf{U}_n)) = (\mathbf{X}, \mathbf{A}, \hat{\mathbf{X}}_{\mathbf{A}})$. Similar to before, we'd like to show that this is a convex combination of a few "nice" distributions. Towards this end, we fix $\mathbf{A}$ to $a \in [\hat{k}]$, and fix $\hat{\mathbf{X}}_a$ to $\hat{x}_a \in \{0,1\}$. This yields a convex combination of $\leq 2\hat{k}$ distributions.

[3] Recall that a set $Q \subseteq \{0,1\}^{\hat{k}}$ is a (ρ, L) list decodable code if every point $x \in \{0,1\}^{\hat{k}}$ has at most L elements from Q within distance $\rho\hat{k}$ from it.

But notice now that the random variable $\mathbf{X}$ under this conditioning is no longer guaranteed to have the same nice form as with the simpler address function, since we didn't fix one of its coordinates, but instead fixed one of the coordinates of $\hat{\mathbf{X}}$. However, if we originally picked a *linear* list decodable code in our construction of $\mathsf{address}^*$, then $\mathbf{X}$ will be uniform over an affine space of codimension ≤ 1, which we show is not hard to sample in width 2.

Finally, using the same convex combination argument from our sketch of the original address function, we get that there is a complex sampler of width $w = 2 \cdot 2\hat{k} = O(n)$ for $(\mathbf{U}_n, \mathsf{address}^*(\mathbf{U}_n))$.

In conclusion, we get that for the function $b : \{0,1\}^n \to \{0,1\}$ defined as $\mathsf{address}^*$, it holds that $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be sampled by width $w = O(n)$ complex samplers, but every simple sampler of width $w = 2^{\Omega(n)}$ generates a distribution that is $\approx 1/4$-far from $(\mathbf{U}_n, b(\mathbf{U}_n))$, as desired.

2.3 Sampling lower bounds against input-output pairs

We now begin our sketch of our second main theorem, Theorem 2. Recall that it roughly says that there is an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that any distribution sampled by a width $2^{\Omega(n)}$ ROBP has statistical distance $\frac{1}{2} - 2^{-\Omega(n)}$ from $(\mathbf{U}_n, b(\mathbf{U}_n))$.

Using the equivalence theorems sketched in Section 2.1, we can convert this theorem into a simple statement about complex samplers, and prove that instead. The new version of this theorem (Theorem 16) says that there is an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that any distribution generated by a width $2^{\Omega(n)}$ complex sampler has statistical distance $\frac{1}{2} - 2^{-\Omega(n)}$ from $(\mathbf{U}_n, b(\mathbf{U}_n))$. We sketch its proof below.

As in [DW12, Vio14, Vio20], we can take b to be a certain type of extractor. For a family $\mathcal{X}$ of distributions over $\{0,1\}^n$, an extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ for $\mathcal{X}$ with error ε offers the guarantee that for any $\mathbf{X} \in \mathcal{X}$ and $c \in \{0,1\}$, it holds that $\Pr[\mathsf{Ext}(\mathbf{X}) = c] = \frac{1}{2} \pm \varepsilon$. In order to work, an extractor typically requires the distributions in $\mathcal{X}$ to have some amount of (min-)entropy.

The general approach will be as follows: let $\mathbf{X} \sim \{0,1\}^{n+1}$ be a distribution generated by a complex sampler. We wish to show that as long as its width is not too large, then $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{-\Omega(n)}$. Recall that by definition of statistical distance, it suffices to find a test $S \subseteq \{0,1\}^{n+1}$ such that $\Pr[\mathbf{X} \in S] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) \in S] \geq \frac{1}{2} - 2^{-\Omega(n)}$. It is straightforward to verify that if $\mathbf{X}$ is a convex combination of distributions $\{\mathbf{Y}^{(j)}\}_j$, then it suffices to show $\Pr[\mathbf{Y}^{(j)} \in S] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) \in S] \geq \frac{1}{2} - 2^{-\Omega(n)}$ for all j.

Thus the goal will be to simultaneously break $\mathbf{X}$ into a convex combination of "nice" distributions, while building a set S, that together offer the following guarantees: (1) each participant $\mathbf{Y}^{(j)}$ in the convex combination hits S with probability $\approx 1/2$; and (2) the distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ hits S with probability ≈ 0.

A straw man approach via small-space extractors What we call a complex sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ was originally introduced in [KRVZ11] as a small-space source. Thus, a natural approach is try to take b as an extractor for small-space sources. As we will see, such a black box approach will fail.

Recall that we want to simultaneously design a test S and break $\mathbf{X}$ into a convex combination of distributions $\{\mathbf{Y}^{(j)}\}$. We start with the convex combination just containing $\mathbf{X}$ itself, and focus on building S, which is currently empty. As we want each $\mathbf{Y}^{(j)}$ to hit S with probability $\approx 1/2$, and $(\mathbf{U}_n, b(\mathbf{U}_n)) \approx 0$, it can only help to add both of the following sets to S

$$T_0 := \{(x, 0) : x \in \{0,1\}^n, b(x) = 1\},$$
$$T_1 := \{(x, 1) : x \in \{0,1\}^n, b(x) = 0\},$$

since $(\mathbf{U}_n, b(\mathbf{U}_n))$ will never hit either of these sets. Thus we currently have $\Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) \in S] = 0$, as desired. But is it true that $\Pr[\mathbf{Y}^{(j)} \in S] \approx 1/2$ for every j?

This is a little difficult to analyze with the current state of the convex combination, since it just contains the single distribution $\mathbf{X}$, which is a complex sampler. So the next step is to decompose $\mathbf{X}$ into a convex combination of more distributions, which will hopefully have nicer structure. The first natural thing to do is consider the convex combination $\{\mathbf{Y}^{(j)}\}$ induced by fixing (conditioning on) the last bit of $\mathbf{X}$ to each possible value $0, 1$.

Consider now each participant $\mathbf{Y}^{(j)}$ in the convex combination we have constructed. Notice the last bit is fixed to 0 or 1, and assume that $\mathbf{Y}^{(j)}$ can be generated by a complex sampler.[4] Without loss of generality, assume its last bit is 0, and let $\mathbf{Y}^{(j)}_{\leq n}$ denote its first n bits. Notice that the probability that $\mathbf{Y}^{(j)}$ hits S is the same as the probability that $\mathbf{Y}^{(j)}$ hits T_0, which is the same as the probability that $b(\mathbf{Y}^{(j)}_{\leq n}) = 1$.

If b is an extractor for small-space sources, and $\mathbf{Y}^{(j)}_{\leq n}$ has a decent amount of entropy, then we know that $\Pr[b(\mathbf{Y}^{(j)}_{\leq n}) = 1] = \frac{1}{2} \pm \varepsilon \approx 1/2$ by the extractor property, and we are done. The problem, of course, is that we have no guarantee that $\mathbf{Y}^{(j)}$ has any entropy. Indeed, it could have been the case that our complex sampler $\mathbf{X}$ started out with no entropy at all. At the same time, even if $\mathbf{Y}^{(j)}$ has very little entropy, it could be the case that $\mathbf{Y}^{(j)}$ is "close" to having high (min-)entropy: say, if it outputs 0 with probability $1/2$ and outputs the uniform distribution over its support with probability $1/2$. Thus, in this case, we cannot simply treat $\mathbf{Y}^{(j)}$ as a constant and hope to get a good bound on statistical distance.

To fix this issue, a natural approach might be to continue building our test S, and decomposing $\mathbf{X}$ into a convex combination of even more distributions, as follows. Define for each j a set $\mathsf{Bad}^{(j)}$, which includes all elements hit by $\mathbf{Y}^{(j)}$ with too high a probability. Then, add each set $\mathsf{Bad}^{(j)}$ to S. It may now be the case the $\Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) \in S]$ grows a little in this case (away from 0, where we wanted to keep it), but this won't be a big issue, since each set $\mathsf{Bad}^{(j)}$ can be made to be quite small, and the number of such sets is currently just 2 (the number of elements in our current convex combination). So we will still get $\Pr[(\mathbf{U}_n, b(\mathbf{U}_n))] \approx 0$, as desired.

Now, to try to obtain $\Pr[\mathbf{Y}^{(j)} \in S] \approx 1/2$, we consider each $\mathbf{Y}^{(j)}$ currently in our convex combination, and condition on whether or not it hits $\mathsf{Bad}^{(j)}$. Let $\mathbf{Y}^{(j)}_1$ denote conditioning $\mathbf{Y}^{(j)}$ on hitting this bad set, and let $\mathbf{Y}^{(j)}_0$ denote conditioning $\mathbf{Y}^{(j)}$ on *not* hitting this bad set. We have now written $\mathbf{X}$ as a convex combination over $\{\mathbf{Y}^{(j)}_a\}$ where a ranges over $\{0, 1\}$.

Let us now see if $\Pr[\mathbf{Y}^{(j)}_a \in S] \approx 1/2$ for each element in this convex combination. If $a = 1$, then the answer is certainly yes. In fact, the probability is 1, since it is supported on $\mathsf{Bad}^{(j)}$, which is contained in S. On the other hand, if $a = 0$, then $\mathbf{Y}^{(j)}_a$ will have high min-entropy, since we threw away all the high probability elements with our conditioning.[5] Now that $\mathbf{Y}^{(j)}_a$ has high min-entropy, one might be tempted to finally apply the small-space extractor to finish off the proof.

But we have now arrived at the major flaw of this straw man argument: $\mathbf{Y}^{(j)}_a$ is no longer guaranteed to be (samplable by) a complex sampler of the same width! Moreover, given the conditioning we performed to obtain $\mathbf{Y}^{(j)}_a$ from $\mathbf{Y}^{(j)}$, it seems quite challenging (if possible at all) to modify the complex sampler that generates $\mathbf{Y}^{(j)}$ into one that generates $\mathbf{Y}^{(j)}_a$.

Thus, we reach a road block with this technique, and must turn elsewhere.

[4] This is not obvious, but it is a minor fixable issue, which will be dwarfed by a major issue that is about to appear with this straw man argument.

[5] This technically assumes $\mathsf{Bad}^{(j)}$ didn't cover an enormous part of the support of $\mathbf{Y}^{(j)}$, but it is not too hard to handle this case separately.

Boosting entropy by reducing to independent blocks Let us recap what went wrong above. We had an extractor b for a certain family $\mathcal{X}$ of distributions, but we were feeding it distributions from this family that did not have enough (min-)entropy. Thus, we tried to perform some conditioning to boost the min-entropy. And while our conditioning succeeded at this task, we were unable to prove that the distribution remained in $\mathcal{X}$ after the conditioning, thereby preventing the extractor from working.

Thus, the idea here will be to reduce complex samplers (small-space sources) to a family of distributions that are immune to the conditioning we must perform in order to boost the entropy. In particular, we will take $b : \{0,1\}^n \to \{0,1\}$ to be a two-source extractor of the form $\mathsf{Ext} : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}$ (in particular, we take the inner product extractor, which has error $\varepsilon = 2^{-\Omega(n)}$).

Given this idea, let's start from the beginning (but go at a faster pace). Recall we want to design a test S and break $\mathbf{X}$ into a convex combination of distributions $\{\mathbf{Y}^{(j)}\}$ such that $\Pr[\mathbf{Y}^{(j)} \in S] \approx 1/2$ and $\Pr[(\mathbf{U}_n, b(\mathbf{U}_n))] \approx 0$. Before breaking down $\mathbf{X}$, let's set S to again contain the sets T_0, T_1, as defined in the previous section (but with b now denoting the two source extractor).

Now, consider the random walk $\mathbf{W}$ in the complex sampler that generates $\mathbf{X}$. Note that if we fix the vertex it hits in the central layer, we get that $\mathbf{X}$ can be written as a convex combination of distributions $\{\mathbf{Y}^{(j)}\}_j$, where each $\mathbf{Y}^{(j)}$ is of the form $(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{c}^{(j)}) \sim \{0,1\}^{n/2} \times \{0,1\}^{n/2} \times \{0,1\}$. Furthermore, by a standard observation [KM04, KM05], it will hold that $\mathbf{A}^{(j)}$ is independent from $\mathbf{B}^{(j)}, \mathbf{c}^{(j)}$.

Now, increase the number of participants in the convex combination $\{\mathbf{Y}^{(j)}\}$ by fixing each $\mathbf{c}^{(j)}$ to a value in $\{0,1\}$. Each $\mathbf{Y}^{(j)}$ is now of the form $(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, c^{(j)})$ for some fixed $c^{(j)} \in \{0,1\}$ and $\mathbf{A}^{(j)}, \mathbf{B}^{(j)}$ independent. So far our set S just contains T_0 and T_1, so we currently have $\Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) \in S] = 0$, as desired. But is each $\Pr[\mathbf{Y}^{(j)} \in S] = \Pr[(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, c^{(j)})] \approx 1/2$? As with our earlier argument, we assume $c^{(j)} = 0$ without loss of generality, and we see that this is the same thing as asking whether $\Pr[b(\mathbf{A}^{(j)}, \mathbf{B}^{(j)})] \approx 1/2$. Since $\mathbf{A}^{(j)}, \mathbf{B}^{(j)}$ are independent and b is a two source extractor, this is true as long as they both have high enough entropy.

As before, this is not guaranteed, and we must define some bad sets which will help us do one last step of conditioning. We let $\mathsf{Bad}_A^{(j)}$ be all elements that $\mathbf{A}^{(j)}$ with relatively high probability, and let $\mathsf{Bad}_B^{(j)}$ be all elements that $\mathbf{B}^{(j)}$ hits with relatively high probability. We perform one final conditioning, fixing over whether $\mathbf{A}^{(j)}$ hits $\mathsf{Bad}_A^{(j)}$, and fixing over whether $\mathbf{B}^{(j)}$ hits $\mathsf{Bad}_B^{(j)}$. We can now write $\mathbf{X}$ as a convex combination over distributions of the form

$$\{(\mathbf{A}_z^{(j)}, \mathbf{B}_{z'}^{(j)}, c^{(j)})\}_{j,z,z'},$$

where $z, z' \in \{0,1\}$ indicate whether the corresponding variable hit its bad set. Finally, add every $(x^{(j)}, y^{(j)}, c^{(j)})$ with *either* $x^{(j)} \in \mathsf{Bad}_A^{(j)}$ *or* $y^{(j)} \in \mathsf{Bad}_B^{(j)}$ to the test set S.

Now, the first thing we need is that $\Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) \in S] \approx 0$. This won't be an issue, since each bad set can be made to be quite small, and the number of such sets is the number of current participants in the convex combination, which is $O(w)$ (from our first fixing of $\mathbf{W}$).

Next, we need to show that each $\Pr[(\mathbf{A}_z^{(j)}, \mathbf{B}_{z'}^{(j)}, c^{(j)}) \in S]$ is roughly $1/2$. If either z or z' is 1, someone hit its bad set and so this probability will always be 1. If both indicators are 0, *no one* hit their bad set, meaning that both $\mathbf{A}_z^{(j)}$ and $\mathbf{B}_{z'}^{(j)}$ will have high entropy. Furthermore, it is straightforward to verify that given such a conditioning (i.e., where z, z' are both 0), both of these random variables remain independent!

Because they each have high entropy and are independent, b will successfully extract from them, which tells us $\Pr[(\mathbf{A}_z^{(j)}, \mathbf{B}_{z'}^{(j)}, c^{(j)}) \in S] \approx 1/2$ from our discussion before. This completes the proof that $|\Pr[\mathbf{X} \in S] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n))]| \geq \frac{1}{2} - 2^{-\Omega(n)}$.

2.4 Sampling lower bounds against codes

We now begin our sketch of our second main theorem, Theorem 3. Recall that it roughly says that for any distribution $\mathbf{Q} \sim \{0,1\}^n$ that is uniform over a (ρ, L) list decodable code with good parameters, any distribution sampled by an ROBP (whose width is not too large) will have statistical distance close to 1 from $\mathbf{Q}$.

Using the equivalence theorems sketched in Section 2.1, we can convert this into a statement about complex samplers, and prove that instead. The new version of this theorem (Theorem 22) says that for any distribution $\mathbf{Q} \sim \{0,1\}^n$ that is uniform over a (ρ, L) list decodable code with good parameters, any distribution sampled by a complex sampler (whose width is not too large) will have statistical distance close to 1 from $\mathbf{Q}$.

The proof uses two main ingredients. First, it uses a known lemma [KRVZ11] which says that any (distribution generated by a) complex sampler $\mathbf{X} \sim \{0,1\}^n$ can be written as a convex combination of a few product distributions. More formally: if the sampler has width w, then for any r, ℓ with $r\ell = n$, it can be written as a convex combination of w^r distributions of the form $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_r) \sim (\{0,1\}^\ell)^r$, where each $\mathbf{Y}_i$ is independent.

The second ingredient, which we will prove, is that product distributions, i.e., those of the form $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_r) \sim (\{0,1\}^\ell)^r$ with each $\mathbf{Y}_i$ independent, are statistically far from (distributions that are uniform over) good list decodable codes.

At this point, it would be nice to conclude that the original complex sampler $\mathbf{X}$ must also be far from a good list decodable code $\mathbf{Q}$. In particular, we would like to argue that $\mathbf{X}$ is a convex combination of product distributions $\{\mathbf{Y}^{(j)}\}$, and each of these product distributions is far from $\mathbf{Q}$, so $\mathbf{X}$ must be far from $\mathbf{Q}$. Unfortunately, the bounds we are trying to lift are in the wrong direction: it is true that if each $\mathbf{Y}^{(j)}$ is close to $\mathbf{Q}$, then $\mathbf{X}$ is close to $\mathbf{Q}$, but it is not necessarily true that if each $\mathbf{Y}^{(j)}$ is far from $\mathbf{Q}$, then $\mathbf{X}$ is far from $\mathbf{Q}$. Indeed, it could be the case that each $\mathbf{Y}^{(j)}$ is constant over a (different) codeword, which would make each $\mathbf{Y}^{(j)}$ extremely far from $\mathbf{Q}$, but still allow the overall convex combination over $\{\mathbf{Y}^{(j)}\}$ to exactly sample $\mathbf{Q}$.

Given the above counterexample, a new idea might be to try to argue that *as long as there aren't too many distributions* $\mathbf{Y}^{(j)}$ participating in the convex combination, then if each $\mathbf{Y}^{(j)}$ is far from $\mathbf{Q}$, then $\mathbf{X}$ is relatively far from $\mathbf{Q}$. It turns out this is true, but the corresponding lower bounds on $|\mathbf{X} - \mathbf{Q}|$ that it yields are still not as strong as we would like. To get the strongest possible bounds, we need a slightly more nuanced way to combine our two key ingredients.

Below, we sketch a proof for our second ingredient, and show to combine it with the first ingredient to yield our desired lower bound on $|\mathbf{X} - \mathbf{Q}|$.

Anti-concentration of product distributions in Hamming balls We now argue that product distributions are far from sampling good list decodable codes. Let $\mathbf{Q} \sim \{0,1\}^n$ be a (ρ, L) list decodable code, and let $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_r) \sim (\{0,1\}^\ell)^r$ be such that each $\mathbf{Y}_i$ is independent and $r\ell = n$. Furthermore, we will need the product distribution to have a reasonable number of components r, or else it could clearly sample the code perfectly (if $r = 1$). Towards this end, we enforce the mild requirement $r \geq 1/\rho$, and thus $\ell \leq \rho n$. For a good list decodable code, we can think of $\rho = \Omega(1)$, and thus we just require $r \geq O(1)$.

Now, the key intuition about product distributions is that for any point x in the space $\{0,1\}^n$ to which $\mathbf{Y}$ does not assign too much probability, the following must hold: if we draw a Hamming ball $\mathcal{B}(x)$ around x whose radius is not too small, then the vast majority of probability weight assigned to $\mathcal{B}(x)$ by $\mathbf{Y}$ does *not* land on x. In symbols, $\Pr[\mathbf{Y} \in \mathcal{B}(x)] \gg \Pr[\mathbf{Y} = x]$.

Let us formalize this intuition a little more. Fix any $x \in \{0,1\}^n$, and let $p := \Pr[\mathbf{Y} = x]$. Consider now the ball $\mathcal{B}_\ell(x)$ around x of radius ℓ. Now, parse x as $x = (x_1, \ldots, x_r) \in (\{0,1\}^\ell)^r$. Since $\mathbf{Y}$ is a product distribution consisting of r components, there must be at least some $i \in [r]$ such that $\Pr[\mathbf{Y}_i = x_i] \leq p^{1/r}$. Consider now the set T of all strings of the form $(x_1, \ldots, x_{i-1}, z, x_{i+1}, \ldots, x_r) \in (\{0,1\}^\ell)^r$, where each x_j is fixed as before, but z can be taken as any element in $\{0,1\}^\ell$. Then $\mathbf{Y}$ assigns probability at least $\Pr[\mathbf{Y} = x]/p^{1/r}$ to the set T, and of course T is in the ball $\mathcal{B}_\ell(x)$. Thus, we get that $\Pr[\mathbf{Y} \in \mathcal{B}(x)] \geq \Pr[\mathbf{Y} = x]/p^{1/r}$, and therefore $\Pr[\mathbf{Y} \in \mathcal{B}(x)] \gg \Pr[\mathbf{Y} = x]$, as long as p is not too big.

Now, how can we use this to show $|\mathbf{Y} - \mathbf{Q}|$ is large? Well, by definition of statistical distance, it suffices to pick a set $S \subseteq \{0,1\}^n$ and show that $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{Y} \in S]$ is large. Our choice for S will be all codewords from $\mathbf{Q}$, with some "bad" codewords Bad removed. Then to lower bound $\Pr[\mathbf{Q} \in S]$, we just need to show that $\mathbf{Q}$ lands in Bad with not too high probability: in particular, we can just require that Bad is not too big. And to upper bound $\Pr[\mathbf{Y} \in S]$, we just need to show that the probability that $\mathbf{Y}$ lands on a "not-bad" codeword is small.

So what should we choose as the set Bad? You guessed it: a small set of codewords assigned the highest probability by $\mathbf{Y}$ (say, all codewords assigned probability $\geq p$ for some threshold probability p). As long as this set isn't too big, we will have $\Pr[\mathbf{Q} \in S]$ be very close to 1. And as long as we removed the codewords assigned very high probability by $\mathbf{Y}$, we will have that $\Pr[\mathbf{Y} \in S]$ is very close to 0. We argue the latter, below.

To upper bound the probability that $\mathbf{Y}$ lands in S, we consider the sum $\sum_{q \in S} \Pr[\mathbf{Y} = q]$. By our anti-concentration observation above, this sum is at most $p^{1/r} \cdot \sum_{q \in S} \Pr[\mathbf{Y} \in \mathcal{B}_\ell(q)]$. Intuitively, we will now want to make sure that (i) r is not too big, because otherwise $p^{1/r}$ will be too big; and (ii) ℓ is not too big, because otherwise many of the balls $\{\mathcal{B}_\ell(q)\}$ in the sum will have big overlaps, causing probabilities to be multi-counted and the overall sum to be large.

It turns out that the best tradeoff occurs at setting $r = 1/\rho$ and $\ell = \rho n$. This is because a good list decodable code will have $\rho = \Omega(1)$, which yields $p^{1/r} = p^{\Omega(1)}$, which will be quite small as long as we originally set our threshold probability p to be low enough. Similarly, by definition of list decodability, we will have that any point x in the space $\{0,1\}^n$ will appear in at most L balls $\{\mathcal{B}_\ell(q)\}_{q \in S}$. This implies $\sum_{q \in S} \Pr[\mathbf{Y} \in \mathcal{B}_\ell(q)] \leq L$, since the probability $\mathbf{Y}$ assigns to any point $x \in \{0,1\}^n$ is counted at most L times. For a good list decodable code, L is quite small, and we finally have that $\Pr[\mathbf{Y} \in S]$ will be very close to 0.

Thus, as long as our original product distribution $\mathbf{Y} \sim (\{0,1\}^\ell)^r$ had $r \approx 1/\rho$ and $\ell \approx \rho n$, we have a set $S \subseteq \{0,1\}^n$ that makes $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{Y} \in S]$ very close to 1, implying the statistical distance $|\mathbf{Y} - \mathbf{Q}|$ is very close to 1.

From complex samplers to product distributions Now, the question is: how do we use the fact that product distributions are far from good list decodable codes in order to argue that complex samplers are far from list decodable codes? Well, let $\mathbf{X} \sim \{0,1\}^n$ be the complex sampler, and $\mathbf{Q} \sim \{0,1\}^n$ be the list decodable code. We need to show that $|\mathbf{X} - \mathbf{Q}|$ is large. So we write $\mathbf{X}$ as a convex combination of at most w^r product distributions $\{\mathbf{Y}^{(j)}\}_j$, each of the form $\mathbf{Y}^{(j)} = (\mathbf{Y}_1^{(j)}, \ldots, \mathbf{Y}_r^{(j)}) \sim (\{0,1\}^\ell)^r$.

For any tester $S \subseteq \{0,1\}^n$, the statistical distance $|\mathbf{X} - \mathbf{Q}|$ is lower bounded by $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{X} \in S]$. Furthermore, it is easy to verify that this, in turn, is lower bounded by the worst $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{Y}^{(j)} \in S]$ (meaning the one that gives the smallest value). So what S should we pick?

From above, we know that as long as we set $r = 1/\rho$ and $\ell = \rho n$, then each $\mathbf{Y}^{(j)}$ has a test $S^{(j)}$ which makes $\Pr[\mathbf{Y}^{(j)} \in S^{(j)}]$ very close to 0. Furthermore $S^{(j)}$ is of the form $Q - \mathsf{Bad}^{(j)}$, where Q is the support

of the code and $\mathsf{Bad}^{(j)}$ is some small bad set. Thus, since we want a *single* test $S \subseteq \{0,1\}^n$ guaranteed to make $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{Y}^{(j)} \in S]$ small *for every* j, we can simply take the test to be $S = Q - \cup_j \mathsf{Bad}^{(j)}$. Indeed, this guarantees that each $\Pr[\mathbf{Y}^{(j)} \in S]$ will be very close to 0, and as long as there are not too many elements in the convex combination, our total collection of bad elements won't be too big, and $\Pr[\mathbf{Q} \in S]$ will stay very close to 1. Thus we get that $|\mathbf{X} - \mathbf{Q}|$ is close to 1, as desired.

2.5 A direct product theorem

We now begin our sketch of our fourth (and final) main theorem, Theorem 4. Recall that it roughly says that if a distribution $\mathbf{Q} \sim \{0,1\}^n$ has statistical distance $\geq \delta$ from distributions sampled by ROBPs of width w, then $\mathbf{Q}^{\otimes t} \sim \{0,1\}^{nt}$ has statistical distance $\geq 1 - 2^{-\Omega(t\delta^2)}$ from distributions sampled by ROBPs of width roughly the same width. Here, recall that $\mathbf{Q}^{\otimes t}$ refers to a sequence of t independent copies of $\mathbf{Q}$.

Using the equivalence theorems sketched in Section 2.1, we can convert this into a statement about complex samplers, and prove that instead. The new version of this theorem (Theorem 27) says that if a distribution $\mathbf{Q} \sim \{0,1\}^n$ has statistical distance $\geq \delta$ from distributions generated by complex samplers of width w, then $\mathbf{Q}^{\otimes t}$ has statistical distance $\geq 1 - 2^{-\Omega(t\delta^2)}$ from distributions generated by complex samplers of roughly the same width. We sketch its proof below.

A sequence of random variables that all contribute their fair share Let $\mathbf{X} \sim \{0,1\}^{nt}$ be a distribution generated by a complex sampler of width w, and parse it as $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$, where each $\mathbf{X}_i \sim \{0,1\}^n$ need not be independent. Recall that $\mathbf{Q}^{\otimes t} \sim \{0,1\}^{nt}$ is of the form $\mathbf{Q}^{\otimes t} = (\mathbf{Q}_1, \ldots, \mathbf{Q}_t)$, where each $\mathbf{Q}_i \sim \{0,1\}^n$ is an independent copy of $\mathbf{Q}$. We would like to argue

$$|(\mathbf{X}_1, \ldots, \mathbf{X}_t) - (\mathbf{Q}_1, \ldots, \mathbf{Q}_t)| \geq 1 - 2^{-\Omega(t\delta^2)}, \tag{1}$$

given that any for any complex sampler $\mathbf{X}' \sim \{0,1\}^n$ of width w it holds that $|\mathbf{X}' - \mathbf{Q}| \geq \delta$.

The first observation is that any of the $\mathbf{X}_i \sim \{0,1\}^n$ can be generated by a complex sampler of width w. Indeed, even though it represents a sequence of bits generated in the middle of the complex sampler $\mathbf{X}$, it is easy to create a new complex sampler $\mathcal{B}$ of the same width that only generates $\mathbf{X}_i$, simply by: (1) copying the complex sampler that creates $\mathbf{X}$; (2) throwing out all layers that do not produce bits corresponding to $\mathbf{X}_i$; (3) adding a new start vertex v_{start}; (4) connecting that start vertex the first layer remaining in $\mathcal{B}$, using the appropriate probabilities; and (5) merging the first two layers of $\mathcal{B}$, to deal with the fact that the edges leaving v_{start} currently have no output labels (this is identical to the merge step described in the sampler constructions in Section 2.2).

Thus, we are guaranteed that for each $\mathbf{X}_i \sim \{0,1\}^n$ and $\mathbf{Q}_i \sim \{0,1\}^n$, it holds that $|\mathbf{X}_i - \mathbf{Q}_i| \geq \delta$ by the theorem hypothesis. The question now is: is this enough to guarantee that the statistical distance blows up in Equation (1)?

Well, if each $\mathbf{X}_i$ were independent, it is not too hard to show that the answer is yes. However, this is of course not guaranteed to be the case, since the $\mathbf{X}_i$'s are consecutive slices of the same complex sampler $\mathbf{X}$. Indeed, without further examination, it could potentially be the case that for any $x \in \{0,1\}^n$ and $i \in [n]$, the distributions $(\mathbf{X}_{-i} \mid \mathbf{X}_i = x)$ and $(\mathbf{Q}_{-i}^{\otimes})$ are identical.[6] In this case, we cannot hope to lower bound Equation (1) by anything more than δ.

In some sense, the above adversarial example represents a situation where each $\mathbf{X}_i$ is *not* contributing its "fair share" to the statistical distance in Equation (1). In order to force each $\mathbf{X}_i$ to be a contributing member,

[6] For a random variable $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$, the notation $\mathbf{X}_{-i}$ denotes $\mathbf{X}$ with $\mathbf{X}_i$ removed.

we would like a different guarantee than just $|\mathbf{X}_i - \mathbf{Q}_i| \geq \delta$. One natural way to encode the idea that each $\mathbf{X}_i$ is contributing its fair share is to require that for every $i \in [n]$ and $x \in (\{0,1\}^n)^{i-1}$,

$$|(\mathbf{X}_i \mid \mathbf{X}_1, \ldots, \mathbf{X}_{i-1} = x) - (\mathbf{Q}_i \mid \mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1} = x)| \geq \delta. \tag{2}$$

This leaves us with two questions: (i) Given a guarantee like Equation (2), can we actually prove Equation (1)? (ii) Is the guarantee given in Equation (1) even true? If we can answer both questions in the affirmative, then we are done.

It turns out that (i) is true, but it is a little cumbersome to do so using statistical distance. To avoid this, we use simple and well known facts to convert the statement into one about squared Hellinger distance, which can further be phrased in terms of the *Bhattacharyya coefficient*. Phrasing (i) in this way allows for a simple inductive proof, which we can then convert back to a result about statistical distance.

It also turns out that (ii) is true. To see why, note that $(\mathbf{Q}_i \mid \mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1} = x)$ is just the same distribution as $\mathbf{Q} \sim \{0,1\}^n$, since each $\mathbf{Q}_i$ is an independent copy of $\mathbf{Q}$. Furthermore, it is straightforward to show that the distribution $(\mathbf{X}_i \mid \mathbf{X}_1, \ldots, \mathbf{X}_{i-1} = x)$ can be generated by a width w complex sampler, using a similar idea to the one we presented for why $\mathbf{X}_i$ has this property. Thus the hypothesis of the direct product theorem implies Equation (2), and we are done.

Finally, we show that we can use our direct product theorem to get a strong complexity separation between sampling with AC^0 circuits and sampling with ROBPs. To do so, we recall our result from Section 2.3 that for any distribution $\mathbf{X} \sim \{0,1\}^n$ samplable by an ROBP of width $2^{\Omega(n)}$, it holds that $|\mathbf{X} - (\mathbf{U}_n, \mathsf{IP}(\mathbf{U}_n))| \approx 1/2$ for the inner product function IP. Thus we can now set $\mathbf{Q} = (\mathbf{U}_n, \mathsf{IP}(\mathbf{U}_n))$. The distribution $\mathbf{Q}^{\otimes t}$ becomes no harder to sample for AC^0 (since it can just sample each copy in parallel), whereas our direct product theorem shows that $|\mathbf{X} - \mathbf{Q}^{\otimes t}|$ becomes exponentially close to 1.

3 Preliminaries

Before we start our formal proofs, we introduce some basic notation, definitions, and facts.

General notation For any natural number $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, 2, \ldots, n\}$. Given a string $x \in \{0,1\}^n$ and index $i \in [n]$, we let x_i denote the i^{th} coordinate of x. Furthermore, for any $1 \leq i \leq j \leq n$, we let $x_{i \to j} := (x_i, \ldots, x_j) \in \{0,1\}^{j-i+1}$, we let $x_{\leq i} := x_{1 \to i}$, and we let $x_{<i} := x_{1 \to i-1}$. Given a permutation $\pi : [n] \to [n]$, we define $x^\pi := (x_{\pi(1)}, \ldots, x_{\pi(n)})$.

Basic probability definitions and notation All of the notation above also applies to random variables. For example, given a random variable $\mathbf{X} \sim \{0,1\}^n$, we let $\mathbf{X}_i \sim \{0,1\}$ denote its i^{th} coordinate, and we let $\mathbf{X}^\pi := (\mathbf{X}_{\pi(1)}, \ldots, \mathbf{X}_{\pi(n)})$. We let $\mathbf{U}_n$ denote the uniform random variable over $\{0,1\}^n$. When $\mathbf{U}_n$ appears in the same expression twice, it denotes the same random variable - that is, they are *not* independent. However, if $\mathbf{U}_n, \mathbf{U}_m$ appear in the same expression with $n \neq m$, these random variables are assumed to be independent.

Throughout, we slightly abuse notation and let $\mathbf{X} \sim \{0,1\}^n$ denote both a random variable and its underlying distribution. However, it should always be clear from context which interpretation is intended. The *min-entropy* of a random variable $\mathbf{X} \sim \{0,1\}^n$, denoted $H_\infty(\mathbf{X})$, is defined as the largest k such that $\Pr[\mathbf{X} = x] \leq 2^{-k}$ for all $x \in \text{support}(\mathbf{X})$. As is standard, we measure the distance between two distributions using statistical distance:

Definition 1. *The* statistical distance *between two discrete random variables* $\mathbf{X}, \mathbf{Y}$ *over* V *is defined as*

$$|\mathbf{X} - \mathbf{Y}| := \max_{S \subseteq V} |\Pr[\mathbf{X} \in S] - \Pr[\mathbf{Y} \in S]| = \frac{1}{2} \sum_{x \in V} |\Pr[\mathbf{X} = x] - \Pr[\mathbf{Y} = x]|.$$

We say that $\mathbf{X}, \mathbf{Y}$ are ε-close if the statistical distance between them is at most ε. The following so-called *data processing inequality* is very useful for bounding statistical distance:

Fact 1. *For any discrete random variables* $\mathbf{X}, \mathbf{Y}$ *over* V *and any function* $f : V \to W$*,*

$$|f(\mathbf{X}) - f(\mathbf{Y})| \leq |\mathbf{X} - \mathbf{Y}|.$$

We say that $\mathbf{X}$ is a convex combination of distributions $\mathbf{Y}_1, \ldots, \mathbf{Y}_k$ if $\mathbf{X} = \sum_i p_i \mathbf{Y}_i$, for some probabilities $\{p_i\}$ that sum to 1. That is, $\mathbf{X}$ samples from $\mathbf{Y}_i$ with probability p_i. Finally, we say that a distribution $\mathbf{X}$ is α-granular if $\Pr[\mathbf{X} = x]$ is an integer multiple of α for every $x \in \text{support}(\mathbf{X})$.

Bias, correlation, and covariance Our equivalence theorems will allow us to convert sampling lower bounds into worst-case and average-case lower bounds for computation. For this, we will need the following.

Definition 2. *The* bias *of a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *is defined as*

$$\text{bias}(f) := \mathbb{E}_x[(-1)^{f(x)}].$$

Definition 3. *The* correlation *between two Boolean functions* $f, g : \{0,1\}^n \to \{0,1\}$ *is defined as*

$$\text{corr}(f, g) := \mathbb{E}_x[(-1)^{f(x)}(-1)^{g(x)}] = \Pr_x[f(x) = g(x)] - \Pr_x[f(x) \neq g(x)].$$

Definition 4. *The* covariance *between two Boolean functions* $f, g : \{0,1\}^n \to \{0,1\}$ *is defined as*

$$\text{cov}(f, g) := \text{corr}(f, g) - \text{bias}(f)\,\text{bias}(g).$$

Basic coding theory definitions and facts We provide here some basic coding theory definitions. For all $i \in [n]$, we let $e_i \in \{0,1\}^n$ denote the i^{th} elementary basis vector: that is, the vector with 1 at coordinate i and 0 everywhere else. Given two points $x, y \in \{0,1\}^n$, the Hamming distance $\Delta(x, y)$ between x, y is the number of coordinates where they differ. Next, given a point $x \in \{0,1\}^n$, the Hamming ball $\mathcal{B}_r(x)$ centered at x with radius r is the collection of points in $\{0,1\}^n$ that are Hamming distance at most r from x. Each such ball has volume $\binom{n}{\leq r} := \sum_{i=0}^{r} \binom{n}{i}$. An error correcting code is defined as follows:

Definition 5. *An* (n, k, d) *code* $Q \subseteq \{0,1\}^n$ *is a collection of* 2^k *points such that the minimum Hamming distance between any two points is* d*. We call* k *its* dimension*, and* d *its* distance*. We say that* Q *is a* linear $[n, k, d]$ *code if it is also a subspace of* $\mathbb{F}_2^n$*.*

Given a linear $[n, k, d]$ code $Q \subseteq \mathbb{F}_2^n$, the *dual code* or *orthogonal complement* of Q is the set $Q^\perp := \{y \in \mathbb{F}_2^n : \langle x, y \rangle = 0, \forall x \in Q\}$, where $\langle \cdot, \cdot \rangle$ denotes the inner product over $\mathbb{F}_2$. Note that $Q^\perp \subseteq \mathbb{F}_2^n$ is a subspace of dimension $n - k$. The following well-known existential result is known as the *Gilbert-Varshamov bound* (for linear codes). It can be proven via a greedy construction.

Theorem 5 ([Gil52, Var57])**.** *There exists a linear* $[n, k, d]$ *code for all* n, k, d *satisfying* $2^k \leq 2^n / \binom{n}{\leq d-1}$*.*

Next, a list decodable code relaxes the distance requirement of an (n, k, d) code:

Definition 6. *A subset $Q \subseteq \{0,1\}^n$ is a (ρ, L) list decodable code if every Hamming ball in $\{0,1\}^n$ of radius at most ρn contains at most L points from Q.*

A straightforward application of the triangle inequality shows that every (n, k, d) code has the following list decoding properties:

Fact 2. *If $Q \subseteq \{0,1\}^n$ is an (n, k, d) code, then Q is (ρ, L) list decodable for $L = 1$ and any $\rho < \frac{d}{2n}$.*

Next, we will discuss our models for computing in small space.

3.1 Models for computing in small space

Read-once branching programs (ROBPs) are a popular model for computation in small space. We provide their standard definition, below.

Definition 7 (ROBP)**.** *An* ROBP *$\mathcal{B}$ of width w and length n is a directed acyclic graph $G = (V, E)$ consisting of $n + 1$ disjoint layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$, each holding w vertices. For every $i \in [n]$, each vertex $v \in V_{i-1}$ has exactly two outgoing edges into V_i, one of which is labeled* 0*, and the other labeled* 1*. There is a designated* start vertex $v_{\mathsf{start}} \in V_0$*, and a designated* accept vertex $v_{\mathsf{accept}} \in V_n$*.*

The branching program $\mathcal{B}$ computes a function $f_{\mathcal{B}} : \{0,1\}^n \to \{0,1\}$ as follows: on input $x \in \{0,1\}^n$, the program starts at v_{start} and traverses the unique path $P(x)$ whose edges are labeled with input bits $x_1, x_2, \ldots, x_n$. The program outputs 1 *if $P(x)$ terminates on v_{accept}, and* 0 *otherwise.*

As discussed, ROBPs are a useful model for computing boolean functions $f : \{0,1\}^n \to \{0,1\}$. But if we wish to use ROBPs to sample distributions over $\{0,1\}^m$, we need to extend the definition of ROBPs to model the computation of functions $f : \{0,1\}^n \to \{0,1\}^m$ with multi-bit outputs. We use the following definition, which can be thought of as a read-once version of standard multi-output branching programs, as defined in, e.g., [BFK+79, BC82, Bea89].

Definition 8 (Multi-output ROBP)**.** *A* multi-output ROBP *$\mathcal{B}$ of width w and (input) length n is a directed acyclic graph $G = (V, E)$ consisting of $n + 1$ disjoint layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$, each holding w vertices. For every $i \in [n]$, each vertex $v \in V_{i-1}$ has exactly two outgoing edges into V_i, one of which is labeled with the input bit* 0*, and the other labeled with the input bit* 1*. Each edge e is also labeled with* output bits *$\Gamma(e) \in \{0,1\}^*$, and we assume that all edges e between the same two layers V_{i-1}, V_i have the same output length $|\Gamma(e)| = \gamma_i \geq 0$. The* output length *of $\mathcal{B}$ is $m = \sum_i \gamma_i$. Finally, there is a designated* start vertex $v_{\mathsf{start}} \in V_0$*.*

The branching program $\mathcal{B}$ computes a function $f_{\mathcal{B}} : \{0,1\}^n \to \{0,1\}^m$ as follows: on input $x \in \{0,1\}^n$, the program starts at v_{start} and traverses the unique path $P(x)$ whose edges are labeled with input bits $x_1, x_2, \ldots, x_n$. The program outputs the concatenation of all output bits seen along this path, so that $f_{\mathcal{B}}(x) = (\Gamma(e))_{e \in P(x)}$.

It is straightforward to verify that Definition 8 is a strict generalization of Definition 7:

Fact 3. *For any function $f : \{0,1\}^n \to \{0,1\}$, there exists a width w ROBP (as per Definition 7) that computes f if and only if there exists a width w ROBP (as per Definition 8) that computes f.*

Because of this, we will omit the qualifier "multi-output" when referring to ROBPs, since the intended definition will either be clear from context (when computing functions with multi-bit outputs), or it will not matter (when computing functions with single-bit outputs). For brevity, we will also sometimes call a function $f : \{0,1\}^n \to \{0,1\}^m$ an ROBP, when we really mean that f is the function computed by an ROBP.

3.2 Models for sampling in small space

We now introduce our models for sampling in small space. We start with main motivating model of simply feeding uniform bits into an ROBP:

Definition 9 (ROBP sampler)**.** *An* ROBP sampler $\mathcal{B}$ *of width* w *and input length* ℓ *and output length* n *is just an ROBP with the same parameters (as per Definition 8). The distribution* $\mathbf{X} \sim \{0,1\}^n$ *sampled by the ROBP* $\mathcal{B}$ *is* $\mathbf{X} = f_{\mathcal{B}}(\mathbf{U}_\ell)$*, where* $f_{\mathcal{B}}$ *is the function computed by* $\mathcal{B}$*.*

The next type of sampler we consider was defined by by Kamp, Rao, Vadhan, and Zuckerman under the name of *small space sources* [KRVZ11].

Definition 10 (Complex sampler)**.** *A* complex sampler $\mathcal{B}$ *of width* w *and output length* n *is a directed acyclic graph* $G = (V, E)$ *consisting of* $n+1$ *disjoint layers* $V = V_0 \cup V_1 \cup \cdots \cup V_n$*, each holding* w *vertices. For every* $i \in [n]$*, each vertex* $v \in V_{i-1}$ *has an* arbitrary number of outgoing edges *into* V_i*, some of which are labeled* 0*, and the rest labeled* 1*. There is a designated start vertex* $v_{\mathsf{start}} \in V_0$*, and each vertex* $v \in V$ *has a probability distribution* p_v *over its outgoing edges. The distribution* $\mathbf{X} \sim \{0,1\}^n$ *sampled by* $\mathcal{B}$ *is the one generated by taking a random walk over* $\mathcal{B}$*, which starts at* v_{start}*, transitions according to* $\{p_v\}$*, and outputs the edge labels seen along the way.*

The last type of sampler we consider is identical to the complex sampler, except that each vertex in the branching program is restricted to have out-degree exactly two.

Definition 11 (Simple sampler)**.** *A* simple sampler $\mathcal{B}$ *of width* w *and output length* n *is a directed acyclic graph* $G = (V, E)$ *consisting of* $n+1$ *disjoint layers* $V = V_0 \cup V_1 \cup \cdots \cup V_n$*, each holding* w *vertices. For every* $i \in [n]$*, each vertex* $v \in V_{i-1}$ *has* exactly two outgoing edges *into* V_i*, one of which is labeled* 0*, and the other labeled* 1*. There is a designated start vertex* $v_{\mathsf{start}} \in V_0$*, and each vertex* $v \in V$ *has a probability distribution* p_v *over its outgoing edges. The distribution* $\mathbf{X} \sim \{0,1\}^n$ *sampled by* $\mathcal{B}$ *is the one generated by taking a random walk over* $\mathcal{B}$*, which starts at* v_{start}*, transitions according to* $\{p_v\}$*, and outputs the edge labels seen along the way.*

Since its definition is more restrictive, we of course have the following:

Remark 1. *Any simple sampler is also a complex sampler.*

For brevity, we will sometimes call a distribution $\mathbf{X} \sim \{0,1\}^n$ an (ROBP, complex, simple) sampler, when we really mean that $\mathbf{X}$ is the distribution sampled by an (ROBP, complex, simple) sampler. The following type of simple and complex samplers will be important when we convert simple and complex samplers into ROBP samplers.

Definition 12. *A complex sampler or simple sampler is called* α-granular *if each edge probability is an integer multiple of* α*.*

We now record some basic facts about simple and complex samplers.

Fact 4. *Any distribution* $\mathbf{X} \sim \{0,1\}^n$ *can be sampled by a simple sampler of width* $w = |\mathit{support}(\mathbf{X})|$*.*

Proof. We construct the simple sampler $\mathcal{B}$, consisting of graph $G = (V, E)$, as follows. Let $V = V_0 \cup V_1 \cup \cdots \cup V_n$, where V_0 consists of the single start vertex v_{start} and $V_i := \text{support}(\mathbf{X}_{1 \to i}) \subseteq \{0,1\}^i$ for each $i \in [n]$. Then for each $u \in V_{i-1}$ and $(u, b) \in V_i$, draw an edge from u to (u, b), label it with bit b, and give it probability $\Pr[\mathbf{X}_{1 \to i} = (u, b) \mid \mathbf{X}_{1 \to i-1} = u]$. □

The following is immediate by combining Fact 4 and Remark 1, but we give a different proof which allows for more flexibility when designing *granular* complex samplers.

Fact 5. *Any distribution $\mathbf{X} \sim \{0,1\}^n$ can be sampled by a complex sampler of width $w = |\mathit{support}(\mathbf{X})|$.*

Proof. We construct the simple sampler $\mathcal{B}$, consisting of graph $G = (V, E)$, as follows. Let $V = V_0 \cup V_1 \cup \cdots \cup V_n$, where V_0 consists of the single start vertex v_{start}, and each $V_i, i \in [n]$ is a fresh copy of $\mathrm{support}(\mathbf{X}) \subseteq \{0,1\}^n$. For each $v \in V_1$, draw an edge from v_{start} to v, label it with the bit v_1, and give it probability $\Pr[\mathbf{X} = v]$. Then, for every $i \in [n-1]$ and $v \in V_i$, draw an edge from v to its copy in V_{i+1}, label it v_{i+1}, and give it probability 1. □

Finally, the following fact is straightforward to show, by wiring two samplers in a series configuration and "merging" their boundaries.

Fact 6. *Let $\mathbf{X} \sim \{0,1\}^n, \mathbf{Y} \sim \{0,1\}^k$ be independent distributions, where each can be sampled by a simple (resp., complex) sampler of width w. Then the distribution $(\mathbf{X}, \mathbf{Y})$ can be sampled by a simple (resp., complex) sampler of width w. Moreover, if the samplers for $\mathbf{X}$ and $\mathbf{Y}$ were α-granular, then so is the sampler for $(\mathbf{X}, \mathbf{Y})$.*

3.3 Extensions to the unknown-order setting

As we have seen, our main *computational model* is the read-once branching program (Definition 8), and our main *sampling models* are the ROBP sampler (Definition 9), complex sampler (Definition 10), and simple sampler (Definition 11). Thus, our results are primarily focused on these models. However, it turns out that most of our results can easily be extended to stronger, *unknown-order* versions of these models. These are defined in the natural way, and we review them below.

Unknown-order computation The *unknown-order ROBP*, introduced by Forbes and Kelley [FK18], is simply an ROBP that can read its input in any order. More formally, it can be defined as a tuple $(\mathcal{B}, \pi)$, where $\mathcal{B}$ is an ROBP (as per Definition 7) that computes a function $f : \{0,1\}^n \to \{0,1\}$, and $\pi : [n] \to [n]$ is a permutation. The unknown-order ROBP then computes a function $g : \{0,1\}^n \to \{0,1\}$ defined as $g(x) := f(x^\pi)$.

We can extend the above definition to *multi-output unknown-order ROBPs* by allowing the ROBP to also write its output in any order. More formally, such an object can be defined as a tuple $(\mathcal{B}, \pi, \rho)$, where $\mathcal{B}$ is an ROBP (as per Definition 8), and $\pi : [n] \to [n], \rho : [m] \to [m]$ are permutations. The multi-output unknown-order ROBP then computes a function $g : \{0,1\}^n \to \{0,1\}^m$ defined as $g(x) := (f(x^\pi))^\rho$. Notice that multi-output unknown-order ROBPs strictly generalize unknown-order ROBPs defined above, and thus we henceforth omit the qualifier "multi-output."

Unknown-order sampling The most natural way to generalize our sampling models to the unknown-order setting is to allow our samplers to write their output in any order. For ROBP samplers, this is already taken care of in the definition of unknown-order ROBPs. Namely, an *unknown-order ROBP sampler* is an unknown-order ROBP $(\mathcal{B}, \pi, \rho)$ as defined above, and the distribution it generates is $(f(\mathbf{U}_n^\pi))^\rho$. Notice, however, that the permutation π does not affect the output distribution, whereas the permutation ρ certainly can.

We can allow complex samplers to write their output in any order using a similar formalization. That is, we define an *unknown-order complex sampler* to be a tuple $(\mathcal{B}, \pi)$, where $\mathcal{B}$ is a complex sampler (as

per Definition 10) that generates a distribution $\mathbf{X} \sim \{0,1\}^n$, and $\pi : [n] \to [n]$ is a permutation. The unknown-order complex sampler then generates the distribution $\mathbf{X}^\pi$.

Finally, we can define unknown-order simple samplers in an analogous way. That is, we define an *unknown-order simple sampler* to be a tuple $(\mathcal{B}, \pi)$, where $\mathcal{B}$ is a simple sampler (as per Definition 11) that generates a distribution $\mathbf{X} \sim \{0,1\}^n$, and $\pi : [n] \to [n]$ is a permutation. The unknown-order simple sampler then generates the distribution $\mathbf{X}^\pi$. Just like Remark 1, we have the following.

Remark 2. *Any unknown-order simple sampler is also an unknown-order complex sampler.*

Throughout, when we refer to the *width* of an unknown-order ROBP or sampler, we are referring to the width of the standard ROBP or sampler than underlies it. Just like with the known-order case, for brevity we sometimes refer to unknown-order ROBPs (or samplers) as the functions (or distributions) they compute.

4 Equivalence theorems

In this section, we will provide and prove a small collection of equivalence theorems. These will allow us to convert our main theorem statements into ones about simple and complex samplers and prove those instead. Furthermore, they will also allow us to obtain correlation bounds from our sampling lower bounds.

4.1 An equivalence theorem between complex samplers and sampling with ROBPs

First, we will prove that for any distribution $\mathbf{X} \sim \{0,1\}^n$, it holds that $\mathbf{X}$ can be sampled by a complex sampler if and only if $\mathbf{X}$ can be sampled by an ROBP (up to a small loss in parameters). More precisely, we prove the following.

Theorem 6. *For any distribution $\mathbf{X} \sim \{0,1\}^n$:*

- *If there is an ROBP of width w and input length ℓ that samples $\mathbf{X}$, then there exists a complex sampler of width $2w$ that samples $\mathbf{X}$.*
- *If there exists a complex sampler of width w that samples $\mathbf{X}$, then for any $\varepsilon > 0$, there exists an ROBP of width $7w$ and input length $\ell = 8nw \log(nw/\varepsilon)$ that samples a distribution that is ε-close to $\mathbf{X}$.*

In many settings, the second bullet of the above theorem will be applied to a complex sampler that is α-granular (Definition 12), for some $\alpha = 2^{-t}$ and $t \in \mathbb{N}$. In this case, we can strengthen this result to get an ROBP of width $7w$ and input length $\ell = 4nwt$ that *exactly* samples $\mathbf{X}$. Furthermore, we will also provide an alternate version of the second bullet that saves on input length at the cost of a greater blow-up in width.

For organizational convenience, we isolate the first item in Theorem 6 as the following lemma.

Lemma 1. *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists an ROBP of width w and input length ℓ that samples $\mathbf{X}$, then there exists a complex sampler of width $2w$ that samples $\mathbf{X}$.*

To prove this result, we will need to show that complex samplers can efficiently simulate ROBP samplers. In other words, we will need to transform an ROBP sampler into a complex sampler that generates the same distribution. This will require some basic local modifications to the ROBP sampler, but will not require any complex machinery.

The second bullet in Theorem 6 will be more challenging to prove. Here, we will need to show that ROBP samplers can efficiently simulate complex samplers; or rather, that any complex sampler can be

transformed into an ROBP sampler that generates the same distribution. This transformation will proceed in two stages.

The first stage addresses the following issue: a complex sampler can generate distributions that assign certain elements arbitrarily precise probabilities (since its edges may be assigned probabilities that are arbitrary reals), whereas the distribution generated by an ROBP sampler has some fundamental limit to its precision (the probability assigned to any element will be an integer multiple of $2^{-\ell}$, where ℓ is the input length of the ROBP). Thus, the distributions generated by ROBP samplers are inherently "granular," so if we would like to transform a complex sampler into an ROBP sampler, we would first like to transform the complex sampler into a *granular* complex sampler (as per Definition 12), without introducing too much error.

Kamp, Rao, Vadhan, and Zuckerman proved a lemma of exactly this type (in the language of *small space sources*), which we use as the first stage of our transformation.

Lemma 2 ([KRVZ11, Lemma 8.4])**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be a complex sampler of width w. For any $\alpha = 1/A$ with $A \in \mathbb{N}$, there exists an α-granular complex sampler $\mathbf{X}^* \sim \{0,1\}^n$ of width w that is $(\alpha n w)$-close to $\mathbf{X}$.*

After we use the above lemma to make our complex sampler into a granular complex sampler, the next step will be to transform this granular complex sampler directly into an ROBP sampler. We will prove the following, which is the second stage of the transformation needed to prove the second bullet of Theorem 6.

Lemma 3. *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists a (2^{-t})-granular complex sampler of width w that samples $\mathbf{X}$, then there exists an ROBP of width $7w$ and input length $\ell = 4nwt$ that samples $\mathbf{X}$.*

Indeed, Lemma 3 is the result mentioned earlier, which strengthens the second bullet of Theorem 6 when the complex sampler is granular. Given Lemmas 1 to 3, we can easily prove Theorem 6 as follows.

Proof of Theorem 6. The first bullet is clearly true by Lemma 1. For the second bullet: let $\mathbf{X} \sim \{0,1\}^n$ be any distribution that can be generated by a complex sampler of width w, and let $\varepsilon > 0$ be any positive real number. Now, set $\alpha = 2^{-t}$ and $t = \lceil \log(nw/\varepsilon) \rceil$. By Lemma 2, there is a 2^{-t}-granular complex sampler $\mathbf{X}^* \sim \{0,1\}^n$ of width w that is $(2^{-t}nw \leq \varepsilon)$-close to $\mathbf{X}$. By Lemma 3, there is an ROBP of width $7w$ and input length $\ell = 4nwt = 4nw\lceil \log(nw/\varepsilon) \rceil \leq 8nw \log(nw/\varepsilon)$ that exactly samples $\mathbf{X}^*$. The result follows. □

Thus, if we can show Lemmas 1 and 3, then we are done. We prove these lemmas in the next two subsections. We start with Lemma 3, since its proof is more interesting than Lemma 1.

4.1.1 Proof of Lemma 3

In order to prove this lemma, we will warm-up by proving the following easier lemma. It will be independently useful, and its proof contains all the intuitions needed to ultimately show Lemma 3.

Lemma 4. *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists a (2^{-t})-granular complex sampler of width w that samples $\mathbf{X}$, then there exists an ROBP of width $4w^2$ and input length $\ell = nt$ that samples $\mathbf{X}$.*

This lemma is similar to Lemma 3, except that the ROBP has *larger width* but uses *less randomness*. Since the main parameter we care about when sampling using ROBPs is width, Lemma 3 is usually more useful than Lemma 4. Still, we will see that in some applications, squaring the width is considered a trivial loss in parameters, in which case it is preferable to use as little randomness as possible, thereby favoring

Lemma 4 over Lemma 3. For this reason, we also record the general version of Lemma 4, which can be obtained by combining it with Lemma 2 (set $\alpha := 2^{-t}$ and $t := \lceil \log(nw/\varepsilon) \rceil$.)

Corollary 7. *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists a complex sampler of width w that samples $\mathbf{X}$, then for any $\varepsilon > 0$, there exists an ROBP of width $4w^2$ and input length $\ell = 2n \log(nw/\varepsilon)$ that samples a distribution that is ε-close to $\mathbf{X}$.*

We remark that Corollary 7 is an alternate version of the second bullet in Theorem 6 that uses more width but less randomness.

The plan now is to prove Lemma 4, and then show how we can extend the intuitions developed in this proof to prove Lemma 3. In order to prove Lemma 4, we will transform a complex sampler into an ROBP sampler in a vertex-by-vertex fashion. In particular, we will replace each vertex v in the complex sampler with a small ROBP sampler gadget. The goal of the gadget will roughly be to simulate the edge probabilities coming out of v. The exact gadget that we will use will look something like an (efficient) ROBP that computes a "multi-threshold" function.

To make things more formal, we introduce a slightly new type of ROBP that can be viewed as an intermediate model between single-output ROBPs (Definition 7) and multi-output ROBPs (Definition 8). We will call it a *Σ-ROBP*, and it will compute a function of the form $f : \{0,1\}^n \to \Sigma$ for an arbitrary alphabet Σ. A Σ-ROBP $\mathcal{B}$ has the exact same definition as Definition 7, except instead of having a designated accept vertex $v_{\mathsf{accept}} \in V_n$, each vertex $v \in V_n$ is labeled with an element of Σ. The program $\mathcal{B}$ then computes a function $f_{\mathcal{B}} : \{0,1\}^n \to \Sigma$ in the natural way: on input $x \in \{0,1\}^n$, the program starts at v_{start} and traverses the unique path $P(x)$ whose edges are labeled with input bits $x_1, x_2, \ldots, x_n$, and outputs the label (in Σ) of the final vertex on this path.

We now introduce the formalisms necessary for defining the "multi-threshold" function. Given distinct strings $x, y \in \{0,1\}^n$, recall the definition of the lexicographic order: in this order, it is said that $x < y$ if $x_i < y_i$ at the smallest index $i \in [n]$ where $x_i \neq y_i$. Given this ordering, we define (open and closed) intervals in the natural way: for example, for $x, y \in \{0,1\}^n$, we let $(x, y] := \{s \in \{0,1\}^n : x < s \leq y\}$. It will also be convenient to let $\vec{0} \in \{0,1\}^n$ denote the all zeroes bitstring, $\vec{1} \in \{0,1\}^n$ denote the all ones bitstring, and $\vec{-1}$ denote an imaginary bitstring that is strictly less than all $x \in \{0,1\}^n$. This lets us write $(\vec{-1}, x] = [\vec{0}, x]$ for any $x \in \{0,1\}^n$.

We are now ready to define the multi-threshold function: for any bitstring "thresholds" $\vec{-1} = \tau_0 < \tau_1 < \cdots < \tau_t = \vec{1}$, we define the *$t$-threshold function* $f : \{0,1\}^n \to [t]$ over these thresholds to output the label of the "bucket" into which the input falls. Formally, $f(x)$ is defined as the unique $i \in [t]$ such that $x \in (\tau_{i-1}, \tau_i]$. At last, we are ready to state the key ingredient that goes into proving Lemma 4, Lemma 3, and Theorem 6. That is, we construct a (near-optimal) Σ-ROBP for computing t-threshold functions.

Lemma 5 (Key ingredient for Theorem 6)**.** *For any t-threshold function $f : \{0,1\}^n \to [t]$, there exists a Σ-ROBP of width $2t$ that computes f. Furthermore, this is almost tight: there exist many t-threshold functions that cannot be computed in width $< 2t - 1$.*

The tightness of this result will imply that some constant blow-up in width is necessary when simulating complex samplers with ROBP samplers via this gadget. It is natural to ask whether this gadget can be replaced by a different gadget (computing a different function) that only requires width t. We answer this question in the positive in Section 8, where the different gadget is used to keep our direct product theorem strong. While the different gadget will have very low width, it will only be able to help us approximately sample distributions; it will therefore be useful for the direct product theorem, but less useful for the exact sampling required by Lemmas 3 and 4.

We will prove Lemma 5 at the very end of this section. But first, we show how it can be used to prove Lemmas 3 and 4.

Proof of Lemma 4. We must show that if there is a 2^{-t}-granular complex sampler $\mathcal{B}$ of width w that samples $\mathbf{X} \sim \{0,1\}^n$, then there exists an ROBP of width $4w^2$ and input length $\ell = nt$ that samples $\mathbf{X}$.

The first step is just developing a single gadget. Let $\mathcal{A}$ be a 2^{-t}-granular complex sampler of width w for just one bit $\mathbf{Y} \sim \{0,1\}$. Let its underlying graph be $G = (V, E)$ with layers $V = V_0 \cup V_1$. Label the vertices in the last layer $V_1 = \{v_1, \ldots, v_w\}$. For each $i \in [w]$ and $b \in \{0,1\}$, let $p_{i,b}$ denote the probability that the complex sampler transitions from its start state to v_i and outputs b. Assume without loss of generality that each $p_{i,b} > 0$ (it is straightforward, but notationally inconvenient, to handle when this is not the case). We would like to construct an ROBP sampler $\mathcal{A}'$ of width $4w$ (and length t) that exactly simulates this.

Since $\mathcal{A}$ is 2^{-t}-granular, we know that each $p_{i,b} = P_{i,b} \cdot 2^{-t}$ for some nonnegative integer $P_{i,b}$. Furthermore, since we must have $\sum_{i,b} p_{i,b} = 1$ it must hold that $\sum_{i,b} P_{i,b} = 2^t$. Recalling our discussion of thresholding functions before Lemma 5, it is straightforward to define thresholds in $\{0,1\}^t$

$$\vec{-1} = \tau_\emptyset < \tau_{1,0} < \tau_{2,0} < \cdots < \tau_{w,0} < \tau_{1,1} < \tau_{2,1} < \cdots < \tau_{w,1} = \vec{1}$$

so that for any threshold $\tau_{i,b}$, if we consider the threshold τ' immediately preceding it, then the set $(\tau', \tau_{i,b}] \subseteq \{0,1\}^t$ has exactly $P_{i,b}$ elements.

Now, let $f : \{0,1\}^t \to [2w]$ be the multi-threshold function over the above thresholds. Identify $[2w]$ with the set $[w] \times \{0,1\}$. By definition of our thresholds, note that for any $i \in [w], b \in \{0,1\}$, if we uniformly draw $x \sim \{0,1\}^t$, then the output $f(x) = (i,b)$ with probability $P_{i,b} \cdot 2^{-t} = p_{i,b}$. By Lemma 5, there is an ROBP $\mathcal{C}$ of width $4w$ (and length t) that exactly computes this function. We label the nodes in the last layer of this ROBP with the set $\{u_{i,b}\}_{i \in [w], b \in \{0,1\}}$. We can assume without loss of generality that, upon feeding random bits into this ROBP, the computation path reaches $u_{i,b}$ with probability $p_{i,b}$.

Finally, we can construct $\mathcal{A}'$ from $\mathcal{C}$, as follows. Add a final layer consisting of w nodes, which we will call $\{q_i\}_{i \in [w]}$. Now, for every $i \in [w], b \in \{0,1\}$, draw two edges from $u_{i,b}$ to q_i, with input labels $0, 1$ respectively, but both with the same *output* label b. This completes the construction of our ROBP gadget $\mathcal{A}'$. It is straightforward to verify that for any $i \in [w], b \in \{0,1\}$, it holds that if we feed random bits into $\mathcal{A}'$, then we arrive at q_i and output b with probability $p_{i,b}$. Notice that $\mathcal{A}'$ has length $t+1$. In fact, since the transitions into the last layer are trivial, it is easy to redirect edges from the third-to-last layer to bypass the second-to-last layer and give $\mathcal{A}'$ length t.

The second step is to use the above gadget to transform our 2^{-t}-granular complex sampler $\mathcal{B}$ into an ROBP $\mathcal{B}'$. Let $G = (V, E)$ be the underlying graph of the complex sampler, with layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. For each $i \in [n]$ and $v \in V_{i-1}$, replace its outgoing edges with a new gadget described above. For any $u \in V_i$ and $b \in \{0,1\}$, note that the probability of traversing from v to u and outputting b remains the same, by the gadget construction. Thus we have constructed an ROBP $\mathcal{B}$ that *exactly* samples the same distribution, as desired. Since we construct a fresh gadget (which has width $4w$) for each vertex in each layer, $\mathcal{B}'$ will have width $w \cdot 4w = 4w^2$. And since each gadget has length t, and we are concatenating n gadgets in a series configuration, $\mathcal{B}$ will have length $\ell = nt$. □

Using the ideas in the above proof, we finally turn towards proving Lemma 3.

Proof of Lemma 3. We must show that if there is a 2^{-t}-granular complex sampler $\mathcal{B}$ of width w that samples $\mathbf{X} \sim \{0,1\}^n$, then there is an ROBP of width $7w$ and input length $\ell = 4nwt$ that samples $\mathbf{X}$.

By the proof of Lemma 4, we know that for any 2^{-t}-granular complex sampler $\mathcal{A}$ of width w that samples just 1 bit $\mathbf{Y} \sim \{0,1\}$, there is an ROBP sampler $\mathcal{A}'$ of width $4w$ and length t that exactly samples $\mathbf{Y}$. We call $\mathcal{A}'$ a gadget.

The next step is to use the above gadget to transform the granular complex sampler $\mathcal{B}$ that samples $\mathbf{X} \sim \{0,1\}^n$ into an ROBP $\mathcal{B}'$ that generates the same distribution. Let $G = (V, E)$ be the underlying graph of the complex sampler, with layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. In the proof to Lemma 4, we transformed $\mathcal{B}$ into $\mathcal{B}'$ by looking at each boundary V_{i-1}, V_i, and replacing each vertex $v \in V_{i-1}$ and its outgoing edges (and neighbors) with a gadget $\mathcal{A}'$. However, the gadgets corresponding to each $v \in V_{i-1}$ were stacked on top of each other in a *parallel configuration*, meaning that the width of $\mathcal{B}$ was forced to grow by a factor of w. To prevent this from happening, our goal will be to arrange the gadgets corresponding to each $v \in V_{i-1}$ in a *series configuration*.

In more detail, we will transform $\mathcal{B}$ into $\mathcal{B}'$ via a layer-by-layer process as follows. For each $i \in [n]$, consider the boundary between layers V_{i-1} and V_i. We will replace the edges that cross this boundary with a large ROBP $\mathcal{Z}^*$ that consists of three medium-sized ROBPs $\mathcal{S}, \mathcal{W}, \mathcal{T}$ stacked atop one another. We call $\mathcal{S}$ the "source" or "pre-processing" ROBP, we call $\mathcal{W}$ the "working" or "processing" ROBP, and we call $\mathcal{T}$ the "sink" or "post-processing" ROBP. Intuitively, the source ROBP will take inputs coming from each $v \in V_{i-1}$ and keep them in a "holding pattern." Then, the source ROBP will send these inputs into the working ROBP, which is used to simulate the appropriate probabilities coming out of the edges of each $v \in V_{i-1}$ in the complex sampler. In particular, $\mathcal{W}$ will consist of several gadgets of the form $\mathcal{A}'$ arranged in a series configuration. Finally, the working ROBP will send its inputs to the sink ROBP, which will keep its inputs in a holding pattern before finally passing them off to the proper vertices in V_i.

We will now formalize the above intuition. The plan is to start by formally describing each of the medium-sized ROBPs $\mathcal{S}, \mathcal{W}, \mathcal{T}$.[7] Then, we will describe how to interface these ROBPs together in order to create the large ROBP $\mathcal{Z}^*$. Then, we will describe how to interface $\mathcal{Z}^*$ with the layers V_{i-1}, V_i (i.e., replace the edges crossing between V_{i-1}, V_i with $\mathcal{Z}^*$) and argue that this transformation preserves the desired edge probabilities. For convenience, we will label the vertices in V_{i-1} as $u_1, \ldots, u_w$ and the vertices in V_i as $v_1, \ldots, v_w$.

Construction of the source ROBP $\mathcal{S}$: the directed acyclic graph $G_S = (S, E_S)$ underlying this ROBP will consist of $1 + tw$ layers $S = S_0 \cup S_1 \cup \cdots \cup S_{tw}$, each holding w vertices. Label the vertices in S_i as $s_1^{(i)}, \ldots, s_w^{(i)}$. Then, for every $i \in [tw]$ and $j \in [w]$, draw two edges from $s_j^{(i-1)}$ to $s_j^{(i)}$, one of which is given the input label 0 and the other is given the input label 1. This completes the construction of $\mathcal{S}$. Notice that $\mathcal{S}$ should appear as w parallel lines (of length $1 + tw$) drawn atop one another.

Construction of the sink ROBP $\mathcal{T}$: the directed acyclic graph $G_T = (T, E_T)$ underlying this ROBP will consist of $1 + tw$ layers $T = T_1 \cup T_2 \cup \cdots \cup T_{tw+1}$, each holding $2w$ vertices. Label the vertices in T_i as $\{t_{j,b}^{(i)}\}_{j \in [w], b \in \{0,1\}}$. Then, for every $i \in [tw], j \in [w], b \in \{0,1\}$, draw two edges from $t_{j,b}^{(i)}$ to $t_{j,b}^{(i+1)}$, one of which is given the input label 0 and the other is given the input label 1. This completes the construction of $\mathcal{T}$. Notice that $\mathcal{T}$ should appear as $2w$ parallel lines (of length $1 + tw$) drawn atop one another.

Construction of the working ROBP $\mathcal{W}$: the directed acyclic graph $G_W = (W, E_W)$ underlying this ROBP will consist of tw layers $W = W_1 \cup W_2 \cup \cdots \cup W_{tw}$, each holding $4w$ vertices. We break the construction of $\mathcal{W}$ into the construction of w smaller ROBPs $\mathcal{A}_1, \ldots, \mathcal{A}_w$. Each $\mathcal{A}_i$ will have width $4w$ and length t, and they will be arranged in a series configuration (i.e., consecutively) in order to create $\mathcal{W}$.

Each $\mathcal{A}_i$ will be constructed as follows. First, let us return to thinking about the complex sampler, and for every $j \in [w], b \in \{0,1\}$, let $p_{i,j,b}$ be the probability assigned to the edge (u_i, v_j) with label b by the

[7] These "ROBPs" will actually just be layered DAGs with edge labels, and won't perfectly match the formal definition of ROBP.

complex sampler. Using the proof of Lemma 4, we construct an ROBP $\mathcal{A}_i$ that has width $4w$ and length t (with the last layer having just $2w$ vertices) such that the following holds: if the vertices in the last layer of $\mathcal{A}_i$ are called $\{a_{j,b}\}_{j\in[w],b\in\{0,1\}}$, then vertex $a_{j,b}$ is hit with probability $p_{i,j,b}$ when a random string $x \in \{0,1\}^t$ is fed as input into $\mathcal{A}_i$.

This completes the construction of $\mathcal{W}$. Notice that $\mathcal{W}$ should appear as w gadgets (each of width $4w$ and length t) $\mathcal{A}_1, \ldots, \mathcal{A}_w$ arranged in a series configuration. That is, the start vertex of gadget $\mathcal{A}_i$ will belong to layer $W_{(i-1)t+1}$ and the final layer of $\mathcal{A}_i$ will belong to layer W_{it}.

Combining ROBPs $\mathcal{S}, \mathcal{W}, \mathcal{T}$ into the final ROBP $\mathcal{Z}^*$: the final ROBP $\mathcal{Z}^*$ will be combined by stacking the ROBPs $\mathcal{S}, \mathcal{W}, \mathcal{T}$ one atop another (i.e., arranging them in a parallel configuration). In more detail, the directed acyclic graph $G_Z = (Z, E_Z)$ underlying this ROBP will consist of $tw + 2$ layers $Z = Z_0 \cup Z_1 \cup \cdots \cup Z_{tw+1}$. We will have $Z_0 = S_0, Z_1 = S_1 \cup W_1 \cup T_1, Z_2 = S_2 \cup W_2 \cup T_2, \ldots, Z_{tw} = S_{tw} \cup W_{tw} \cup T_{tw}, Z_{tw+1} = T_{tw+1}$. Thus, $\mathcal{Z}^*$ has width $w + 4w + 2w = 7w$ and length $tw + 2$.

Next, we add and rearrange a few edges. These modifications will focus on connecting the gadgets $\mathcal{A}_i$ in the working ROBP $\mathcal{W}$ to the source ROBP $\mathcal{S}$ and sink ROBP $\mathcal{T}$. In particular, for each $i \in [w]$, consider the gadget $\mathcal{A}_i$. Define $\beta' < \beta'' \in [tw]$ such that $Z_{\beta'}$ holds the start vertex of $\mathcal{A}_i$ and $Z_{\beta''}$ holds the last layer of $\mathcal{A}_i$. Then, consider the i^{th} vertex in layer $S_{\beta'-1}$ of the source ROBP $\mathcal{S}$. Recall it is called $s_i^{(\beta'-1)}$. Furthermore, recall that it had two edges going into the next layer of the source ROBP $\mathcal{S}$. Delete these edges, and replace them with two edges from $s_i^{(\beta'-1)}$ into the start vertex of $\mathcal{A}_i$, and give them input labels 0 and 1, respectively. This completes the connection of the source ROBP $\mathcal{S}$ to the working ROBP $\mathcal{W}$.

Next, recall that the vertices in the final layer of $\mathcal{A}_i$ are labeled $\{a_{j,b}\}_{j\in[w],b\in\{0,1\}}$, and are located in layer $Z_{\beta''}$. Currently, they have no edges leaving them. Now, for each $j \in [w], b \in \{0,1\}$, draw two edges from $a_{j,b}$ to $t_{j,b}^{(\beta''+1)} \in Z_{\beta''+1}$, and give them input labels 0 and 1, respectively. This completes the connection of the working ROBP $\mathcal{W}$ to the sink ROBP $\mathcal{T}$.

We have therefore completed the connection of source ROBP $\mathcal{S}$ to working ROBP $\mathcal{W}$, and the connection from working ROBP $\mathcal{W}$ to sink ROBP $\mathcal{T}$. Thus we have fully completed the construction of ROBP $\mathcal{Z}^*$. The most important property of $\mathcal{Z}^*$ is as follows: for any $i \in [w]$, if we start at vertex $s_i^{(0)}$ and feed a random string $x \in \{0,1\}^{tw+1}$ as input into the ROBP $\mathcal{Z}^*$, then for any $j \in [w], b \in \{0,1\}$, we arrive at vertex $t_{j,b}^{(tw+1)}$ with probability $p_{i,j,b}$. This is straightforward to verify via the above construction and the guaranteed properties of each gadget $\mathcal{A}_i$.

All that remains now is to interface the ROBP $\mathcal{Z}^*$ with the layers V_{i-1}, V_i.

Inserting ROBP $\mathcal{Z}^*$ between layers V_{i-1}, V_i: this is the final and easiest step of our construction. Recall that the vertices in V_{i-1} are labeled as $u_1, \ldots, u_w$ and the vertices in V_i are labeled as $v_1, \ldots, v_w$. For each $i \in [w]$, we do the following: first, delete the edges leaving u_i in the complex sampler. Then, draw two edges from u_i to $s_i^{(0)}$, and give them input labels 0 and 1, respectively. Then, for each $j \in [w], b \in \{0,1\}$, draw two edges from $t_{j,b}^{(tw+1)}$ to v_j, and give them input labels 0 and 1, respectively, and give them both *output label* b. This completes the interfacing of ROBP $\mathcal{Z}^*$ with layers V_{i-1}, V_i.

It is now straightforward to verify that for any $u \in V_{i-1}$ and $v \in V_i$ and $b \in \{0,1\}$, the probability of transitioning from u to v and outputting b is the same in the original complex sampler as it is in the new ROBP sampler. Thus if we replace the boundary between every pair of layers V_{i-1}, V_i with an appropriate ROBP $\mathcal{Z}^*$ as constructed above, we obtain an ROBP $\mathcal{B}'$ that samples the exact same distribution as the original complex sampler, as desired. Furthermore, each $\mathcal{Z}^*$ used in this construction has width $7w$ and length $tw + 2$. Since the overall ROBP $\mathcal{B}'$ will contain a $\mathcal{Z}^*$ between each consecutive layers V_{i-1}, V_i for

$i \in [n]$, the overall ROBP $\mathcal{B}'$ will have length $\ell = n \cdot (tw + 2) + n \leq 4ntw$ and width $7w$, as desired. $\square$

At last, all that remains is to prove our result on computing multi-threshold functions in low width, Lemma 5. We do so below.

Proof of Lemma 5. Let $f : \{0,1\}^n \to [t]$ be a t-threshold function over the thresholds

$$\vec{-1} = \tau^0 < \tau^1 < \cdots < \tau^t = \vec{1},$$

where each $\tau^\alpha \in \{0,1\}^t$ (in particular, the superscripts are labels, not powers). We wish to show that there is an ROBP $\mathcal{B}$ of width $2t$ that computes f. To specify $\mathcal{B}$, we must specify its underlying graph $G = (V, E)$, which will have t layers $V = V_0 \cup V_1 \cup \cdots \cup V_t$. We first specify the construction, and then explain why it works.

For each layer $i \in [t]$, we label the nodes $V_i = \{v_i^1, v_i^2, \ldots, v_i^t, \widetilde{v}_i^1, \widetilde{v}_i^2, \ldots, \widetilde{v}_i^t\}$. The nodes of the form $\widetilde{v}_i^\alpha$ can be thought of as "short-circuit" nodes. In particular, for every short circuit node $\widetilde{v}_i^\alpha$, the both edges leaving it (with input labels $0, 1$ respectively) will simply connect to $\widetilde{v}_{i+1}^\alpha$. The edges leaving the nodes of the form v_i^α will be a little more complex. For each $b \in \{0,1\}$, we draw an edge leaving v_i^α into the next layer, and give the edge the input label b, based on the following logic:

- If $b < \tau_{i+1}^\alpha$, connect the edge to $\widetilde{v}_{i+1}^\alpha$.
- If $b = \tau_{i+1}^\alpha$, connect the edge to v_{i+1}^α.
- If $b > \tau_{i+1}^\alpha$, let $\beta > \alpha$ be the smallest integer such that $(\tau_1^\alpha, \ldots, \tau_i^\alpha, b) \leq (\tau_1^\beta, \ldots, \tau_i^\beta, \tau_{i+1}^\beta)$, and:
  - If the above (rightmost) inequality is strict, connect the edge to $\widetilde{v}_{i+1}^\beta$.
  - Otherwise, connect the edge to v_{i+1}^β.

Now, we just need to specify the edges leaving $v_{\mathsf{start}} \in V_0$. For each $b \in \{0,1\}$, we draw an edge from v_{start} into V_1 with the label b using the same logic as the third bullet above. In particular, let β be the smallest integer such that $b \leq \tau_1^\beta$, and: if this inequality is strict, connect the edge to $\widetilde{v}_1^{(\beta)}$; otherwise, connect the edge to v_1^β. Finally, we give each v_n^α and $\widetilde{v}_n^\alpha$ the output label α.

To see why this ROBP computes the threshold function f, consider its computation path as it reads the input $x = (x_1, \ldots, x_t) \in \{0,1\}^t$. Note that as it reads each input bit x_i, it follows the branching instructions described by our itemized list (think of x_i as b). We make a few observations:

1. Suppose that the branching program reaches node v_i^α after reading $x_1, \ldots, x_i$. Then it must hold that $(x_1, \ldots, x_i) = (\tau_1^\alpha, \ldots, \tau_i^\alpha)$ and $x > \tau^{\alpha-1}$. This follows easily by induction on i.
2. Suppose the branching program reaches short circuit node $\widetilde{v}_i^\alpha$ after reading $x_1, \ldots, x_i$. Then it must hold that $\tau^{\alpha-1} < x < \tau^\alpha$. This is straightforward to show using the above observation.

Thus by combining the above observations, if a string x leads to v_n^α or $\widetilde{v}_n^\alpha$, it must hold that $\tau^{\alpha-1} < x \leq \tau^\alpha$.

Now, consider any $x = (x_1, \ldots, x_n)$ that the ROBP will read. By the definition of our thresholds, there must be some α such that $\tau^{\alpha-1} < x \leq \tau^\alpha$. Of course, x must lead to some final state in the branching program. In order to not contradict the above, this state must be either v_n^α or $\widetilde{v}_n^\alpha$, both of which have the output label α. So the ROBP will output α, and therefore compute the multi-threshold function f, as desired. This completes the proof that any t-threshold function f can be computed by a Σ-ROBP of width $2t$.

We now show that many t-threshold functions $f : \{0,1\}^n \to [t]$ cannot be computed in width $< 2t-1$. In particular, pick any thresholds

$$\vec{-1} = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_t = \vec{1}$$

in $\{0,1\}^n$ such that both of the following hold:

- For every $i \in [t-1]$ the last bit of τ_i is 0.
- For every $i \in [t]$ the interval $(\tau_{i-1}, \tau_i]$ contains at least 3 strings.

We will show that for any such thresholds $\tau_0, \tau_1, \ldots, \tau_t$, the corresponding t-threshold function $f : \{0,1\}^n \to [t]$ cannot be computed in width $< 2t-1$.

To see why, consider any Σ-ROBP $\mathcal{B}$ of width $< 2t-1$. Let $g : \{0,1\}^n \to [t]$ denote the function it computes. We will show that $g \neq f$. First, notice that the lower bound on the size of each $(\tau_{i-1}, \tau_i]$ implies that for every $i \in [t]$ there exists some $\tau_{i-1} < \alpha_i < \tau_i$ such that the last bit of α_i is 0. Next, note that since $\mathcal{B}$ has width $< 2t-1$, there must exist two distinct strings $x < y$ in the sequence

$$\alpha_1 < \tau_1 < \alpha_2 < \tau_2 < \cdots < \tau_{t-1} < \alpha_t$$

that lead to the same state in the second-to-last layer of $\mathcal{B}$. Now, define x^0 to be x with its last bit replaced by 0, and x^1 to be x with its last bit replaced by 1. Similarly, define y^0 to be y with its last bit replaced by 0, and y^1 to be y with its last bit replaced by 1. Since the ROBP is agnostic to which of x, y it read once it reaches the second to last layer, we know $g(x^0) = g(y^0)$ and $g(x^1) = g(y^1)$.

Suppose now that there is no $i \in [t]$ such that x^0, y^0 both belong to the interval $(\tau_{i-1}, \tau_i]$. Then by definition of thresholding functions, we clearly have $f(x^0) \neq f(y^0)$, and thus $g \neq f$. Thus assume that there is some $i \in [t]$ such that x^0, y^0 both belong to $(\tau_{i-1}, \tau_i]$. Note that this is only possible if $x = \alpha_i$ and $y = \tau_i$ for some $i \in [t-1]$. But then $\tau_{i-1} < x^1 \leq \tau_i$ and $y^1 > \tau_i$, meaning that there is no $j \in [t]$ such that x^1, y^1 both belong to the interval $(\tau_{j-1}, \tau_j]$. In other words, $f(x^1) \neq f(y^1)$, and thus $g \neq f$, as desired. □

4.1.2 Proof of Lemma 1

We have now arrived at the final missing piece for Theorem 6. To prove Lemma 1, we must show that complex samplers can simulate ROBP samplers using roughly the same width.

Proof of Lemma 1. We must show that if there is an ROBP $\mathcal{B}$ of width w and length ℓ that samples $\mathbf{X} \sim \{0,1\}^n$, then there is a complex sampler $\mathcal{B}'$ of width $2w$ that samples $\mathbf{X}$.

The first step is transforming $\mathcal{B}$ into an ROBP where each edge is labeled by 0 or 1 output bits. Towards this end, let $G = (V, E)$ be the underlying graph of $\mathcal{B}$, with layer $V = V_0 \cup V_1 \cup \cdots \cup V_\ell$. Fix any $i \in [n]$, and consider the edges between layers V_{i-1} and V_i. They are each labeled by γ_i output bits. If γ_i is already 0 or 1, we do not change anything about layers V_{i-1} and V_i. So henceforth assume $\gamma_i > 1$.

The idea is to simply add γ_{i+1} new layers $L_i^0, L_i^1, \ldots, L_i^{\gamma_i}$ in between V_{i-1} and V_i. For each vertex $v \in V_{i-1}$, we do the following: suppose that v originally had an edge to $u \in V_i$ with input label 0 and output label $s \in \{0,1\}^{\gamma_i}$. Now, delete that edge and simulate it as follows: add a new vertex v^j to each new layer L_i^j. Draw an edge from v to v^0, and give it input label 0 and no output label. Then, draw two new edges from v^0 to v^1, with input labels $0, 1$, and output label s_1. Then, draw two new edges from v^1 to v^2,

with input labels $0, 1$, and output label s_2. Continue this process until we have drawn edges up to vertex v^{γ_i}. Finally, draw two new edges from v^{γ_i} to u with input labels $0, 1$ and an empty output label.

Now suppose that v originally had an edge to $w \in V_i$ with input label 1 and output label $t \in \{0,1\}^{\gamma_i}$. Do the exact same process as before, except give the first new edge that is drawn the input label 1 (instead of 0). Finally, recall that we needed to do this for every $v \in V_{i-1}$. After this is done, repeat the process for any V_{i-1}, V_i with $\gamma_i > 1$. Now, it is straightforward to verify that for any $v \in V_{i-1}, u \in V_i$, if we plug random bits into our new branching program and arrive at v, then the probability of then transitioning from v to u and outputting any given string s of bits will be the same as it was before. Thus the new ROBP samples the same distribution $\mathbf{X}$ as before, has width $2w$, and every edge is labeled with at most 1 output bit.

The second step is to transform this new ROBP into a complex sampler for $\mathbf{X}$. Let $\mathcal{B}$ now denote the new ROBP we have sampling $\mathbf{X}$, which has each edge labeled with at most 1 output bit. Let its underlying graph $G = (V, E)$ have layers $V = V_0 \cup V_1 \cup \cdots \cup V_\ell$, where we are now guaranteed $\ell \geq n$. Define a collection of indices $0 = a_0 < a_1 < \cdots < a_n$ as follows: let a_j be the smallest integer greater than a_{j-1} such that V_{a_j} has incoming edges labeled with 1 output bit. To translate our ROBP sampler into a complex sampler for $\mathbf{X}$, we will do a transformation for each pair of layers $V_{a_{j-1}}, V_{a_j}$.

Fix some $v \in V_{a_{j-1}}, u \in V_{a_j}$. Notice that all paths between v, u are labeled with exactly 1 output bit. For every $b \in \{0,1\}$, compute the following probability, $p_{v,u,b}$: plug random bits into the ROBP, condition on reaching v, then let $p_{v,u,b}$ be the probability of traversing from v to u and outputting b. Given this probability, draw a *complex sampler* edge from v to u, and give it label b and probability $p_{v,u,b}$. Now do the same for every $v \in V_{a_{j-1}}, u \in V_{a_j}$. Then, repeat this process for all $j \in [n]$.

At the end of the above process, delete all edges that are not complex sampler edges, and delete all vertices that are not in $V_{a_1}, V_{a_2}, \ldots, V_{a_n}$. Thus we obtain a complex sampler of width $2w$. Furthermore, for any $v \in V_{a_{j-1}}, u \in V_{a_j}$, it is straightforward to verify that the probability of transitioning from v to u and outputting any single bit b (conditioned on reaching v in the first place) is the same in the original ROBP and the new complex sampler. Thus we have a complex sampler $\mathcal{B}'$ of width $2w$ that samples $\mathbf{X}$. □

4.2 An equivalence theorem between simple samplers and ROBPs for input-output pairs

In this section, we will show that a simple sampler can generate the distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ if and only if an ROBP can compute the function b. More formally, we have the following theorem.

Theorem 7. *For any function $b : \{0,1\}^n \to \{0,1\}$, there exists a simple sampler of width w that samples $(\mathbf{U}_n, b(\mathbf{U}_n))$ if and only if there exists an ROBP of width w that computes b.*

Proof. We start by proving that a simple sampler implies an ROBP. Let $\mathcal{B}$ be the simple sampler of width w that can sample $(\mathbf{U}_n, b(\mathbf{U}_n))$, with underlying graph $G = (V, E)$ and layers $V = V_0 \cup V_1 \cup \cdots \cup V_{n+1}$. We now define an ROBP $\mathcal{B}'$ that computes b as follows. Define its underlying graph $G' = (V', E')$ to have layers $V'_0 \cup V'_1 \cup \cdots \cup V'_n$, where each V'_i is an exact copy of V_i. Furthermore, for each $e \in E$ that goes between layers V_{i-1}, V_i for some $i \in [n]$, and which is not assigned probability 0, copy this edge (with its label) into E'. Finally, for all $v' \in V'_n$ whose corresponding vertex in $v \in V_n$ has an outgoing edge labeled 1, label the vertex v' as an *accepting state*.[8]

Now, notice that for every $x \in \{0,1\}^n$ with $b(x) = 1$, the simple sampler $\mathcal{B}$ must output $(x, 1)$ with nonzero probability, and so x must lead to an accepting state in $\mathcal{B}'$. And for every $x \in \{0,1\}^n$ with

[8]Technically, the definition of ROBP requires a single vertex in V_n to be labeled v_{accept}, but we can easily adjust for this by selecting one of the accepting states to be designated v_{accept}, and redirecting all edges that go into an accepting state to go into v_{accept}, instead.

$b(x) = 0$, it must hold that x does *not* lead to an accepting state in $\mathcal{B}'$, because otherwise this would imply that $\mathcal{B}$ samples $(x, 1) = (x, \neg b(x))$ with nonzero probability (meaning that it does not sample $(\mathbf{U}_n, b(\mathbf{U}_n))$ exactly). Thus $\mathcal{B}'$ computes b, and has width w.

We now prove the reverse direction. Let $\mathcal{B}$ be an ROBP of width w that computes b, with underlying graph $G = (V, E)$ and layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. We define a simple sampler $\mathcal{B}'$ that samples $(\mathbf{U}_n, b(\mathbf{U}_n))$ as follows. Define its underlying graph $G' = (V', E')$ to have layers $V'_0 \cup V'_1 \cup \cdots \cup V'_n \cup V'_{n+1}$, where each V'_i, for $i \in [n]$, is an exact copy of V_i. Furthermore, copy the entire edge set of E into E' (including its labels).

Let V'_{n+1} consist of a single vertex, which we call v^*. For each $v' \in V'_n$, check if the corresponding vertex $v \in V_n$ is labeled v_{accept}: if so, draw an edge from $v' \in V'_n$ to $v^* \in V'_{n+1}$ and label it 1; otherwise, draw the same edge but label it 0. Finally, for each vertex $v' \in V'$, let $p_{v'}$ be the uniform probability distribution over its outgoing edges.

It is straightforward to verify that for any fixed $x \in \{0,1\}^n$, the first n bits produced by the simple sampler $\mathcal{B}'$ are exactly x with probability 2^{-n}, and if this is true then the final bit produced by $\mathcal{B}'$ is $b(x)$ with probability 1. Thus $\mathcal{B}'$ exactly samples $(\mathbf{U}_n, b(\mathbf{U}_n)$ and has width w. □

4.2.1 An extension to correlation bounds

Theorem 7 provides a way to convert worst-case lower bounds against sampling $(\mathbf{U}_n, b(\mathbf{U}_n))$ into worst-case lower bounds against computing b. In this section, we strengthen this direction and provide a way to convert average-case sampling lower bounds into average-case computing lower bounds.

Theorem 8. *Fix any function $b : \{0,1\}^n \to \{0,1\}$, and suppose that for any simple sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width w, it holds that $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| > \frac{1-\varepsilon}{2}$. Then for any ROBP $f : \{0,1\}^n \to \{0,1\}$ of width w, it holds that $|\operatorname{corr}(f, b)| < \varepsilon$.*

Proof. We show the contrapositive: that if there exists an ROBP $\mathcal{B}$ of width w computing a function $f : \{0,1\}^n \to \{0,1\}$ with $|\operatorname{corr}(f, b)| \geq \varepsilon$, then there is a simple sampler $\mathcal{B}'$ of width w sampling a distribution $\mathbf{X} \sim \{0,1\}^n$ such that $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| \leq \frac{1-\varepsilon}{2}$. We start by assuming, without loss of generality, that $\operatorname{corr}(f, b) \geq \varepsilon$. To see why, simply note that $\operatorname{corr}(f, b) = -\operatorname{corr}(1 - f, b)$ and that if f is computable by a width w ROBP then so is $1 - f$ (by swapping the accept and reject states). Thus if we started with $\operatorname{corr}(f, b) \leq -\varepsilon$, we could instead consider the ROBP computing $f' := 1 - f$ which has $\operatorname{corr}(f', b) \geq \varepsilon$.

So we now assume $\mathcal{B}$ is a width w ROBP computing a function f with $\operatorname{corr}(f, b) \geq \varepsilon$. By Definition 3, $\operatorname{corr}(f, b) = \Pr[f = b] - \Pr[f \neq b] = 2\Pr[f = b] - 1 \geq \varepsilon$, which implies that $\Pr_{x \sim \mathbf{U}_n}[f(x) = b(x)] \geq \frac{1+\varepsilon}{2}$. We will use this in a moment.

Now, let $\mathcal{B}'$ be a simple sampler that is constructed from the ROBP $\mathcal{B}$ in the exact same way as in the proof to Theorem 7. It is easy to verify that the first n bits produced by $\mathcal{B}'$ are equal to any given $x \in \{0,1\}^n$ with probability 2^{-n}, and if this is true then the final bit produced by $\mathcal{B}'$ is $f(x)$ with probability 1. Thus if

$\mathbf{X} \sim \{0,1\}^{n+1}$ is the distribution produced by $\mathcal{B}'$, we have

$$\begin{aligned}
|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| &= \frac{1}{2} \sum_{x \in \{0,1\}^n, y \in \{0,1\}} |\Pr[\mathbf{X} = (x,y)] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) = (x,y)]| \\
&= \frac{1}{2} \sum_x (|\Pr[\mathbf{X} = (x, b(x))] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) = (x, b(x))]| + \\
&\qquad |\Pr[\mathbf{X} = (x, \neg b(x))] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) = (x, \neg b(x))]|) \\
&= \frac{1}{2} \left(\sum_{x: f(x) = b(x)} (|2^{-n} - 2^{-n}| + |0 - 0|) + \sum_{x: f(x) \neq b(x)} (|0 - 2^{-n}| + |2^{-n} - 0|) \right) \\
&= \frac{1}{2} \sum_{x: f(x) \neq b(x)} 2 \cdot 2^{-n} \\
&= \Pr[f(x) \neq b(x)] = 1 - \Pr[f(x) = b(x)] \\
&\leq 1 - \frac{1+\varepsilon}{2} = \frac{1-\varepsilon}{2}.
\end{aligned}$$

Thus $\mathcal{B}'$ achieves the claimed sampling bound, and has width w. □

4.2.2 An extension to unknown-order ROBPs

Finally, we prove an unknown-order version of Theorem 8, and provide a way to obtain lower bounds against unknown-order ROBPs from lower bounds against unknown-order samplers (see Section 3.3 for definitions).

Theorem 9. *Fix any function $b : \{0,1\}^n \to \{0,1\}$, and suppose that for any unknown-order simple sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width w, it holds that $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| > \frac{1-\varepsilon}{2}$. Then for any unknown-order ROBP $f : \{0,1\}^n \to \{0,1\}$ of width w, it holds that $|\operatorname{corr}(f,b)| < \varepsilon$.*

Proof. We prove the contrapositive. Let $f : \{0,1\}^n \to \{0,1\}$ be an unknown-order ROBP of width w such that $|\operatorname{corr}(f,b)| \geq \varepsilon$. By definition of unknown-order ROBP, there exists an ROBP $f' : \{0,1\}^n \to \{0,1\}$ of width w and a permutation $\pi : [n] \to [n]$ such that $f(x) = f'(x^\pi)$. Now, let π^{-1} denote the inverse of π, let $b' : \{0,1\}^n \to \{0,1\}$ be defined as $b'(x) := b(x^{\pi^{-1}})$, and observe that

$$\operatorname{corr}(f,b) = \mathbb{E}_x[(-1)^{f'(x^\pi)}(-1)^{b(x)}] = \mathbb{E}_x[(-1)^{f'(x)}(-1)^{b(x^{\pi^{-1}})}] = \operatorname{corr}(f', b').$$

Thus since $|\operatorname{corr}(f,b)| \geq \varepsilon$, we also know that $|\operatorname{corr}(f',b')| \geq \varepsilon$. Thus we have an ROBP $f' : \{0,1\}^n \to \{0,1\}$ of width w such that $\operatorname{corr}(f',b') \geq \varepsilon$, so by Theorem 8 we know there exists a simple sampler $\mathbf{Y} \sim \{0,1\}^{n+1}$ of width w such that

$$|\mathbf{Y} - (\mathbf{U}_n, b'(\mathbf{U}_n))| \leq \frac{1-\varepsilon}{2}.$$

Observe that $(\mathbf{U}_n, b'(\mathbf{U}_n)) = (\mathbf{U}_n, b(\mathbf{U}_n^{\pi^{-1}})) = (\mathbf{U}_n^\pi, b(\mathbf{U}_n))$, and thus

$$|\mathbf{Y} - (\mathbf{U}_n^\pi, b(\mathbf{U}_n))| \leq \frac{1-\varepsilon}{2}.$$

Consider now a permutation $\pi^* : [n+1] \to [n+1]$ such that $\pi^*(n+1) = n+1$ and π^* restricted to $[n]$ is exactly π^{-1}. By the data-processing inequality (Fact 1), we have

$$|\mathbf{Y}^{\pi^*} - (\mathbf{U}_n, b(\mathbf{U}_n))| = |\mathbf{Y}^{\pi^*} - (\mathbf{U}_n^{\pi}, b(\mathbf{U}_n))^{\pi^*}| \leq |\mathbf{Y} - (\mathbf{U}_n^{\pi}, b(\mathbf{U}_n))| \leq \frac{1-\varepsilon}{2}.$$

Since $\mathbf{Y} \sim \{0,1\}^{n+1}$ is a simple sampler of width w, we have found an unknown-order simple sampler $\mathbf{X} := \mathbf{Y}^{\pi^*}$ of width w such that $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| \leq \frac{1-\varepsilon}{2}$, as desired. □

4.3 An equivalence theorem between simple samplers and ROBPs for flat distributions

In this section, we will show another equivalence between simple samplers and ROBPs for computation. This time, we will consider distributions $\mathbf{Q} \sim \{0,1\}^n$ that are uniform over some subset $S \subseteq \{0,1\}^n$. We let $1_S : \{0,1\}^n \to \{0,1\}$ denote the indicator function for S, and prove the following theorem.

Theorem 10. *For any distribution $\mathbf{Q} \sim \{0,1\}^n$ that is uniform over some subset $S \subseteq \{0,1\}^n$:*

- *If there exists a simple sampler of width w that samples $\mathbf{Q}$, then there exists an ROBP of width $w+1$ that computes $1_S : \{0,1\}^n \to \{0,1\}$.*
- *If there exists an ROBP of width w that computes $1_S : \{0,1\}^n \to \{0,1\}$, then there exists a simple sampler of width w that samples $\mathbf{Q}$.*

Proof. We start by proving that a simple sampler implies an ROBP. Let $\mathcal{B}$ be the simple sampler of width w that can sample $\mathbf{Q}$, with underlying graph $G = (V, E)$ and layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. We now define an ROBP $\mathcal{B}'$ that computes 1_S as follows. Define its underlying graph $G' = (V', E')$ to have layers $V_0' \cup V_1' \cup \cdots \cup V_n'$, where each V_i' is an exact copy of V_i. Furthermore, copy each edge $e \in E$, which is not assigned probability 0, into E' (with its label).

Now, for each $i \in [n]$, we add an additional vertex to V_i', which we call $\widetilde{v}_i$. And for every $i \in [n-1]$, draw two edges from $\widetilde{v}_i$ to $\widetilde{v}_{i+1}$: one labeled with 0 and the other labeled 1. Intuitively, this new row of vertices might be considered as the "reject gutter." Next, for each $i \in [n]$ and $v' \in V_{i-1}'$, if v' has no outgoing edge labeled 0, draw an edge from v' to $\widetilde{v}_i$ and label it 0. And if v' has no outgoing edge labeled 1, draw an edge from v' to $\widetilde{v}_i$ and label it 1. Finally, for all $v' \in V_n'$ *except* $\widetilde{v}_n$, label v' as an accepting state.

Now, notice that for every $x \in \{0,1\}^n$ such that $1_S(x) = 1$, it holds by definition that $x \in \text{support}(\mathbf{Q})$, which means that the simple sampler $\mathcal{B}$ of course outputs x with nonzero probability. Thus, x must lead to an accepting state in $\mathcal{B}'$. And for every $x \in \{0,1\}^n$ such that $1_S(x) = 0$, it holds by definition that $x \notin \text{support}(\mathbf{Q})$, which means that the simple sampler $\mathcal{B}$ of course outputs x with zero probability. This means that the unique path in $\mathcal{B}$ labeled with x must have some edge assigned zero probability, which means that x must enter the "reject gutter" at some point in $\mathcal{B}'$, and ultimately arrive at $\widetilde{v}_n$, the only reject state in V_n. Thus $\mathcal{B}'$ computes 1_S, and has width $w+1$.

We now prove the reverse direction. Let $\mathcal{B}$ be an ROBP of width w that computes 1_S, with underlying graph $G = (V, E)$ and layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. We define a simple sampler $\mathcal{B}'$ that samples $\mathbf{Q}$ as follows. First, set its underlying graph to be identical to $G = (V, E)$.

Now, for each $v' \in V'$, let $P(v')$ count the number of paths from v' to an accept state. In particular, $P(v') = 0$ for vertices in the last layer that are not accept states. Then, we assign edge transition probabilities as follows. For any $u' \in V'$ with $P(u') = 0$, assign any arbitrary distribution over its outgoing edges (it won't matter). For all other edges (u', v'), assign it probability $P(v')/P(u')$. Note that this is indeed a

valid probability distribution: suppose u' has outgoing edges to v' and w': then it will always be true that $P(u') = P(v') + P(w')$, and thus the probabilities we assigned over its outgoing edges, namely $P(v')/P(u')$ and $P(w')/P(u')$, must add up to 1.

Suppose now that x is not accepted by $\mathcal{B}$. Then the last vertex v' on its computation path in $\mathcal{B}$ will not be an accept state. Thus $P(v') = 0$, which means that the probability on the last edge before hitting v' is 0, and thus the overall probability assigned to this path in $\mathcal{B}'$ is 0.

Suppose now that x *is* accepted by $\mathcal{B}$, and that its computation path uses edges $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \ldots, e_n = (v_1, v_n)$. Then of course $P(v_i) > 0$ for each vertex on this path, and the overall probability of the path is

$$\frac{P(v_1)}{P(v_0)} \cdot \frac{P(v_2)}{P(v_1)} \cdots \frac{P(v_n)}{P(v_{n-1})} = \frac{P(v_n)}{P(v_0)}.$$

But $P(v_n)$ is just 1, and v_0 must be the start vertex of the program, so $P(v_0)$ must be exactly the number of strings accepted by $\mathcal{B}$. Thus our simple sampler $\mathcal{B}'$ samples each accepting string of $\mathcal{B}$ with the same probability $1/P(v_0)$. In other words, the distribution it outputs is uniform over $1_S^{-1}(1) = S$, meaning that it exactly samples $\mathbf{Q}$. □

4.3.1 An extension to covariance bounds

Theorem 10 shows how to convert worst-case lower bounds against sampling a flat distribution $\mathbf{Q} \sim \{0,1\}^n$ into worst-case lower bounds against computing the indicator function 1_S of its support. Here, we strengthen this direction and show how to convert average-case sampling lower bounds into covariance bounds. Note that we must use the more general notion of covariance (instead of correlation, as in Theorem 8) since it is possible to get strong sampling lower bounds against $\mathbf{Q}$ even if 1_S is very biased.

Theorem 11. *Let $\mathbf{Q} \sim \{0,1\}^n$ be any distribution that is uniform over some subset $S \subseteq \{0,1\}^n$, and suppose that for any simple sampler $\mathbf{X} \sim \{0,1\}^n$ of width w, it holds that $|\mathbf{X} - \mathbf{Q}| > 1 - \frac{\varepsilon}{4}$. Then for any ROBP $f : \{0,1\}^n \to \{0,1\}$ of width w, it holds that $|\operatorname{cov}(f, 1_S)| < \varepsilon$.*

Proof. We show the contrapositive: that if there exists an ROBP $\mathcal{B}$ of width w computing $f : \{0,1\}^n \to \{0,1\}$ such that $|\operatorname{cov}(f, 1_S)| \geq \varepsilon$, then there is a simple sampler $\mathcal{B}'$ of width w sampling a distribution $\mathbf{X} \sim \{0,1\}^n$ such that $|\mathbf{X} - \mathbf{Q}| \leq 1 - \varepsilon/4$. We start by assuming, without loss of generality, that $\operatorname{cov}(f, 1_S) \geq \varepsilon$. We can do this because $\operatorname{cov}(f, 1_S) = -\operatorname{cov}(1 - f, 1_S)$, and because $1 - f$ must also be computable by an ROBP of width w (by swapping the accept and reject states).

So now we assume $\mathcal{B}$ is a width w ROBP computing a function f with $\operatorname{cov}(f, 1_S) \geq \varepsilon$. Let $\mathcal{B}'$ be a simple sampler that is constructed from the ROBP $\mathcal{B}$ in the exact same way as in the proof to Theorem 10. We know it outputs a distribution $\mathbf{X} \sim \{0,1\}^n$ that is uniform over $f^{-1}(1)$. We want to show that $|\mathbf{X} - \mathbf{Q}| \leq 1 - \varepsilon/4$.

It will now be notationally convenient to define the following quantities, taking uniform $x \sim \{0,1\}^n$:

$$\begin{aligned} a &:= \Pr[f(x) = 1] \\ b &:= \Pr[1_S(x) = 1] \\ c &:= \Pr[f(x) = 1 \text{ and } 1_S(x) = 1]. \end{aligned}$$

Without loss of generality, we may assume both $a, b > 0$. Now, by definition of covariance, we have $\operatorname{cov}(f, 1_S) = \operatorname{corr}(f, 1_S) - \operatorname{bias}(f)\operatorname{bias}(1_S)$, and using the definitions of correlation and bias, it is a straightforward calculation to obtain

$$\operatorname{cov}(f, 1_S) = 4c - 4ab.$$

Now, notice that for any $x \in \text{support}(\mathbf{X})$, it holds that $\Pr[\mathbf{X} = x] = 2^{-n}/a$. Similarly, for any $q \in \text{support}(\mathbf{Q})$, it holds that $\Pr[\mathbf{Q} = q] = 2^{-n}/b$. So using the (half L_1-norm) definition of statistical distance, it is straightforward to compute:

$$\begin{aligned} 2 \cdot |\mathbf{X} - \mathbf{Q}| &= \frac{1}{a} \cdot \Pr_x[f(x) = 1, 1_S(x) = 0] + \frac{1}{b} \cdot \Pr_x[f(x) = 0, 1_S(x) = 1] \\ &\quad + |\frac{1}{a} - \frac{1}{b}| \cdot \Pr[f(x) = 1, 1_S(x) = 1] \\ &= (1/a)(a - c) + (1/b)(b - c) + |1/a - 1/b| \cdot c. \end{aligned}$$

Without loss of generality assume $1/a \geq 1/b$, and notice this quantity is $2 - 2c/b$. Thus we have:

$$\begin{aligned} |\mathbf{X} - \mathbf{Q}| &= 1 - c/b, \\ \text{cov}(f, 1_S) &= 4(c - ab). \end{aligned}$$

Notice we have $c/b \geq c \geq c - ab$. Thus $c/b \geq \text{cov}(f, 1_S)/4$, and thus

$$|\mathbf{X} - \mathbf{Q}| = 1 - c/b \leq 1 - \text{cov}(f, 1_S)/4 \leq 1 - \varepsilon/4,$$

as desired. □

4.3.2 An extension to unknown-order ROBPs

Finally, we prove an unknown-order version of Theorem 11, and provide a way to obtain lower bounds against unknown-order ROBPs from lower bounds against unknown-order samplers (see Section 3.3 for definitions).

Theorem 12. *Let $\mathbf{Q} \sim \{0,1\}^n$ be any distribution that is uniform over some subset $S \subseteq \{0,1\}^n$, and suppose that for any unknown-order simple sampler $\mathbf{X} \sim \{0,1\}^n$ of width w, it holds that $|\mathbf{X} - \mathbf{Q}| > 1 - \frac{\varepsilon}{4}$. Then for any unknown-order ROBP $f : \{0,1\}^n \to \{0,1\}$ of width w, it holds that $|\text{cov}(f, 1_S)| < \varepsilon$.*

Proof. We prove the contrapositive. Let $f : \{0,1\}^n \to \{0,1\}$ be an unknown-order ROBP of width w such that $|\text{cov}(f, 1_S)| \geq \varepsilon$. By definition of unknown-order ROBP, there exists an ROBP $f' : \{0,1\}^n \to \{0,1\}$ of width w and a permutation $\pi : [n] \to [n]$ such that $f(x) = f'(x^\pi)$. Now, let π^{-1} denote the inverse of π, and let $1'_S : \{0,1\}^n \to \{0,1\}$ be defined as $1'_S(x) := 1_S(x^{\pi^{-1}})$.

As we saw in the proof to Theorem 9, $\text{corr}(f, 1_S) = \text{corr}(f', 1'_S)$. Furthermore, observe that permuting a function's input does not change its bias (Definition 2). Thus we have:

$$\text{cov}(f, 1_S) = \text{corr}(f, 1_S) - \text{bias}(f)\,\text{bias}(1_S) = \text{corr}(f', 1'_S) - \text{bias}(f')\,\text{bias}(1'_S) = \text{cov}(f', 1'_S).$$

Thus since $|\text{cov}(f, 1_S)| \geq \varepsilon$, we also know that $|\text{cov}(f', 1'_S)| \geq \varepsilon$. Now, define $T := \{x : x^{\pi^{-1}} \in S\}$, and notice that $1'_S = 1_T$. Thus we have an ROBP $f' : \{0,1\}^n \to \{0,1\}$ of width w such that $|\text{cov}(f', 1_T)| \geq \varepsilon$. Then if we define $\widetilde{\mathbf{Q}}$ being uniform over T, we know via Theorem 11 that there exists a simple sampler $\mathbf{Y} \sim \{0,1\}^{n+1}$ of width w such that

$$|\mathbf{Y} - \widetilde{\mathbf{Q}}| \leq 1 - \frac{\varepsilon}{4}.$$

Observe that $\widetilde{\mathbf{Q}}^{\pi^{-1}} = \mathbf{Q}$, and thus by the data-processing inequality (Fact 1) we have

$$|\mathbf{Y}^{\pi^{-1}} - \mathbf{Q}| = |\mathbf{Y}^{\pi^{-1}} - \widetilde{\mathbf{Q}}^{\pi^{-1}}| \leq |\mathbf{Y} - \widetilde{\mathbf{Q}}| \leq 1 - \frac{\varepsilon}{4}.$$

Since $\mathbf{Y} \sim \{0,1\}^n$ is a simple sampler of width w, we have found an unknown-order simple sampler $\mathbf{X} := \mathbf{Y}^{\pi^{-1}}$ of width w such that $|\mathbf{X} - \mathbf{Q}| \leq 1 - \frac{\varepsilon}{4}$, as desired. □

5 A complexity separation

In this section, we prove our main result separating the complexity of sampling with simple samplers from the complexity of sampling with complex samplers. In particular, we prove the following.

Theorem 13. *For any fixed $\varepsilon > 0$, there exist constants $C, c > 0$ such that the following holds. There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any simple sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width $w \leq 2^{cn}$,*

$$|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| > \frac{1}{4} - \varepsilon,$$

but there is a complex sampler $\mathbf{X}^* \sim \{0,1\}^{n+1}$ of width $w^* \leq Cn$ that exactly samples $(\mathbf{U}_n, b(\mathbf{U}_n))$.

We start by defining the function that will be used for b.

Definition 13. *Let $f : \{0,1\}^k \to \{0,1\}^{\hat{k}}$ be any function such that $\hat{k}$ is a power of 2. We let $n := k + \log \hat{k}$, and define the* address function over f *as the function* $\mathsf{address}_f : \{0,1\}^n \to \{0,1\}$ *such that*

$$\mathsf{address}_f(x, \alpha) := f(x)_\alpha,$$

where the input to the function is parsed as $x \in \{0,1\}^k, \alpha \in [\hat{k}]$.

To prove Theorem 13, the plan will be to use this address function, instantiated with an encoder f for a great explicit (linear) list decodable code. Towards this end, we show that simple samplers have a hard time sampling such functions.

Lemma 6. *Let $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^{\hat{k}}$ be the encoding function of a (ρ, L) list decodable code, where $\hat{k}$ is a power of 2. We write $n := k + \log \hat{k}$, and let $b : \{0,1\}^n \to \{0,1\}$ be the address function over Enc from Definition 13. Then for any simple sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width w,*

$$|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| > \frac{1}{2} \cdot \rho \cdot (1 - wL \cdot 2^{-k}).$$

Proof. We recall that b computes the function $b(x, \alpha) = \mathsf{Enc}(x)_\alpha$, and start by defining a *bad* set of inputs on which $\mathbf{X}$ is more likely to fail at "computing" b than succeed at at "computing" b. More formally, for every $x \in \{0,1\}^k$ we define the set

$$\mathsf{Bad\_Indices}(x) := \{\alpha \in [\hat{k}] : \Pr[\mathbf{X} = (x, \alpha, \neg b(x, \alpha))] \geq \Pr[\mathbf{X} = (x, \alpha, b(x, \alpha))]\},$$

and for some threshold t to be set later, we define

$$\mathsf{Bad\_Messages} := \{x \in \{0,1\}^k : |\mathsf{Bad\_Indices}(x)| > t\}.$$

The plan now is to (i) lower bound $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))|$ using the size of $\mathsf{Bad\_Messages}$, and then (ii) lower bound the size of $\mathsf{Bad\_Messages}$ using the list decodability of Enc and the bounded width of our simple

sampler $\mathbf{X}$. We proceed with (i), setting $\mathbf{Y} := (\mathbf{U}_n, b(\mathbf{U}_n))$ for notational convenience.

$$\begin{aligned}
2 \cdot |\mathbf{X} - \mathbf{Y}| &= \sum_{x \in \{0,1\}^k, \alpha \in [\hat{k}], c \in \{0,1\}} |\Pr[\mathbf{X} = (x, \alpha, c)] - \Pr[\mathbf{Y} = (x, \alpha, c)]| \\
&\geq \sum_{\substack{x \in \mathsf{Bad_Messages} \\ \alpha \in \mathsf{Bad_Indices}(x) \\ c \in \{0,1\}}} |\Pr[\mathbf{X} = (x, \alpha, c)] - \Pr[\mathbf{Y} = (x, \alpha, c)]| \\
&= \sum_{\substack{x \in \mathsf{Bad_Messages} \\ \alpha \in \mathsf{Bad_Indices}(x)}} \big(|\Pr[\mathbf{X} = (x, \alpha, b(x,\alpha))] - \Pr[\mathbf{Y} = (x, \alpha, b(x,\alpha))]| \\
&\qquad + |\Pr[\mathbf{X} = (x, \alpha, \neg b(x,\alpha))] - \Pr[\mathbf{Y} = (x, \alpha, \neg b(x,\alpha))]| \big) \\
&= \sum_{x,\alpha} \big(|\Pr[\mathbf{X} = (x, \alpha, b(x,\alpha))] - \Pr[\mathbf{Y} = (x, \alpha, b(x,\alpha))]| + \Pr[\mathbf{X} = (x, \alpha, \neg b(x,\alpha))] \big) \\
&\geq \sum_{x,\alpha} \big(|\Pr[\mathbf{X} = (x, \alpha, b(x,\alpha))] - \Pr[\mathbf{Y} = (x, \alpha, b(x,\alpha))]| + \Pr[\mathbf{X} = (x, \alpha, b(x,\alpha))] \big) \\
&\geq \sum_{x,\alpha} \Pr[\mathbf{Y} = (x, \alpha, b(x,\alpha))],
\end{aligned}$$

where the second to last inequality follows by definition of $\mathsf{Bad\_Indices}(x)$, and the last inequality follows from the fact that $|x - y| + x \geq y$ for any $x, y \in \mathbb{R}$. Continuing, we have

$$\begin{aligned}
\sum_{\substack{x \in \mathsf{Bad_Messages} \\ \alpha \in \mathsf{Bad_Indices}(x)}} \Pr[\mathbf{Y} = (x, \alpha, b(x,\alpha))] &= \sum_{\substack{x \in \mathsf{Bad_Messages} \\ \alpha \in \mathsf{Bad_Indices}(x)}} \frac{1}{2^k} \cdot \frac{1}{\hat{k}} \\
&> |\mathsf{Bad_Messages}| \cdot t \cdot \frac{1}{2^k} \cdot \frac{1}{\hat{k}},
\end{aligned}$$

since each bad message has $> t$ bad indices, by definition. Thus, recalling that $\mathbf{Y} := (\mathbf{U}_n, b(\mathbf{U}_n)))$, we have

$$|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| > \frac{1}{2} \cdot |\mathsf{Bad\_Messages}| \cdot t \cdot \frac{1}{2^k} \cdot \frac{1}{\hat{k}}. \tag{3}$$

Our goal now is to get a lower bound on $|\mathsf{Bad\_Messages}|$, using the list decodability of Enc and the bounded width w of our simple sampler $\mathbf{X} \sim \{0,1\}^{n+1}$. Towards this end, let $G = (V, E)$ be the underlying graph of our simple sampler, with layers $V = V_0 \cup V_1 \cup \cdots \cup V_{n+1}$. Consider layer V_k, and focus on some vertex $v \in V_k$ in this layer. Consider taking a random walk from v to V_{n+1} in the simple sampler according to its edge probabilities, outputting the edge labels seen. Let $\mathbf{W} \in \{0,1\}^{\log \hat{k}+1}$ denote these edge labels.

Now, for each $\alpha \in [\hat{k}]$, let $y^*_\alpha \in \{0,1\}$ denote the bit more likely to be seen at the end of this random walk, if this random walk first outputs α (recall that $[\hat{k}]$ is identified with $\{0,1\}^{\log \hat{k}}$). More formally, we define

$$y^*_\alpha := \begin{cases} 0 & \text{if } \Pr[\mathbf{W} = (\alpha, 0)] \geq \Pr[\mathbf{W} = (\alpha, 1)], \\ 1 & \text{if } \Pr[\mathbf{W} = (\alpha, 1)] > \Pr[\mathbf{W} = (\alpha, 0)]. \end{cases}$$

Now, define $y^* := (y_1^*, y_2^*, \ldots, y_{\hat{k}}^*) \in \{0,1\}^{\hat{k}}$. Intuitively, y^* is the codeword that is "most consistent" with the remainder of the random walk from v to V_{n+1}; that is, it is the codeword $y^* \in \{0,1\}^{\hat{k}}$ for which the simple sampler does the best job at sampling (α, y_α^*) on the remainder of the walk from v to V_{n+1}. We will now use the list decodability of our code to argue that most messages $x \in \{0,1\}^k$ will encode to a codeword $y = \mathsf{Enc}(x) \in \{0,1\}^{\hat{k}}$ that looks quite different from y^*, meaning that the simple sampler will do a bad job at sampling (α, y_α) on the walk from v to V_{n+1}.

More formally, recall that we are still focusing on some vertex $v \in V_k$. Now, for each $x \in \{0,1\}^k$, let $P(x)$ denote the unique path in our simple sampler, from v_{start} to V_k, which has output labels corresponding to x. Furthermore, define $S_v := \{x \in \{0,1\}^k : P(x) \text{ hits } v\}$. Next, let

$$S_v^* := \{x \in S_v : \Delta(\mathsf{Enc}(x), y^*) \leq t\}.$$

We now observe that if $x \in S_v - S_v^*$, then $x \in \mathsf{Bad\_Messages}$. To see why, note that for each $x \in S_v - S_v^*$, it must hold that $y := \mathsf{Enc}(x)$ differs from y^* at $> t$ indices. Let $A \subseteq [\hat{k}]$ denote these indices. Then, for each $\alpha \in A$, it must hold that

$$\Pr[\mathbf{X} = (x, \alpha, b(x,\alpha)] = \Pr[\mathbf{X} = (x, \alpha, y_\alpha)] \leq \Pr[\mathbf{X} = (x, \alpha, \neg y_\alpha)] = \Pr[\mathbf{X} = (x, \alpha, \neg b(x,\alpha))],$$

because otherwise the random walk from v to V_{n+1} is strictly more likely to output (α, y_α) than $(\alpha, \neg y_\alpha)$, which means that it is strictly more likely to output $(\alpha, \neg y_\alpha^*)$ than (α, y_α^*) (since $\alpha \in A$), which contradicts the definition of y_α^*. Thus each $\alpha \in A$ is also in $\mathsf{Bad\_Indices}(x)$, and since $|A| > t$, we know that $|\mathsf{Bad\_Indices}(x)| > t$, and thus $x \in \mathsf{Bad\_Messages}$.

So we now know that $S_v - S_v^* \subseteq \mathsf{Bad\_Messages}$ for each $v \in V_k$, and thus

$$|\mathsf{Bad\_Messages}| \geq |\bigcup_{v \in V_k} (S_v - S_v^*)| \geq |\bigcup_{v \in V_k} S_v| - |\bigcup_{v \in V_k} S_v^*| \geq 2^k - w \cdot \max_{v \in V_k} |S_v^*|, \tag{4}$$

where the last inequality follows because each $x \in \{0,1\}^k$ falls into some S_v, and because $|V_k| \leq w$ by the width of our simple sampler.

We are finally ready to set t, and we set it to $t = \rho\hat{k}$. Notice that since $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^{\hat{k}}$ is the encoding function of a (ρ, L) list decodable code, we must have for every $v \in V_k$ that $|S_v^*| \leq L$. Thus, combining this observation with Equation (4) and Equation (3), we have

$$\begin{aligned}
|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| &> \frac{1}{2} \cdot |\mathsf{Bad_Messages}| \cdot t \cdot \frac{1}{2^k} \cdot \frac{1}{\hat{k}} \\
&\geq \frac{1}{2} \cdot (2^k - w \cdot \max_{v \in V_k} |S_v^*|) \cdot t \cdot \frac{1}{2^k} \cdot \frac{1}{\hat{k}} \\
&\geq \frac{1}{2} \cdot (2^k - wL) \cdot \rho\hat{k} \cdot \frac{1}{2^k} \cdot \frac{1}{\hat{k}} \\
&= \frac{1}{2} \cdot \rho \cdot (1 - wL \cdot 2^{-k}),
\end{aligned}$$

as desired. □

The next step is to show that complex samplers have an easy time sampling our enhanced addres function. The first step in this direction is the following fact, which is straightforward to verify.

Fact 7. *For any affine function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, there exists an ROBP of width $w = 2$ that computes f.*

Indeed, using just 1 bit of memory (width 2), the ROBP can keep track of a running sum (modulo 2) of the relevant bits. We now prove that complex samplers have an easy time sampling our address function, when it is instantiated with a linear function f.

Lemma 7. *Let $f : \{0,1\}^k \to \{0,1\}^{\hat{k}}$ be any $\mathbb{F}_2$-linear function, where $\hat{k}$ is a power of 2. We write $n := k + \log \hat{k}$, and let $b : \{0,1\}^n \to \{0,1\}$ be the address function over f from Definition 13. Then there is a complex sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width $w = 4\hat{k}$ that exactly samples $(\mathbf{U}_n, b(\mathbf{U}_n))$.*

Proof. The overall plan is to write the distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ as a convex combination of a few nicer distributions. We will show that each of the nicer distributions can easily be sampled by a small width complex sampler, and that we can combine these complex samplers to create one complex sampler $\mathbf{X}$ that exactly samples $(\mathbf{U}_n, b(\mathbf{U}_n))$.

Recall that $b : \{0,1\}^n \to \{0,1\}$ is of the form $b(x, \alpha) = f(x)_\alpha$, and note that f is of the form $f(x) = (f_1(x), \ldots, f_{\hat{k}}(x))$, where each $f_i : \{0,1\}^k \to \{0,1\}$ is a linear function. Thus each element of the support of $(\mathbf{U}_n, b(\mathbf{U}_n))$ is of the form $(x, \alpha, f_\alpha(x))$, where $x \in \{0,1\}^k$ and $\alpha \in [\hat{k}]$ and $f_\alpha : \{0,1\}^k \to \{0,1\}$ is a linear function. So let $\mathbf{B}$ be uniform over $\{0,1\}^k$, let $\mathbf{A}$ be uniform over $[\hat{k}]$, and write

$$(\mathbf{U}_n, b(\mathbf{U}_n)) = (\mathbf{B}, \mathbf{A}, f_{\mathbf{A}}(\mathbf{B})).$$

Now for every $\alpha \in [\hat{k}], c \in \{0,1\}$, let $p_{\alpha,c} := \Pr[\mathbf{A} = \alpha, f_\alpha(\mathbf{B}) = c]$. Also, let $\mathbf{U}_{\alpha,c} \sim \{0,1\}^k$ be the uniform distribution over the set of strings $x \in \{0,1\}^k$ where $f_\alpha(x) = c$. Furthermore, let $\mathbf{Y}_{\alpha,c}$ be the random variable $(\mathbf{U}_{\alpha,c}, \alpha, c)$. Notice that we can now write the convex combination

$$(\mathbf{U}_n, b(\mathbf{U}_n)) = \sum_{\alpha \in [\hat{k}], c \in \{0,1\}} p_{\alpha,c} \cdot \mathbf{Y}_{\alpha,c}.$$

We now show that for each $\mathbf{Y}_{\alpha,c}$, a complex sampler of width 2 can sample it. Note that it suffices to show a complex sampler of width 2 can sample $\mathbf{U}_{\alpha,c}$, since it is easy to have a complex sampler output a sequence of constant bits at the end. Recall that $\mathbf{U}_{\alpha,c}$ is uniform over $f_\alpha^{-1}(c)$, where f_α is an affine function. Thus by combining Fact 7 with Theorem 10, we see that $\mathbf{U}_{\alpha,c}$ can be sampled in width 2, and thus so can $\mathbf{Y}_{\alpha,c}$.

Now, in order to sample the convex combination written above, we do the following. For each $\mathbf{Y}_{\alpha,c}$, let $\mathcal{B}_{\alpha,c}$ be a complex sampler of width 2 that samples it. We define a complex sampler $\mathcal{B}^*$ of width $2\hat{k} \cdot 2 = 4\hat{k}$ that samples the convex combination, and thus $(\mathbf{U}_n, b(\mathbf{U}_n))$, as follows. Draw edges from the start vertex v^* of $\mathcal{B}^*$ to the start vertex of each $\mathcal{B}_{\alpha,c}$, and give it probability $p_{\alpha,c}$.

It is easy to see that a random walk over $\mathcal{B}^*$ samples the desired convex combination in width $4\hat{k}$, but technically a complex sampler must have an output bit on each edge. This is easy to fix: consider any $\mathcal{B}_{\alpha,c}$, and let $v_{\alpha,c}$ denote its start vertex. Instead of having an edge from v^* to $v_{\alpha,c}$, we can do the following: take each edge $(v_{\alpha,c}, u)$ leaving $v_{\alpha,c}$, and replace it with an edge (v^*, u) with the same output label, but multiply its probability by $p_{\alpha,c}$. It is not hard to see that this new version of $\mathcal{B}^*$ samples the same distribution as the old version (which is $(\mathbf{U}_n, b(\mathbf{U}_n))$, and indeed it is a bona fide complex sampler of width $4\hat{k}$, as desired. □

We are now ready to combine Lemma 6 and Lemma 7 to prove Theorem 13. To do so, we will use the following list-decodable codes of Guruswami and Rudra.

Theorem 14 ([GR08], Theorem 5.2)**.** *For any constant $\varepsilon > 0$ there exists constant $C, \delta > 0$ and an explicit $\mathbb{F}_2$-linear (ρ, L)-list decodable code $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^{\hat{k}}$ with dimension $k \geq \delta\hat{k}$, decoding radius $\rho = \frac{1}{2} - \varepsilon$, and list size $L = \hat{k}^C$.*

Finally, we prove Theorem 13 by combining Lemmas 6 and 7 and Theorem 14.

Proof of Theorem 13. Let $\varepsilon' > 0$ be a parameter to be set later, and let $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^{\hat{k}}$ be the list-decodable code from Theorem 14, which has decoding radius $\frac{1}{2} - \varepsilon'$ and dimension $k \geq \delta\hat{k}$. We let $n := k + \log \hat{k}$, and define $b : \{0,1\}^n \to \{0,1\}$ to be the function $\mathsf{address}_{\mathsf{Enc}}$, as defined by Definition 13. By Lemma 6, we know that for any simple sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width w,

$$|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| > \frac{1}{2} \cdot \rho \cdot (1 - wL \cdot 2^{-k}) = \frac{1}{2} \cdot \left(\frac{1}{2} - \varepsilon'\right) \cdot (1 - w \cdot (1/\delta)^C \cdot 2^{-k})$$

It is straightforward to verify that this can be lower bounded by $\frac{1}{4} - \varepsilon$ for width $w = 2^{cn}$, sufficiently small $c, \varepsilon' > 0$, and sufficiently large n. On the other hand, by Lemma 7, we know that there is a complex sampler $\mathbf{X}^* \sim \{0,1\}^{n+1}$ of width $w = 4\hat{k} \leq (1/\delta) \cdot k = O(n)$ that exactly samples $(\mathbf{U}_n, b(\mathbf{U}_n))$. □

5.1 Corresponding result for ROBPs

In this section, we briefly show how to combine our equivalence theorems with the above results in order to obtain our first main result: sampling with limited memory is easier than computing with limited memory.

Theorem 15 (Theorem 1, restated)**.** *For any fixed $\varepsilon > 0$, there exist constants $C, c > 0$ and an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that the following holds. For every ROBP $F : \{0,1\}^n \to \{0,1\}$ of width $w \leq 2^{cn}$,*

$$\Pr_x[F(x) \neq b(x)] > \frac{1}{4} - \varepsilon,$$

but there exists an ROBP $G : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $w \leq Cn$ and input length $\ell \leq Cn^3$ such that

$$G(\mathbf{U}_\ell) = (\mathbf{U}_n, b(\mathbf{U}_n)).$$

Proof. Let $b : \{0,1\}^n \to \{0,1\}$ be the function we used in the proof to Theorem 13. We know that any simple sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width $w \leq 2^{cn}$ has $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))|$. Combining this with Theorem 8 and the definition of correlation (Definition 3) yields the first part of the theorem. To obtain the second part of the result, simply note that the complex sampler $\mathbf{X}^*$ we designed for $(\mathbf{U}_n, b(\mathbf{U}_n))$ is 2^{-n}-granular (Definition 12), and thus we can simply apply Lemma 3. □

6 Sampling lower bounds against input-output pairs

In this section, we prove our main result about sampling input-output pairs. In particular, we prove the following.

Theorem 16. *There is an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any complex sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width w,*

$$|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 16w \cdot 2^{-n/8}.$$

In order to prove Theorem 16, the main ingredient we will use is an explicit *two-source extractor*. A two-source extractor for min-entropy k with error ε is a (deterministic) function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ such that for any two independent distributions $\mathbf{X}, \mathbf{Y} \sim \{0,1\}^n$ with min-entropy at least k each, it holds that $|\mathsf{Ext}(\mathbf{X}, \mathbf{Y}) - \mathbf{U}_1| \leq \varepsilon$. First, we show via the following lemma that b can be taken to be any two-source extractor. Then, we instantiate this lemma with a well-known explicit two-source extractor (the inner product function) in order to obtain Theorem 16.

Lemma 8 (Main lemma for Theorem 16)**.** *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *be a two-source extractor for min-entropy* k *with error* ε*. Then for any complex sampler* $\mathbf{X} \sim \{0,1\}^{2n+1}$ *of width* w*,*

$$|\mathbf{X} - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| \geq \frac{1}{2} - \varepsilon - 8w \cdot 2^{-n+k}.$$

Proof. $\mathbf{X} \sim \{0,1\}^{2n+1}$ is of the form $(\mathbf{A}, \mathbf{B}, \mathbf{b}) \sim \{0,1\}^n \times \{0,1\}^n \times \{0,1\}$, where the components are not necessarily independent. Let $\mathbf{W} \sim [w]^{2n+1}$ be the vertices hit on the random walk that generates $\mathbf{X}$, where $\mathbf{W}_n \sim [w]$ is the vertex hit in layer n of the branching program of the complex sampler. Using a standard observation [KM04, KM05], it holds that upon fixing $\mathbf{W}_n$ to any value $i \in [w]$, the random variables $\mathbf{A}$ and $(\mathbf{B}, \mathbf{b})$ become independent. That is, we can write $\mathbf{X}$ as a convex combination of the form

$$\mathbf{X} = \sum_{i \in [w]} p_i \cdot (\mathbf{A}^{(i)}, \mathbf{B}^{(i)}, \mathbf{b}^{(i)}),$$

where p_i is the probability that $\mathbf{W}_n$ hits vertex i in layer n of the complex sampler, and where each $\mathbf{A}^{(i)}$ is independent of $\mathbf{B}^{(i)}, \mathbf{b}$. Consider now fixing each $\mathbf{b}^{(i)}$ to some $b \in \{0,1\}$. Then we can clearly write

$$\mathbf{X} = \sum_{i \in [w], b \in \{0,1\}} p_{i,b} \cdot (\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b), \tag{5}$$

for some probabilities $p_{i,b}$. Notice also that $\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}$ remain independent. Now, by definition of statistical distance, it holds that for any test $S \subseteq \{0,1\}^{2n+1}$,

$$|\mathbf{X} - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| \geq \Pr[\mathbf{X} \in S] - \Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S].$$

We now aim to construct a test S by keeping the above mentioned convex combination in mind. Towards this end, we start by letting t be a parameter that we will fix later. For each $i \in [w]$ and $b \in \{0,1\}$, we define

$$\begin{aligned}
\mathsf{Bad}_A^{(i,b)} &:= \{x \in \{0,1\}^n : \Pr[\mathbf{A}^{(i,b)} = x] > 2^{-t}\},\\
\mathsf{Bad}_B^{(i,b)} &:= \{y \in \{0,1\}^n : \Pr[\mathbf{B}^{(i,b)} = y] > 2^{-t}\},\\
\mathsf{Bad}^{(i,b)} &:= \left\{(x,y,b) \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\} : x \in \mathsf{Bad}_A^{(i,b)} \text{ or } y \in \mathsf{Bad}_B^{(i,b)}\right\}\\
\mathsf{Bad} &:= \bigcup_{i \in [w], b \in \{0,1\}} \mathsf{Bad}^{(i,b)}
\end{aligned}$$

Furthermore, we define the set

$$T := \{(x,y,b) \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\} : \mathsf{Ext}(x,y) \neq b\}.$$

At last, we are ready to define our test set $S \subseteq \{0,1\}^{2n+1}$ as:

$$S := T \cup \mathsf{Bad}.$$

The goal now is to lower bound $\Pr[\mathbf{X} \in S]$, and upper bound $\Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S]$. We start with the latter, as it is easier. Notice that it is impossible for $(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))$ to land in T, so we just have to worry about it landing in Bad. Notice also that each bad set of the form $\mathsf{Bad}_A^{(i,b)}$ or $\mathsf{Bad}_B^{(i,b)}$ has $< 2^t$ elements, or else it contradicts the definition of probability distribution. Thus for every i, b we have

$$|\mathsf{Bad}^{(i,b)}| \leq |\mathsf{Bad}_A^{(i,b)} \times \{0,1\}^n| + |\{0,1\}^n \times \mathsf{Bad}_B^{(i,b)}| < 2^t \cdot 2^n + 2^n \cdot 2^t = 2^{t+n+1},$$

which yields

$$|\mathsf{Bad}| < 2w \cdot 2^{t+n+1} = 4w \cdot 2^{t+n}.$$

Now notice that $(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))$ assigns each element in its support a probability of 2^{-2n}, so we have

$$\Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S] = \Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in \mathsf{Bad}] \leq 2^{-2n} \cdot |\mathsf{Bad}| < 4w \cdot 2^{-n+t}. \tag{6}$$

We now move towards lower bounding $\Pr[\mathbf{X} \in S]$. By Equation (5), it suffices to lower bound $\Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S]$ for every i, b. Now, let

$$p := \Pr[\mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)} \text{ and } \mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)}]$$

and note that we can rewrite $\Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S]$ as

$$\begin{aligned} &p \cdot \Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S \mid \mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)} \text{ and } \mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)}] \\ +&(1-p) \cdot \Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S \mid \mathbf{A}^{(i,b)} \in \mathsf{Bad}_A^{(i,b)} \text{ or } \mathbf{B}^{(i,b)} \in \mathsf{Bad}_B^{(i,b)}] \end{aligned}$$

Notice that the probability attached to $(1-p)$ will always be 1, by our construction of S. On the other hand, in the term attached to p, we can replace S with T since $(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b)$ will never hit Bad in this conditioning. And the probability that $(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in T$ is, by definition of T, the probability that $\mathsf{Ext}(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}) \neq b$. Thus we can rewrite the above expression as

$$p \cdot \Pr[\mathsf{Ext}(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}) \neq b \mid \mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)} \text{ and } \mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)}] + (1-p).$$

Notice now that since $\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}$ were originally independent, the conditionings above keep them independent. In particular, we can define independent random variables $\widetilde{\mathbf{A}}^{(i,b)} := (\mathbf{A}^{(i,b)} \mid \mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)})$ and $\widetilde{\mathbf{B}}^{(i,b)} := (\mathbf{B}^{(i,b)} \mid \mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)})$ and the above expression becomes

$$p \cdot \Pr[\mathsf{Ext}(\widetilde{\mathbf{A}}^{(i,b)}, \widetilde{\mathbf{B}}^{(i,b)}) \neq b] + (1-p). \tag{7}$$

We would now like to get a lower bound on the entropy of each input to the extractor. Towards this end, we start by defining the probabilities

$$\begin{aligned} q_A &:= \Pr[\mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)}], \\ q_B &:= \Pr[\mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)}], \end{aligned}$$

and we observe that $p = q_A \cdot q_B \leq \min\{q_A, q_B\}$. We now have two possible cases. In the first case, either q_A or q_B is at most $1/2$. In this case, $p \leq 1/2$ and $1 - p \geq 1/2$, which implies that Equation (7) is $\geq 1/2$.

In the second possible case, both q_A and q_B are $> 1/2$. In this case, it is straightforward to verify the following min entropy lower bounds:

$$\begin{aligned} H_\infty(\widetilde{\mathbf{A}}^{(i,b)}) &= \log\left(\frac{1}{\max_{x \notin \mathsf{Bad}_A^{(i,b)}} \Pr[\widetilde{\mathbf{A}}^{(i,b)} = x]}\right) = \log\left(\frac{q_A}{\max_{x \notin \mathsf{Bad}_A^{(i,b)}} \Pr[\mathbf{A}^{(i,b)} = x]}\right) \\ &> \log\left(\frac{\frac{1}{2}}{2^{-t}}\right) = t - 1, \end{aligned}$$

where the last inequality follows from the definition of bad sets. Of course, using the same reasoning,

$$H_\infty(\widetilde{\mathbf{B}}^{(i,b)}) > t - 1.$$

We are finally ready to pick t. We set it to $t := k + 1$, so that both of the above min-entropies become $> k$. Now, since Ext is a two-source extractor for min-entropy k with error ε, and since we are calling it on two independent sources of min-entropy k, we the extractor property tells us

$$\begin{aligned} p \cdot \Pr[\mathsf{Ext}(\widetilde{\mathbf{A}}^{(i,b)}, \widetilde{\mathbf{B}}^{(i,b)}) \neq b] + (1-p) &\geq p \cdot (\frac{1}{2} - \varepsilon) + (1-p) \\ &= 1 - p \cdot (\frac{1}{2} + \varepsilon) \geq 1 - (\frac{1}{2} + \varepsilon) \\ &= \frac{1}{2} - \varepsilon. \end{aligned}$$

Thus, we finally see that in the case where both q_A and q_B are $> 1/2$, then Equation (7) is $\geq 1/2 - \varepsilon$ (given that we set $t := k + 1$). Thus in all cases, Equation (7) is $\geq 1/2 - \varepsilon$. And tracing back to the expression it originally represented, we get

$$\Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S] \geq \frac{1}{2} - \varepsilon.$$

And we know that this holds for all i, b, since we made no assumption on their values. As discussed earlier, this therefore implies

$$\Pr[\mathbf{X} \in S] \geq \frac{1}{2} - \varepsilon, \tag{8}$$

as long as we set $t = k + 1$. Wrapping everything up, we combine Equation (6) and Equation (8) to get

$$\begin{aligned} |\mathbf{X} - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| &\geq \Pr[\mathbf{X} \in S] - \Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S] \\ &> \frac{1}{2} - \varepsilon - 4w \cdot 2^{-n+t} \\ &= \frac{1}{2} - \varepsilon - 4w \cdot 2^{-n+k+1} \\ &= \frac{1}{2} - \varepsilon - 8w \cdot 2^{-n+k}, \end{aligned}$$

as desired. □

We now turn back to proving Theorem 16. We will instantiate Lemma 8 with the following two source extractor.

Theorem 17 ([Vaz85, CG88])**.** *Let* $\mathsf{IP} : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2$ *denote the inner product over* $\mathbb{F}_2$*. Then* IP *is a two-source extractor for min-entropy* k *with error* $\varepsilon = 2^{-(2k-n-1)/2}$*.*

We are now ready to prove Theorem 16.

Proof of Theorem 16. First, assume without loss of generality that $n \geq 2$ (if $n = 1$ the statement is trivial to show). Write $n = 2\ell + a$ for some positive ℓ and nonnegative a. We take b to be the function $\mathsf{IP}^* : \{0,1\}^n \to \{0,1\}$, which will just be the inner product function extended to handle odd length n. It is defined as follows. On input $x \in \{0,1\}^n$, let z denote the last ℓ bits of x and let y denote the ℓ bits before that. Then IP^* will output $\mathsf{IP}(y, z)$ (from Theorem 17).

Suppose now that $a = 0$, namely that $n = 2\ell$. Then $\mathsf{IP}^*(x) = \mathsf{IP}(x)$, and so plugging Theorem 17 into Lemma 8 we get

$$|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{-(2k-\ell-1)/2} - 8w \cdot 2^{-\ell+k}.$$

Plugging in $k = 3\ell/4$ yields

$$|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 10w \cdot 2^{-\ell/4} = \frac{1}{2} - 10w \cdot 2^{-n/8}.$$

Suppose now that $a = 1$, namely that $n = 2\ell + 1$. Let $\mathbf{Y} = (\mathbf{U}_n, b(\mathbf{U}_n))$. Let $\mathbf{X}_{-1}$ denote all but the first bit of $\mathbf{X}$, and let $\mathbf{Y}_{-1}$ denote all but the first bit of $\mathbf{Y}$. It is easy to verify that $|\mathbf{X} - \mathbf{Y}| \geq |\mathbf{X}' - \mathbf{Y}'|$. Furthermore, it is straightforward to verify that $\mathbf{X}'$ can be sampled by a complex sampler of the same width w, and that $\mathbf{Y}' = (\mathbf{U}_{2\ell}, \mathsf{IP}(\mathbf{U}_{2\ell}))$. Thus following the same steps as before we get:

$$\begin{aligned}
|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| &\geq |\mathbf{X}' - \mathbf{Y}'| \geq \frac{1}{2} - 10w \cdot 2^{-\ell/4} \\
&= \frac{1}{2} - 10w \cdot 2^{-(n-1)/8} \\
&\geq \frac{1}{2} - 16w \cdot 2^{-n/8},
\end{aligned}$$

where we did not focus on optimizing the constant (and chose to make it a power of 2). Thus for odd and even n we get the claimed bound. □

6.1 An extension to unknown-order samplers

We now show that, without too much trouble, our lower bounds against input-output pairs (Theorem 16) can be extended to the unknown-order setting. In particular, we prove the following:

Theorem 18. *There exist universal constants $C, \delta > 0$ such that for any prime $p \in \mathbb{N}$ and $n := Cp$, there exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any unknown-order complex sampler $\mathbf{X} \sim \{0,1\}^{n+1}$ of width w,*

$$|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 32w \cdot 2^{-\delta n}.$$

In order to prove Theorem 18, the main ingredient we will use is an explicit extractor for *interleaved sources*. This can be thought of as an "unknown-order" two-source extractor. More formally, an extractor for two interleaved sources, for min-entropy k with error ε, is a (deterministic) function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ with the following property: for any two independent $\mathbf{X}, \mathbf{Y} \sim \{0,1\}^n$, each with min-entropy at least k, and any permutation $\pi : [2n] \to [2n]$, it holds that $|\mathsf{Ext}((\mathbf{X}, \mathbf{Y})^\pi) - \mathbf{U}_1| \leq \varepsilon$. Just like in the known-order setting, we start with the following lemma, which shows that we can take b to be any extractor of the appropriate type. Then, we instantiate the lemma with an explicit extractor from Raz and Yehudayoff in order to obtain Theorem 18.

Lemma 9 (Main lemma for Theorem 18)**.** *Let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be an extractor for two interleaved sources, for min-entropy k with error ε. Then for any unknown-order complex sampler $\mathbf{X} \sim \{0,1\}^{2n+1}$ of width w,*

$$|\mathbf{X} - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| \geq \frac{1}{2} - \varepsilon - 16w \cdot 2^{-n+k}.$$

In order to prove this lemma, we will use Lemma 8 as a black box. Towards this end, we will use the following fact about (known-order) complex samplers, which is straightforward to verify.

Fact 8. *For any complex sampler $\mathbf{X} \sim \{0,1\}^n$ of width w, and any $i \in [n]$, there exists a complex sampler $\mathbf{X}'$ of width $2w$ that samples the distribution $(\mathbf{X}_1, \dots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \mathbf{X}_n, \mathbf{X}_i)$.*

Indeed, note that such an $\mathbf{X}'$ can be constructed from $\mathbf{X}$ by suppressing its i^{th} bit of output, writing this bit to its memory instead (using an extra bit), and then writing this bit of memory to the output at the very end. With this fact in hand, we now prove Lemma 9.

Proof of Lemma 9. Let $\mathbf{X} \sim \{0,1\}^{2n+1}$ be an unknown-order complex sampler of width w. By definition, there exists a complex sampler $\mathbf{Y} \sim \{0,1\}^{2n+1}$ of width w and a permutation $\pi : [2n+1] \to [2n+1]$ such that $\mathbf{X} = \mathbf{Y}^\pi$. Thus by the data-processing inequality (Fact 1) we have

$$|\mathbf{X} - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| = |\mathbf{Y}^\pi - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| \geq |\mathbf{Y} - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))^{\pi^{-1}}|.$$

Now, we say that a permutation $\rho : [2n+1] \to [2n+1]$ is a *deletion permutation* if there is some $i \in [2n+1]$ such that for all $z \in \{0,1\}^{2n+1}$ we have $z^\pi = (z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_{2n+1}, z_i)$. It is straightforward to verify that there exists a deletion permutation $\rho : [2n+1] \to [2n+1]$ and some other permutation $\sigma : [2n] \to [2n]$ such that

$$((\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))^{\pi^{-1}})^\rho = (\mathbf{U}_{2n}^\sigma, \mathsf{Ext}(\mathbf{U}_{2n})).$$

Combining this with another application of the data-processing inequality (Fact 1), we have

$$|\mathbf{Y} - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))^{\pi^{-1}}| \geq |\mathbf{Y}^\rho - ((\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))^{\pi^{-1}})^\rho| = |\mathbf{Y}^\rho - (\mathbf{U}_{2n}^\sigma, \mathsf{Ext}(\mathbf{U}_{2n}))|.$$

Recall that $\mathbf{Y}$ is a complex sampler of width w, and ρ is a deletion permutation. Thus by Fact 8 there exists a complex sampler $\mathbf{Y}'$ of width $2w$ such that $\mathbf{Y}' = \mathbf{Y}^\rho$, and in particular such that

$$|\mathbf{Y}^\rho - (\mathbf{U}_{2n}^\sigma, \mathsf{Ext}(\mathbf{U}_{2n}))| = |\mathbf{Y}' - (\mathbf{U}_{2n}^\sigma, \mathsf{Ext}(\mathbf{U}_{2n}))|.$$

Next, it is straightforward to verify that $(\mathbf{U}_{2n}^\sigma, \mathsf{Ext}(\mathbf{U}_{2n})) = (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}^{\sigma^{-1}}))$. Now, define the function $\mathsf{Ext}' : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ as $\mathsf{Ext}'(x,y) := \mathsf{Ext}((x,y)^{\sigma^{-1}})$. Clearly we have $(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}^{\sigma^{-1}})) = (\mathbf{U}_{2n}, \mathsf{Ext}'(\mathbf{U}_{2n}))$. Furthermore, since Ext is an extractor for two interleaved sources for min-entropy k with error ε, it is immediate that Ext' is a two-source extractor for min-entropy k and error ε. Combining everything, we have

$$|\mathbf{X} - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| \geq |\mathbf{Y}' - (\mathbf{U}_{2n}^\sigma, \mathsf{Ext}(\mathbf{U}_{2n}))| = |\mathbf{Y}' - (\mathbf{U}_{2n}, \mathsf{Ext}'(\mathbf{U}_{2n}))|,$$

where $\mathbf{Y}'$ is a complex sampler of width $2w$ and Ext' is a two-source extractor for min-entropy k and error ε. By Lemma 8, this statistical distance is at least $\frac{1}{2} - 16w \cdot 2^{-n+k}$, which completes the proof. □

We now turn back to proving Theorem 18. We will instantiate Lemma 9 with the following extractor of Raz and Yehudayoff for two interleaved sources.

Theorem 19 ([RY11])**.** *There exist universal constants $C, \delta > 0$ such that for all prime $p \in \mathbb{N}$ and $n := Cp$, there exists an explicit extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ for two interleaved sources, for min-entropy $k \geq (1-\delta)n$ with error $\varepsilon = 2^{-\delta n}$.*

We are now ready to prove Theorem 18.

Proof of Theorem 18. Let $C', \delta' > 0$ be the universal constants in Theorem 19, let $C = 2C'$, and let $\delta > 0$ be a constant to be defined later. Now, let $b : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}$ be the explicit extractor for two interleaved sources given by Theorem 19 for min-entropy $k \geq (1-\delta')n/2$ with error $\varepsilon = 2^{-\delta' n/2}$. By Lemma 9 we have

$$\begin{aligned}|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| &\geq \frac{1}{2} - \varepsilon - 16w \cdot 2^{-n/2+k} = \frac{1}{2} - 2^{-\delta' n/2} - 16w \cdot 2^{-n/2+(1-\delta')n/2} \\ &\geq \frac{1}{2} - 32w \cdot 2^{-\delta' n/2} = \frac{1}{2} - 32w \cdot 2^{-\delta n}\end{aligned}$$

for $\delta := \delta'/2 > 0$, as desired. □

6.2 Corresponding results for ROBPs

In this section, we briefly show how to combine our equivalence theorems with the above results in order to obtain our second main result: sampling lower bounds against input-output pairs for ROBPs.

Theorem 20 (Theorem 2, restated)**.** *There is a universal constant $c > 0$ and an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $w \leq 2^{cn}$,*

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{-cn}.$$

Proof. Combine Theorem 16 with the first bullet of Theorem 6. □

Furthermore, we also obtain the stronger unknown-order version of the above theorem:

Theorem 21. *There is a universal constant $c > 0$ and an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any unknown-order ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $w \leq 2^{cn}$,*

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{-cn}.$$

Proof. Combine Theorem 18 with the first bullet of Theorem 6 and the definition of unknown-order ROBPs and samplers (Section 3.3). □

Furthermore, recall that our function $b : \{0,1\}^n \to \{0,1\}$ can be taken as the inner product function. Since $(\mathbf{U}_n, b(\mathbf{U}_n))$ is known to be samplable by AC^0 circuits [IN96], Theorem 20 gives the following complexity separation.

Corollary 8. *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that $|F(\mathbf{U}_\ell) - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{-\Omega(n)}$ for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $2^{\Omega(n)}$, but such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be exactly sampled by AC^0 circuits.*

Furthermore, by combining Theorem 16 with Theorem 8, we obtain the following corollary.

Corollary 9. *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any ROBP $F : \{0,1\}^n \to \{0,1\}$ of width $2^{\Omega(n)}$, it holds that $|\operatorname{corr}(F, b)| \leq 2^{-\Omega(n)}$.*

Finally, by combining Theorem 18 with Theorem 9, we obtain the following.

Corollary 10. *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any unknown-order ROBP $F : \{0,1\}^n \to \{0,1\}$ of width $2^{\Omega(n)}$, it holds that $|\operatorname{corr}(F, b)| \leq 2^{-\Omega(n)}$.*

More generally, note that we can use Lemma 8 and Lemma 9 to obtain correlation bounds against two-source extractors and extractors for two interleaved sources.

7 Sampling lower bounds against list-decodable codes

In this section, we will prove our main sampling lower bound against list decodable codes. In particular, we prove the following.

Theorem 22. *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over a (ρ, L) list decodable code of dimension k. Then for any complex sampler $\mathbf{X} \sim \{0,1\}^n$ of width w,*

$$|\mathbf{X} - \mathbf{Q}| \geq 1 - 4wL \cdot 2^{-\frac{\rho}{1+2\rho}k}.$$

Note that by combining this theorem with Theorem 6, we immediately get Theorem 3. Furthermore, by combining it with Theorem 11, we immediately get Corollary 4.

Proof of Theorem 22. We start by writing our complex sampler as a convex combination of random variables with nice structure. Let $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_n)$ be the vertices hit on the random walk that generates $\mathbf{X}$ (excluding the start vertex of the branching program). Let r, ℓ be positive integers that will be set later to ensure $r\ell = n$. Define $\mathbf{W}^* = (\mathbf{W}_\ell, \mathbf{W}_{2\ell}, \ldots, \mathbf{W}_{r\ell})$, and recall the following standard observation (first made in [KM04, KM05, KRVZ11]): for any $W \in \text{support}(\mathbf{W}^*)$, the random variable $(\mathbf{X} \mid \mathbf{W}^* = W)$ is of the form $\mathbf{X}^{(W)} := (\mathbf{X}_1^{(W)}, \mathbf{X}_2^{(W)}, \ldots, \mathbf{X}_r^{(W)})$, where each $\mathbf{X}_i^{(W)} \sim \{0,1\}^\ell$ is independent. Thus the complex sampler $\mathbf{X}$ is a convex combination of the form

$$\mathbf{X} = \sum_{W \in \text{support}(\mathbf{W}^*)} p_W \cdot \mathbf{X}^{(W)},$$

where each $p_W := \Pr[\mathbf{W}^* = W]$.

The goal now is to use the above decomposition to help us get a good lower bound on $|\mathbf{X} - \mathbf{Q}| = \max_S |\Pr[\mathbf{X} \in S] - \Pr[\mathbf{Q} \in S]|$. Towards this end, we note that for any S,

$$\begin{aligned}|\mathbf{X} - \mathbf{Q}| &\geq \Pr[\mathbf{Q} \in S] - \Pr[\mathbf{X} \in S] = \Pr[\mathbf{Q} \in S] - \sum_W p_W \cdot \Pr[\mathbf{X}^{(W)} \in S] \\ &\geq \Pr[\mathbf{Q} \in S] - \max_W \Pr[\mathbf{X}^{(W)} \in S].\end{aligned}$$

Thus, we would like to pick a test S that maximizes the quantity $\Pr[\mathbf{Q} \in S]$ while minimizing the quantity $\max_W \Pr[\mathbf{X}^{(W)} \in S]$. A natural candidate for S is the entire codebook $Q = \text{support}(\mathbf{Q})$, minus some small set of "bad codewords" Bad, which are assigned too high of a probability by some $\mathbf{X}^{(W)}$. As such, we let $t > 0$ be a parameter to be set later, and we define $S = Q - \mathsf{Bad}$ where

$$\begin{aligned}\mathsf{Bad} &:= \bigcup_W \mathsf{Bad}^{(W)}, \\ \mathsf{Bad}^{(W)} &:= \{q \in Q : \Pr[\mathbf{X}^{(W)} = q] > 2^{-t}\}.\end{aligned}$$

Plugging in this definition of S, we get

$$\begin{aligned}|\mathbf{X} - \mathbf{Q}| &\geq \Pr[\mathbf{Q} \in Q - \mathsf{Bad}] - \max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}] \\ &\geq 1 - \Pr[\mathbf{Q} \in \mathsf{Bad}] - \max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}^{(W)}].\end{aligned}$$

Thus, we would like to upper bound both quantities that are subtracted. To upper bound the first quantity, simply note that

$$\Pr[\mathbf{Q} \in \mathsf{Bad}] = 2^{-k} \cdot |\mathsf{Bad}| \leq 2^{-k} \sum_{W \in \text{support}(\mathbf{W}^*)} |\mathsf{Bad}^{(W)}| < 2^{-k+t+r\log(w)}$$

via the trivial upper bounds $|\text{support}(\mathbf{W}^*)| \leq w^r$ and $|\mathsf{Bad}^{(W)}| < 2^t$ for each W.

To upper bound the second quantity $\max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}^{(W)}]$, we start by making notation more convenient: let W^* be the maximizer of the above quantity, and define $\mathbf{Y} := \mathbf{X}^{(W^*)}$ and $\mathsf{Bad}^* := \mathsf{Bad}^{(W^*)}$. Of course we have $\max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}^{(W)}] = \Pr[\mathbf{Y} \in Q - \mathsf{Bad}^*]$, and we focus on upper bounding the latter.

Recall that $\mathbf{Y}$ is of the form $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_r)$ where each $\mathbf{Y}_i \sim \{0,1\}^\ell$ is independent, and $\mathsf{Bad}^* := \{q \in Q : \Pr[\mathbf{Y} = q] > 2^{-t}\}$ contains all codewords hit by $\mathbf{Y}$ with large probability. Thus each $q \in Q - \mathsf{Bad}^*$ must have $\Pr[\mathbf{Y} = q] \leq 2^{-t}$. So, if we parse each q as $(q_1, q_2, \ldots, q_r) \in (\{0,1\}^\ell)^r$, we have that $\Pr[\mathbf{Y} = q] = \Pr[\mathbf{Y}_1 = q_1] \cdot \Pr[\mathbf{Y}_2 = q_2] \cdots \Pr[\mathbf{Y}_r = q_r]$ by the independence of these random variables, and so there must be some $\pi(q) \in [r]$ such that $\Pr[\mathbf{Y}_{\pi(q)} = q_{\pi(q)}] \leq 2^{-t/r}$.

Now, for a string $x = (x_1, x_2, \ldots, x_r) \in (\{0,1\}^\ell)^r$, we let $x_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r)$ denote x with its i^{th} chunk removed, and proceed as follows:

$$\begin{aligned}
\Pr[\mathbf{Y} \in Q - \mathsf{Bad}^*] &= \sum_{q \in Q - \mathsf{Bad}^*} \Pr[\mathbf{Y} = q] \\
&= \sum_{q \in Q - \mathsf{Bad}^*} \Pr[\mathbf{Y}_{\pi(q)} = q_{\pi(q)}] \cdot \Pr[\mathbf{Y}_{-\pi(q)} = q_{-\pi(q)}] \\
&\leq 2^{-t/r} \sum_{q \in Q - \mathsf{Bad}^*} \Pr[\mathbf{Y}_{-\pi(q)} = q_{-\pi(q)}] \\
&\leq 2^{-t/r} \sum_{q \in Q - \mathsf{Bad}^*} \Pr[\mathbf{Y} \in \mathsf{Ball}(q, \ell)] \\
&= 2^{-t/r} \sum_{v \in \{0,1\}^n} \Pr[\mathbf{Y} = v] \cdot \#\{q \in Q - \mathsf{Bad}^* : v \in \mathsf{Ball}(q, \ell)\} \\
&\leq 2^{-t/r} \sum_{v \in \{0,1\}^n} \Pr[\mathbf{Y} = v] \cdot \#\{q \in Q : \Delta(v, q) \leq \ell\} \\
&\leq 2^{-t/r} \sum_{v \in \{0,1\}^n} \Pr[\mathbf{Y} = v] \cdot |\mathsf{Ball}(v, \ell) \cap Q| \\
&\leq 2^{-t/r} \cdot \max_v |\mathsf{Ball}(v, \ell) \cap Q| \\
&\leq 2^{-t/r} \cdot L \textbf{ if } \ell \leq \rho n,
\end{aligned}$$

where the last line follows since Q is a (ρ, L)-list decodable code (see Definition 6). Thus, provided that we have selected $r, \ell \in \mathbb{N}$ such that $r\ell = n$ and $\ell \leq \rho n$, we can combine all of the above to get

$$\begin{aligned}
|\mathbf{X} - \mathbf{Q}| &\geq 1 - \Pr[\mathbf{Q} \in \mathsf{Bad}] - \max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}^{(W)}] \\
&= 1 - \Pr[\mathbf{Q} \in \mathsf{Bad}] - \Pr[\mathbf{Y} \in Q - \mathsf{Bad}^*] \\
&> 1 - 2^{-k+t+r\log w} - 2^{-t/r+\log L}.
\end{aligned}$$

Before picking r, ℓ, we set[9] $t = \frac{r}{r+1} \cdot (k - r \log w + \log L)$ as the value that equalizes the two exponents to $-\frac{1}{r+1} \cdot (k - r \log w + \log L) + \log L \leq -\frac{k}{r+1} + \log(wL)$ to obtain

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-t/r + \log L + 1} \geq 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1}.$$

Thus all that remains is to pick $r, \ell \in \mathbb{N}$ such that $r\ell = n$. If $1/\rho$ and ρn are integers, we simply set $r = 1/\rho$ and $\ell = \rho n$ to obtain

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1} = 1 - 2wL \cdot 2^{-\frac{\rho}{1+\rho} k}.$$

If $1/\rho$ and ρn are not integers, there is an easy way to slightly modify the proof so that everything works out (with a minor loss in parameters): first, recall that we originally defined $\mathbf{W}^* = (\mathbf{W}_\ell, \mathbf{W}_{2\ell}, \ldots, \mathbf{W}_{r\ell})$ so that each $(\mathbf{X} \mid \mathbf{W}^* = W)$ is of the form $\mathbf{X}^{(W)} := (\mathbf{X}_1^{(W)}, \ldots, \mathbf{X}_r^{(W)})$, where each $\mathbf{X}_i^{(W)}$ is independent and over ℓ bits. Observe that the argument actually does not require that each $\mathbf{X}_i^{(W)}$ has the same length; instead, it simply requires that each $\mathbf{X}_i^{(W)}$ has length at most ρn. Thus, we could have actually started with any $\mathbf{W}^*$ of the form

$$\mathbf{W}^* = (\mathbf{W}_{\alpha_1}, \mathbf{W}_{\alpha_2}, \ldots, \mathbf{W}_{\alpha_r}),$$

where $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_r = n$, and each gap $\alpha_j - \alpha_{j-1}$ is bounded above by ρn. And the exact same argument as above yields

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1}. \tag{9}$$

Thus, we now have more flexibility in dealing with the case where $1/\rho$ and ρn are not integers: we can simply pick an integer r and define $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_r = n$ such that each gap is at most ρn. In more detail, we can force the first $r - 1$ gaps to be exactly $\lfloor \rho n \rfloor$, while the last gap is at most $\lfloor \rho n \rfloor$, by picking $r := \left\lceil \frac{n}{\lfloor \rho n \rfloor} \right\rceil$ and setting $\alpha_j := \lfloor \rho n \rfloor \cdot j$ for all $j \in [r-1]$.

Before we plug the value of r into Equation (9), it is useful to get a clean lower bound on $\frac{1}{r+1}$:

$$\begin{aligned}\frac{1}{r+1} = \frac{1}{\lceil \frac{n}{\lfloor \rho n \rfloor} \rceil + 1} \geq \frac{1}{\frac{n}{\lfloor \rho n \rfloor} + 2} \geq \frac{1}{\frac{n}{\rho n - 1} + 2} = \frac{\rho n - 1}{n + 2(\rho n - 1)} = \frac{\rho n}{n + 2(\rho n - 1)} - \frac{1}{n + 2(\rho n - 1)} \\ \geq \frac{\rho n}{n + 2\rho n} - \frac{1}{n} = \frac{\rho}{1 + 2\rho} - \frac{1}{n} \geq \frac{\rho}{1 + 2\rho} - \frac{1}{k},\end{aligned}$$

where the last inequality follows since $k \leq n$, because a code's dimension cannot exceed the dimension in which it lives. At last, we can plug our lower bound for $1/(r+1)$ into Equation (9) to obtain

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1} \geq 1 - 2^{-k \cdot (\frac{\rho}{1+2\rho} - \frac{1}{k}) + \log(wL) + 1} = 1 - 4wL \cdot 2^{-\frac{\rho}{1+2\rho} k},$$

which completes the proof. □

[9] Technically we originally asked for $t > 0$, but we may assume this without loss of generality, since if this setting of t is nonpositive, then the claimed result will become $|\mathbf{X} - \mathbf{Q}| \geq 0$, which is trivially true.

7.1 Nearly-tight sampling lower bounds against (n, k, d) codes

In the previous section, we gave very general sampling lower bounds against list-decodable codes via Theorem 22. By combining this result with the list-decodability of (n, k, d) codes (Fact 2), we immediately obtain the following sampling lower bounds against (n, k, d) codes, which are nearly tight.

Theorem 23. *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over an (n, k, d) code. Then for any complex sampler $\mathbf{X} \sim \{0,1\}^n$ of width w,*

$$|\mathbf{X} - \mathbf{Q}| \geq 1 - 8w \cdot 2^{-\frac{kd}{4n}}.$$

In particular, we get that for any distribution $\mathbf{Q} \sim \{0,1\}^n$ that is uniform over a good code, every complex sampler $\mathbf{X} \sim \{0,1\}^n$ of width $2^{\Omega(n)}$ has $|\mathbf{X} - \mathbf{Q}| \geq 1 - 2^{-\Omega(n)}$. Furthermore, we can show that Theorem 23 is almost tight, in the following sense: for almost all "valid" n, k, d, there exists an (n, k, d) code $\mathbf{Q} \sim \{0,1\}^n$ that can be exactly sampled by a complex sampler of width $w = 2^{\widetilde{O}(\frac{kd}{n})}$. More formally, we show the following.

Theorem 24. *There is a universal constant $C > 0$ such that the following holds. For all $n, k, d \in \mathbb{N}$ such that there exists a linear $[n, k, d]$ code, there exists a distribution $\mathbf{Q} \sim \{0,1\}^n$ uniform over a linear $[n, k, d]$ code that can be exactly generated by a complex sampler $\mathbf{X} \sim \{0,1\}^n$ of width $w \leq C \cdot 2^{C \cdot \frac{kd}{n} \cdot \log n}$.*

Proof. We split the casework into $k \leq 0.9n$ and $k > 0.9n$.

Case: $k \leq 0.9n$: In this case, we will show that there is a linear $[n, k, d]$ code $Q \subseteq \{0,1\}^n$ such that the uniform distribution $\mathbf{Q}$ over Q can be sampled in width $w \leq C \cdot 2^{Ckd/n}$ for a sufficiently large constant C. To construct Q, the general idea will be to take an $[n', k', d']$ code Q' over a smaller space $\{0,1\}^{n'}$ and repeat it n/n' times.

In more detail, let C be a sufficiently large constant to be chosen later, and set $n' = Cd$. We will aim to repeat our smaller code $t := \lfloor n/n' \rfloor$ times. We may assume $t \geq 20$: Otherwise, $20n'/n = 20Cd/n > 1$ and we can sample the linear $[n, k, d]$ code (guaranteed to exist by the hypothesis) by a simple sampler of width $w = 2^k < 2^{k \cdot 20Cd/n}$ (by Fact 4), and we are done. So we can henceforth assume $t \geq 20$.

Note that there must be some integers $n_1, \ldots, n_t$ such that each $n_i \geq n'$ and $\sum_i n_i = n$. Suppose now that there exist a collection of codes $\{Q_i\}_{i \in [t]}$ such that all of the following hold:

- Each Q_i is a linear $[n_i, k_i, d_i]$ code.
- Each $k_i = \lceil k/t \rceil$.
- Each $d_i = d$.

Then $\sum_i k_i \geq k$, and we may of course find a collection of linear codes $\{Q'_i\}_{i \in [t]}$ with the same properties as $\{Q_i\}_{i \in [t]}$, except that the property $k_i = \lceil k/t \rceil$ is traded for the properties $k_i \leq \lceil k/t \rceil$ and $\sum_i k_i = k$ (simply by reducing the dimension of each code by an appropriate amount). Notice that $Q'_1 \times Q'_2 \times \cdots \times Q'_t$ is then a linear $[n, k, d]$ code. Furthermore, by combining Fact 4 and Fact 6, this code can be exactly generated by a simple sampler $\mathbf{X} \sim \{0,1\}^n$ of width $w \leq 2^{\lceil \frac{k}{t} \rceil} \leq 2 \cdot 2^{2C \cdot \frac{kd}{n}}$.

Thus all that remains for this case is to show the existence of a collection of codes $\{Q_i\}_{i \in [t]}$ with the above mentioned properties. For this, it suffices to show the existence of a linear $[n', k', d']$ code, where

$$\begin{aligned} n' &= Cd, \\ k' &= \left\lceil \frac{k}{t} \right\rceil = \left\lceil \frac{k}{\lfloor \frac{n}{Cd} \rfloor} \right\rceil, \\ d' &= d. \end{aligned}$$

By the Gilbert-Varshamov bound (Theorem 5), such a code exists as long as $2^{k'} \leq 2^{n'}/\binom{n'}{\leq d'-1}$. Plugging in the above values for n', k', d', a straightforward calculation (using the case condition that $k/n \leq 0.9$) shows that such a code must exist whenever $C \geq 250$. Thus we can always find a linear $[n, k, d]$ code $Q \subseteq \{0,1\}^n$ such that the uniform distribution $\mathbf{Q}$ over Q can be sampled in width $w \leq 2 \cdot 2^{5000 \cdot kd/n}$.

Case: $k > 0.9n$: In this case, we will show that there exists a linear $[n, k, d]$ code $Q \subseteq \{0,1\}^n$ such that the uniform distribution $\mathbf{Q}$ over Q can be sampled in width $w \leq 2^{C \cdot \frac{kd}{n} \log n}$ for a sufficiently large constant C. To construct Q, the general idea will be to start with a code Q' of dimension $k' \gg k$, show that membership in Q' can be checked by a small width ROBP (by keeping track of parity checks), and then convert this into a simple sampler via Theorem 10. Then, it will not be too difficult to reduce the dimension of Q' to match the target dimension k, while barely affecting the width of the sampler.

In more detail, let $k' := n - \lceil 4d \log n \rceil$. We consider the subcases $k \geq k'$ and $k < k'$. We first consider the easier subcase $k \geq k'$. By the theorem hypothesis, we know that there is a linear $[n, k, d]$ code $Q \subseteq \{0,1\}^n$. Let $Q^\perp$ denote its dual, and recall that $Q^\perp$ must therefore have dimension $n - k$. In other words, we can find a basis $v^{(1)}, \ldots, v^{(n-k)}$ of $Q^\perp$. Now, define a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ such that $f(x) = 1$ if and only if $\langle x, v^{(i)} \rangle = 0$ over $\mathbb{F}_2$, for all $i \in [n-k]$. That is, f accepts exactly the strings in $(Q^\perp)^\perp = Q$.

We can now design a low width ROBP that computes f. To do so, the ROBP keeps as its state a string $s \in \mathbb{F}_2^{n-k}$, which is originally initialized to the all zeroes vector. Upon reading a bit $x_i \in \{0,1\}$, the ROBP considers a vector $u^{(i)} \in \mathbb{F}_2^{n-k}$ which consists of the i^{th} bit of each of $v^{(1)}, \ldots, v^{(n-k)}$ concatenated together. Then, the ROBP transitions to state $s + x_i \cdot u^{(i)}$. In its final layer, the ROBP treats the all zeroes state as the accept state, and every other state as a reject state.

It is straightforward to verify that the above ROBP has width 2^{n-k}, and that the state $s \in \mathbb{F}_2^{n-k}$ it reaches in the final layer is exactly $(\langle x, v^{(1)} \rangle, \ldots, \langle x, y^{(n-k)} \rangle)$. Since the ROBP accepts if and only if this is the all zeroes vector, we see that the ROBP exactly computes f. And since $f^{-1}(1) = Q$, we can apply Theorem 10 to get a simple sampler of width $w = 2^{n-k}$ that samples the uniform distribution $\mathbf{Q}$ over Q. Since we have $k \geq k' = n - \lceil 4d \log n \rceil$, we know $n - k \leq \lceil 4d \log n \rceil \leq 5d \log n < 2 \cdot \frac{k}{n} \cdot 5d \log n$, where the last inequality follows from the case condition $k/n > 0.9$. Thus our simple sampler for our $[n, k, d]$ code $\mathbf{Q}$ has width $w < 2^{10 \cdot \frac{kd}{n} \log n}$, as desired.

We now consider the subcase $k < k'$. Let $t := k' - k \geq 1$. By the Gilbert-Varshamov bound (Theorem 5), there must exist a linear $[n, k', d]$ code $Q' \subseteq \mathbb{F}_2^n$. As we have seen above, Q' is easy to sample. But we would like to sample an $[n, k, d]$ code using approximately the same width. To do so, we will find a subcode of Q' that is easy to sample.

Since Q' is a linear $[n, k', d]$ code, it must have some vector q of Hamming weight d. Without loss of generality, we may assume the first d coordinates of q are 1, and the last $n - d$ coordinates are 0. Consider now the orthogonal complement S_q of $\{0, q\}$. Note that S_q has dimension $n - 1$. Furthermore, consider defining so-called "augmented elementary basis vectors" $\{\hat{e}^{(i)}\}_{i \in [n], i \neq d}$ as follows: for each $i < d$, let $\hat{e}^{(i)} := e^{(i)} + e^{(i+1)}$, and for each $i > d$, let $\hat{e}^{(i)} := e^{(i)}$, where each $e^{(i)} \in \mathbb{F}_2^n$ denotes a standard elementary basis vector. Then $\{\hat{e}^{(i)}\}$ is a basis for S_q.

Now let $v^{(1)}, \ldots, v^{(n-k')}$ be an arbitrary basis for the orthogonal complement $(Q')^\perp$. By straightforward linear algebra, there must be at least $k' - 1$ vectors in $\{\hat{e}^{(i)}\}_i$ that are mutually independent with $v^{(1)}, \ldots, v^{(n-k')}$. Without loss of generality, assume they are $\hat{e}^{(1)}, \ldots, \hat{e}^{(k'-1)}$. Notice now that $t \leq k' - 1$ (since we may assume $k \geq 1$), and consider the subspace $\widetilde{Q}$ spanned by basis vectors $v^{(1)}, \ldots, v^{(n-k')}, \hat{e}^{(1)}, \ldots, \hat{e}^{(t)}$. Observe that $\widetilde{Q}$ has dimension $n - k' + t = n - k$.

Finally, let $Q^* := \widetilde{Q}^\perp$. Notice that Q^* has dimension $n - (n - k) = k$ and that Q^* is a subspace

of $((Q')^{\perp})^{\perp} = Q'$. Thus Q^* has minimum distance $\geq d$. In fact, by our basis selection for $\widetilde{Q}$, it is straightforward to verify that the Hamming-weight d vector q defined earlier is also in Q^* (since q has inner product 0 with all the basis vectors of $\widetilde{Q}$). Thus, Q^* has minimum distance exactly d and it is therefore a linear $[n, k, d]$ code. Thus all that remains is to show that the uniform distribution over Q^* can be sampled by a low width simple sampler.

As in the first case of this proof, let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be defined such that $f(x) = 1$ if and only if $\langle x, v \rangle = 0$ for all $v \in \{v^{(1)}, \ldots, v^{(n-k')}, \hat{e}^{(1)}, \ldots, \hat{e}^{(t)}\}$. In other words, f tests membership in Q^*. Thus, if there is an ROBP of width w that computes f, then there is a simple sampler of width w that samples the uniform distribution $\mathbf{Q}$ over Q^*, by Theorem 10.

We now design a low width ROBP that computes f as follows. The state space of the ROBP will be $\mathbb{F}_2^{n-k'} \times \{0,1\} \times \{0,1\}$, and it will start at state $s = (\vec{0}, 0, 0)$. Informally, the first part of the state will keep track of the parity checks $v^{(1)}, \ldots, v^{(n-k')}$, while the remaining two parts will keep track of the parity checks $\hat{e}^{(1)}, \ldots, \hat{e}^{(t)}$ through a more compressed representation (which is enabled by the fact that these basis vectors each have Hamming weight at most 2).

More formally, suppose the ROBP is reading the string $x \in \{0,1\}^n$, and at time $i-1$ it arrive at state $(z, b, c) \in \mathbb{F}_2^{n-k'} \times \{0,1\} \times \{0,1\}$. Upon reading the next bit x_i, the ROBP will transition to the state (z', b', c'), defined as follows. First, let $u^{(i)} \in \mathbb{F}_2^{n-k'}$ be the string that consists of the i^{th} bit of each of $v^{(1)}, \ldots, v^{(n-k')}$. Then, define $z' := z + x_i \cdot u^{(i)}$.

Next, define $b' := x_i$.

Finally, we define c' as follows. If $c = 1$ then keep c' as 1 (this indicates one of the parity checks $\hat{e}^{(1)}, \ldots, \hat{e}^{(t)}$ has already been violated). If $i = 1$ then keep c' as 0. If $1 < i \leq d$ then set $c = 1$ if and only if $b \neq x_i$ (since this means $\langle x, \hat{e}^{(i-1)} \rangle = 1$). And if $d < i \leq n$, set $c = x_i$ (since $\langle x, \hat{e}^{(i)} \rangle = x_i$).

In the last layer of the ROBP, let all zeroes string $(\vec{0}, 0, 0)$ be the accept state, and every other string in $\mathbb{F}_2^{n-k'} \times \{0,1\} \times \{0,1\}$ be a reject state. It is straightforward to verify that the accept state is hit if and only if x has inner product 0 with each of $v^{(1)}, \ldots, v^{(n-k')}, \hat{e}^{(1)}, \ldots, \hat{e}^{(t)}$. Furthermore, this ROBP has width $w = |\mathbb{F}_2^{n-k'} \times \{0,1\} \times \{0,1\}| = 2^{n-k'+2} = 2^{\lceil 4d \log n \rceil + 2} < 2^{2\frac{k}{n} \cdot 4d \log n + 3}$, where the last inequality follows from the case condition $k/n > 0.9$. Thus by Theorem 10, there is a simple sampler of width $w < 8 \cdot 2^{8 \cdot \frac{kd}{n} \log n}$ that samples the uniform distribution $\mathbf{Q}$ over the linear $[n, k, d]$ code Q^*, as desired. □

7.2 An extension to unknown-order samplers

We now show that our general sampling lower bounds against list-decodable codes (Theorem 22) can be easily extended to the unknown-order setting. In particular, we prove the following:

Theorem 25. *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over a (ρ, L) list decodable code of dimension k. Then for any unknown-order complex sampler $\mathbf{X} \sim \{0,1\}^n$ of width w,*

$$|\mathbf{X} - \mathbf{Q}| \geq 1 - 4wL \cdot 2^{-\frac{\rho}{1+2\rho}k}.$$

Proof. By definition of unknown-order complex sampler, there exists a complex sampler $\mathbf{Y} \sim \{0,1\}^n$ and a permutation $\pi : [n] \to [n]$ such that $\mathbf{X} = \mathbf{Y}^{\pi}$. Thus by the data-processing inequality (Fact 1) we have

$$|\mathbf{X} - \mathbf{Q}| = |\mathbf{Y}^{\pi} - \mathbf{Q}| \geq |\mathbf{Y} - \mathbf{Q}^{\pi^{-1}}|.$$

It is easy to verify that permuting the coordinates of a code does not change its list-decodability. Thus $\mathbf{Q}^{\pi^{-1}}$ is uniform over a (ρ, L) list decodable code of dimension k. And since $\mathbf{Y}$ is a complex sampler, the result follows immediately from Theorem 22. □

7.3 Corresponding results for ROBPs

In this section, we briefly show how to combine our equivalence theorems with the above results in order to obtain our third main result: sampling lower bounds against codes for ROBPs.

Theorem 26 (Theorem 3, restated)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over a (ρ, L) list decodable code of dimension k. Then for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 8wL \cdot 2^{-\frac{\rho}{1+2\rho}k}.$$

Proof. Combine Theorem 22 with the first bullet of Theorem 6. □

Furthermore, we have seen that it is trivial to extend this to also hold for unknown-order ROBPs. Next, by Fact 2, we obtain the following specialization of Theorem 26:

Corollary 11. *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over an (n,k,d) code of dimension k. Then for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 12w \cdot 2^{-\frac{kd}{4n}}.$$

In particular, for any good code $\mathbf{Q} \sim \{0,1\}^n$ and ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width $2^{\Omega(n)}$, it holds that $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 2^{-\Omega(n)}$. Furthermore, recall that our sampling lower bounds for complex samplers against (n,k,d) codes are almost tight (Theorem 24). In fact, the tightness is actually witnessed by a simple sampler that is 2^{-n}-granular. Thus, combining Theorem 24 with Lemma 4, we obtain the following.

Remark 3. *Corollary 11 is almost tight: for all $n,k,d \in \mathbb{N}$ such that there exists a linear $[n,k,d]$ code, there exists a distribution $\mathbf{Q} \sim \{0,1\}^n$ uniform over a linear $[n,k,d]$ and an ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width $w \leq 2^{O(\frac{kd}{n}\log n)}$ and length $\ell = n^2$ such that $F(\mathbf{U}_\ell) = \mathbf{Q}$.*

We now consider some applications of these sampling lower bounds. Using one direction of one of our equivalence theorems (Theorem 11), we show that ROBPs of exponential width cannot test membership of a good code. In fact, we prove something stronger, and show the following covariance bounds (see Definition 4 for a definition of covariance).

Corollary 12. *Let $b : \{0,1\}^n \to \{0,1\}$ be the indicator function of a good code $Q \subseteq \{0,1\}^n$. Then for any ROBP $F : \{0,1\}^n \to \{0,1\}$ of width $2^{\Omega(n)}$, it holds that $|\operatorname{cov}(F,b)| \leq 2^{-\Omega(n)}$.*

In fact, note that we can get the exact same covariance bounds for unknown-order ROBPs by combining our unknown-order sampling lower bounds with Theorem 12. Next, we show how our sampling lower bounds against codes yield data structure lower bounds against storing codewords succinctly and retrieving them using ROBPs.

7.4 An application to data structure lower bounds

Just like in the previous works on sampling lower bounds against codes [LV12, BIL12], we can use an observation of Viola [Vio12a] to get data structure lower bounds against storing codewords. In particular, we obtain the following:

Corollary 13 (Data structure lower bounds)**.** *Let $Q \subseteq \{0,1\}^n$ be a (ρ, L) list decodable code of dimension k. Suppose that we can store the codewords of Q using only $k + r$ bits so that a codeword can be computed by an ROBP $F : \{0,1\}^{k+r} \to \{0,1\}^n$ of width w. Then*

$$r \geq \frac{\rho}{1+2\rho} \cdot k - \log(wL) - 3.$$

We repeat the proof of Viola [Vio12a] (see also [LV12]), using ROBPs instead of circuits.

Proof of Corollary 13. Suppose the codewords of Q can be stored in $\{0,1\}^{k+r}$ bits of memory so that they can be retrieved by some ROBP $F : \{0,1\}^{k+r} \to \{0,1\}^n$ of width w. Let $\mathbf{Q}$ be uniform over Q, and let $\mathbf{U}_{k+r}$ be uniform over $\{0,1\}^{k+r}$, and observe that

$$|F(\mathbf{U}_{k+r}) - \mathbf{Q}| \leq 1 - 2^{-r}$$

by a simple calculation using the definition of statistical distance. But by Theorem 26, we know that

$$1 - 8wL \cdot 2^{-\frac{\rho}{1+2\rho}k} \leq |F(\mathbf{U}_{k+r}) - \mathbf{Q}|.$$

Combining the bounds yields the result. □

For (n, k, d) codes, the above result specializes to the following.

Corollary 14. *Let $Q \subseteq \{0,1\}^n$ be an (n, k, d) code. Suppose that we can store codewords using only $k + r$ bits so that a codeword can be computed by an ROBP $F : \{0,1\}^{k+r} \to \{0,1\}^n$ of width $2^{\Omega(\frac{dk}{n})}$. Then*

$$r \geq \Omega\left(\frac{dk}{n}\right).$$

Thus for good codes, one must use $r = \Omega(n)$ bits of redundancy, even given an ROBP of width $2^{\Omega(n)}$.

8 A direct product theorem

In this section, we present our direct product theorems. In particular, we prove the following.

Theorem 27. *Let $\mathbf{Q} \sim \{0,1\}^n$ be a distribution such that for any complex sampler $\mathbf{X} \sim \{0,1\}^n$ of width w, it holds that $|\mathbf{X} - \mathbf{Q}| \geq \delta$. Then for any complex sampler $\mathbf{X}^* \sim \{0,1\}^{nt}$ of width w,*

$$|\mathbf{X}^* - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/2}.$$

In our proof of this theorem, *we will also show that the above direct product theorem holds for simple samplers*. In Section 8.1, we will show how to combine Theorem 27 with our equivalence theorems in order to obtain the corresponding result for sampling with ROBPs. In particular, we will prove Theorem 4 and show how it can be used to obtain our complexity separation between sampling with ROBPs and sampling with AC^0 circuits.

Now, before we prove Theorem 27, we introduce the main lemma that we will use in its proof.

Lemma 10 (Main lemma for Theorem 27)**.** *Let $\mathbf{X} \sim V^n$ and $\mathbf{Y} \sim V^n$ each be a sequence of n random variables over V, where elements in the sequence need not be independent. Suppose that for any $i \in [n]$ and $v \in V^{i-1}$,*

$$|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)| \geq \delta.$$

Then

$$|\mathbf{X} - \mathbf{Y}| \geq 1 - e^{-n\delta^2/2}.$$

It turns out that this lemma will be a little cumbersome to prove using statistical distance. Thus, we will convert the statement into one about more amenable notions of distance. The first such measure we consider is the squared Hellinger distance.

Definition 14 (Hellinger distance)**.** *Let $\mathbf{X}, \mathbf{Y}$ be random variables over some discrete V. The squared Hellinger distance between $\mathbf{X}, \mathbf{Y}$ is*

$$H^2(\mathbf{X}, \mathbf{Y}) = \frac{1}{2} \sum_{v \in V} \left(\sqrt{\Pr[\mathbf{X} = v]} - \sqrt{\mathbf{Y} = v} \right)^2$$

We would now like to express Lemma 10 using this more well-behaved measure of distance. For this, we will need the following fact, which is well-known and straightforward to show using Cauchy-Schwarz. It gives us estimates on statistical distance in terms of squared Hellinger distance.

Fact 9. *For any discrete random variables $\mathbf{X}, \mathbf{Y} \sim V$,*

$$H^2(\mathbf{X}, \mathbf{Y}) \leq |\mathbf{X} - \mathbf{Y}| \leq \sqrt{2} H(\mathbf{X}, \mathbf{Y}).$$

We now have the tools necessary to convert Lemma 10 into a statement about squared Hellinger distance. However, it turns out that there is another related measure of distance/similarity that will make Lemma 10 even easier to prove. It is known as the Bhattacharyya coefficient:

Definition 15 (Bhattacharyya coefficient)**.** *Let $\mathbf{X}, \mathbf{Y}$ be random variables over some discrete V. The Bhattacharyya coefficient between $\mathbf{X}$ and $\mathbf{Y}$ is defined as*

$$\mathsf{BC}(\mathbf{X}, \mathbf{Y}) := \sum_{v \in V} \sqrt{\Pr[\mathbf{X} = v] \cdot \Pr[\mathbf{Y} = v]}.$$

It is easy to verify via Definitions 14 and 15 that $1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y}) = H^2(\mathbf{X}, \mathbf{Y})$. We can then combine this observation with Fact 9 to obtain the following estimates on statistical distance in terms of the Bhattacharyya coefficient.

Fact 10. *For any discrete random variables $\mathbf{X}, \mathbf{Y} \sim V$,*

$$1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y}) \leq |\mathbf{X} - \mathbf{Y}| \leq \sqrt{2} \sqrt{1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y})}.$$

The goal now is to prove a version of Lemma 10 that uses the Bhattacharyya coefficient. We will then combine this result with Fact 10 to finally prove Lemma 10, which we then use to prove Theorem 27.

Lemma 11. *Let $\mathbf{X} \sim V^n$ and $\mathbf{Y} \sim V^n$ each be a sequence of n random variables over V, where elements in the sequence need not be independent. Suppose that for any $i \in [n]$ and $v \in V^{i-1}$,*

$$\mathsf{BC}((\mathbf{X}_i \mid \mathbf{X}_{<i} = v), (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)) \leq \delta.$$

Then

$$\mathsf{BC}(\mathbf{X}, \mathbf{Y}) \leq \delta^n.$$

Proof. We prove the slightly stronger statement that for any $i \in [n]$, it holds that $\mathsf{BC}(\mathbf{X}_{\leq i}, \mathbf{Y}_{\leq i}) \leq \delta^i$. We prove the result by induction on i. The base case $i = 1$ is immediate from the hypothesis. For the case $i \geq 2$ we have

$$\begin{aligned}
\mathsf{BC}(\mathbf{X}_{\leq i}, \mathbf{Y}_{\leq i}) &= \sum_{v \in V^i} \sqrt{\Pr[\mathbf{X}_{\leq i} = v] \Pr[\mathbf{Y}_{\leq i} = v]} \\
&= \sum_{v \in V^{i-1}} \sum_{b \in V} \sqrt{\Pr[\mathbf{X}_{<i} = v] \Pr[\mathbf{X}_i = b \mid \mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_{<i} = v] \Pr[\mathbf{Y}_i = b \mid \mathbf{Y}_{<i} = v]} \\
&= \sum_{v \in V^{i-1}} \sqrt{\Pr[\mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_{<i} = v]} \sum_{b \in V} \sqrt{\Pr[\mathbf{X}_i = b \mid \mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_i = b \mid \mathbf{Y}_{<i} = v]} \\
&= \sum_{v \in V^{i-1}} \sqrt{\Pr[\mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_{<i} = v]} \cdot \mathsf{BC}((\mathbf{X}_i \mid \mathbf{X}_{<i} = v), (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)) \\
&\leq \delta \cdot \sum_{v \in V^{i-1}} \sqrt{\Pr[\mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_{<i} = v]} \\
&= \delta \cdot \mathsf{BC}(\mathbf{X}_{<i}, \mathbf{Y}_{<i}) \\
&\leq \delta \cdot \delta^{i-1} \\
&= \delta^i,
\end{aligned}$$

where the inequalities use the lemma hypothesis and the induction hypothesis. □

We now combine the above lemma with our estimates from Fact 10 to prove Lemma 10.

Proof of Lemma 10. By combining the lemma hypothesis with Fact 10, we know that for any $i \in [n]$ and $v \in V^{i-1}$,

$$\begin{aligned}
\mathsf{BC}((\mathbf{X}_i \mid \mathbf{X}_{<i} = v), (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)) &\leq 1 - \frac{|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)|^2}{2} \\
&\leq 1 - \delta^2/2.
\end{aligned}$$

Thus by Lemma 11 we have

$$\mathsf{BC}(\mathbf{X}, \mathbf{Y}) \leq (1 - \delta^2/2)^n \leq e^{-n\delta^2/2},$$

where we use the standard inequality $1 + x \leq e^x$ for all real x. Using Fact 10 once more we get

$$|\mathbf{X} - \mathbf{Y}| \geq 1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y}) \geq 1 - e^{-n\delta^2/2},$$

as desired. □

At last, we are almost ready to apply Lemma 10 to prove our main theorem of the section. But just before we do so, we need the following two facts, which are both straightforward to verify.

Fact 11. *Let $\mathbf{X} \sim \{0,1\}^n$ be a complex sampler of width w. Then for any $1 \leq i \leq j \leq n$ and any $x \in \{0,1\}^{i-1}$, the distribution*

$$(\mathbf{X}_{i \to j} \mid \mathbf{X}_{<i} = x)$$

is also a complex sampler of width w.

Fact 12. *Let $\mathbf{X} \sim \{0,1\}^n$ be a simple sampler of width w. Then for any $1 \le i \le j \le n$ and any $x \in \{0,1\}^{i-1}$, the distribution*

$$(\mathbf{X}_{i\to j} \mid \mathbf{X}_{<i} = x)$$

is also a simple sampler of width w.

Finally, we are ready to prove Theorem 27, and thereby obtain our direct product theorems for simple and complex samplers.

Proof of Theorem 27. Parse $\mathbf{X}^*$ as $(\mathbf{X}_1, \dots, \mathbf{X}_t) \sim (\{0,1\}^n)^t$, and parse $\mathbf{Q}^{\otimes t}$ as $(\mathbf{Q}_1, \dots, \mathbf{Q}_t) \sim (\{0,1\}^n)^t$. By definition, each $\mathbf{Q}_i$ is an independent copy of $\mathbf{Q}$. And by Fact 11 or Fact 12 (depending on whether we want to prove the result for complex or simple samplers), we know that for any fixing of $\mathbf{X}_{<i}$, the random variable $\mathbf{X}_i$ is a sampler of the same type, and it has width at most w. Thus for any $i \in [t]$ and $v \in (\{0,1\}^n)^{i-1}$, we have

$$|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = v)| \ge \delta$$

by the hypothesis $|\mathbf{X} - \mathbf{Q}| \ge \delta$. Applying Lemma 10 completes the proof. □

8.1 Corresponding results for ROBPs

In this section, we will show how to apply the tools from above to obtain our main direct product theorem, for sampling with ROBPs (Theorem 4). Then, we will show how it can be used to obtain a strong separation between sampling with ROBPs and sampling with AC^0 circuits (Corollary 6). We begin with the direct product theorem.

Theorem 28 (Theorem 4, restated)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be a distribution such that for any $\ell \in \mathbb{N}$ and any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w, it holds that $|F(\mathbf{U}_\ell) - \mathbf{Q}| \ge \delta$. Then for any $t, \ell^* \in \mathbb{N}$ and any ROBP $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^{nt}$ of width w, it holds that*

$$|F^*(\mathbf{U}_{\ell^*}) - \mathbf{Q}^{\otimes t}| \ge 1 - e^{-t\delta^2/4}.$$

In order to prove this result, we would like to use our equivalence theorems and the direct product theorem for complex samplers (from the previous section). Such a proof would look roughly as follows: first, if every ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w has $|F(\mathbf{U}_\ell) - \mathbf{Q}| \ge \delta$, then by the second bullet of Theorem 6, every complex sampler $\mathbf{X} \sim \{0,1\}^n$ of width $w/7$ has roughly $|\mathbf{X} - \mathbf{Q}| \ge \delta$. Then, by Theorem 27, we know that every complex sampler $\mathbf{X}^* \sim \{0,1\}^{nt}$ of width $w/7$ has roughly $|\mathbf{X}^* - \mathbf{Q}^{\otimes t}| \ge 1 - e^{-t\delta^2/4}$. Finally, by the first bullet of Theorem 6, every ROBP $F : \{0,1\}^{\ell^*} \to \{0,1\}^{nt}$ of width $w/14$ must have $|F^*(\mathbf{U}_{\ell^*}) - \mathbf{Q}^{\otimes t}| \ge 1 - e^{-t\delta^2/4}$.

The above argument will give us statistical distance lower bounds of the form that we would like, but we lose a factor of 14 in the width. We would like to avoid this, and keep the direct product theorem *strong*, in the sense that the width need not decrease at all. Towards this end, we will prove a version of Fact 11 for sampling using ROBPs. In particular, we prove the following.

Claim 1. *Let $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^n$ be an ROBP of width w, and define $\mathbf{X} := F^*(\mathbf{U}_{\ell^*})$. For any $1 \le i \le j \le n$ and any $x \in \{0,1\}^{i-1}$, and any $\varepsilon > 0$, the following holds. There exists an ROBP $F : \{0,1\}^\ell \to \{0,1\}^{j-i+1}$ of width w and length $\ell = \ell^* + 3w\log(w/\varepsilon)$ such that*

$$|F(\mathbf{U}_\ell) - (\mathbf{X}_{i\to j} \mid \mathbf{X}_{<i} = x)| \le \varepsilon.$$

A natural approach towards proving Claim 1 is to build the ROBP $F : \{0,1\}^\ell \to \{0,1\}^{j-i+1}$ by taking the appropriate slice of the ROBP F^* and simulating the conditioning $\mathbf{X}_{<i} = x$ using a multi-threshold function, as in Section 4.1. However, our ROBP from that section for computing this multi-threshold function required width $2w$, and we showed that this is tight. To simulate this conditioning using width just w, we need a new idea. Before presenting this new idea and proving Claim 1, let us see how it can be used to prove Theorem 28.

Proof of Theorem 28. Let $\mathbf{X}^* = F^*(\mathbf{U}_{\ell^*})$. Parse $\mathbf{X}^*$ as $(\mathbf{X}_1, \dots, \mathbf{X}_t) \sim (\{0,1\}^n)^t$, and parse $\mathbf{Q}^{\otimes t}$ as $(\mathbf{Q}_1, \dots, \mathbf{Q}_t) \sim (\{0,1\}^n)^t$. Now, fix any $i \in [t]$ and $v \in (\{0,1\}^n)^{i-1}$. We want to get a lower bound on $|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = v)|$. Towards this end, set $\varepsilon := \delta \cdot (1 - 1/\sqrt{2})$. By Claim 1, we know that there exists an ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w and length $\ell = \ell^* + 3w \log(\frac{\sqrt{2}w}{(\sqrt{2}-1)\delta})$ such that

$$|F(\mathbf{U}_\ell) - (\mathbf{X}_i \mid \mathbf{X}_{<i} = v)| \le \delta \cdot (1 - 1/\sqrt{2})$$

Furthermore, by the theorem hypothesis, we know that $|F(\mathbf{U}_\ell) - \mathbf{Q}_i| \ge \delta$. Thus by the triangle inequality we have $|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - \mathbf{Q}_i| \ge \delta/\sqrt{2}$. And since each $\mathbf{Q}_i$ is an independent copy of $\mathbf{Q}$, we of course have $\mathbf{Q}_i = (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = v)$ and thus

$$|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = v)| \ge \delta/\sqrt{2}.$$

By applying Lemma 10, we immediately get that $|\mathbf{X}^* - \mathbf{Q}^{\otimes t}| \ge 1 - e^{-t\delta^2/4}$, which completes the proof. Note that we can actually make the constant 4 arbitrarily close to 2 by picking a small enough ε, since the length $\ell = \ell^* + 3w \log(w/\varepsilon)$ of the ROBP F does not matter. □

Our goal now is to prove Claim 1. The new main ingredient we will use is a Σ-ROBP that can achieve roughly the same sampling task that is achieved by our Σ-ROBP for the multi-thresholding function from Section 4.1, but using less width. In particular, we prove the following.

Lemma 12 (Key ingredient for Theorem 28)**.** *For any distribution $\mathbf{X} \sim [w]$ and $\varepsilon > 0$, there exists a Σ-ROBP $f : \{0,1\}^\ell \to [w]$ of width w and length $\ell = 3w \log(w/\varepsilon)$ such that*

$$|f(\mathbf{U}_\ell) - \mathbf{X}| \le \varepsilon.$$

Before we prove Lemma 12, let us see how we can use it to show Claim 1.

Proof of Claim 1. Let $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^n$ be an ROBP of width w, and define $\mathbf{X} := F^*(\mathbf{U}_{\ell^*})$. Let $G = (V, E)$ be the graph underlying this ROBP with layers $V = V_0 \cup V_1 \cup \cdots \cup V_{\ell^*}$. Let γ_1 denote the number of output bits labeling each edge into V_1, let γ_2 denote the number of output bits labeling each edge into V_2, and so on. Note that $\sum_{i \in [\ell^*]} \gamma_i = n$.

Now, fix any $1 \le i \le j \le n$ and $x \in \{0,1\}^{i-1}$ and $\varepsilon > 0$. The claim is easy to show if $i = 1$, so we henceforth assume $i > 1$. Recall that the goal is to construct an ROBP F such that $|F(\mathbf{U}_\ell) - (\mathbf{X}_{i \to j} \mid \mathbf{X}_{<i} = x)|$ is small. Towards this end, let $\alpha \le \beta \in [\ell^*]$ be such that the i^{th} bit of $\mathbf{X}$ is outputted upon entering layer V_α, and the j^{th} bit of $\mathbf{X}$ is outputted upon entering layer V_β. That is, α is the smallest integer such that $\sum_{h \in [\alpha]} \gamma_h \ge i$ and β is the smallest integer such that $\sum_{h \in [\beta]} \gamma_h \ge j$.

We now construct a new branching program $G' = (V', E')$ that will eventually help us construct F. The layers of this new branching program are $V' = V'_{\alpha-1} \cup V'_\alpha \cup \cdots \cup V'_\beta$, where each V'_i is a copy of V_i from the original branching program G. The edge set E' of the new branching program will include all the edges from E that traverse between these layers (including their input and output labels), and nothing more.

We now perform a slight modification to the edges in E'. First, recall that by definition of each γ_h, we have $\sum_{h\in[\alpha-1]}\gamma_h \le i-1 < \sum_{h\in[\alpha]}\gamma_h$. Let $r := (i-1) - \sum_{h\in[\alpha-1]}\gamma_h \ge 0$ denote the number of bits on the edges into layer V'_α that will eventually by fixed by fixing $\mathbf{X}_{<i} = x$. Now, for each $v \in V'_{\alpha-1}$, do the following: consider its outgoing edges e_0 and e_1 with input labels 0 and 1, respectively. We will now examine whether each of these edges have the property that the first r bits in their output label match the last r bits of x. If neither of e_0, e_1 have this property, do nothing. If both of e_0, e_1 have this property, do nothing. If e_0 has this property but e_1 does not, then delete e_1 and replace it with a new copy of e_0 (this copy should connect the same vertices as e_0, it should have the same output label as e_0, but it should have the input label 1). If e_1 has this property but e_0 does not, then delete e_0 and replace it with a new copy of e_1 (this copy should connect the same vertices as e_1, it should have the same output label as e_1, but it should have the input label 0).

Our last modification to the edges of E' will be as follows. First, let $r' := \sum_{h\in[\beta]}\gamma_h - j$. We will erase the first r bits and last r' bits output by G'. In particular, for every edge entering V'_α, delete the first r bits of its output label. And for every edge entering V'_β, delete the last r' bits of its output label.

Now, label the vertices in $V_{\alpha-1}$ as $v_1, \dots, v_w$, and let $v'_1, \dots, v'_w$ denote the corresponding vertices in $V'_{\alpha-1}$. Consider again the original ROBP $G = (V, E)$. For each $s \in [w]$, let p_s denote the probability that a (uniform) random walk from the start vertex of G hits v_s, *conditioned on the event that the first $i-1$ bits output by the random walk exactly match x*. Next, consider the new ROBP G'. For each $s \in [w]$, let $\mathbf{Y}_s \sim \{0,1\}^{j-i+1}$ denote the distribution generated by taking a (uniform) random walk over G', starting at vertex $v'_s \in V'_{\alpha-1}$ and outputting the output labels seen along the way. It is now straightforward to verify

$$\sum_{s\in[w]} p_s \cdot \mathbf{Y}_s = (\mathbf{X}_{i\to j} \mid \mathbf{X}_{<i} = x).$$

We would now like to construct a new ROBP that (almost) generates the distribution $\sum_{s\in[w]} p_s \cdot \mathbf{Y}_s$. First, let $\mathbf{A} \sim [w]$ denote the random variable corresponding to the distribution $\{p_s\}_{s\in[w]}$. By Lemma 12, there is a Σ-ROBP $f : \{0,1\}^{\ell'} \to [w]$ of width w and length $\ell' = 3w\log(w/\varepsilon)$ such that $|f(\mathbf{U}_{\ell'}) - \mathbf{A}| \le \varepsilon$. For each $s \in [w]$, let $q_s := \Pr[f(\mathbf{U}_{\ell'}) = s]$. Now, let $G'' = (V'', E'')$ denote its underlying graph, which has layers $V'' = V''_0 \cup V''_1 \cup \dots \cup V''_{\ell'}$. Label the vertices in $V''_{\ell'}$ as $u_1, \dots, u_w$, so that a random walk on G'' hits u_s with probability q_s.

We are finally ready to construct the ROBP $F : \{0,1\}^\ell \to \{0,1\}^{j-i+1}$ advertised in the claim statement. Its underlying graph $G_F = (V_F, E_F)$ is formed as follows: first, set $V_F = V''_0 \cup \dots \cup V''_{\ell'} \cup V'_{\alpha-1} \cup \dots \cup V'_\beta$. Then, set $E_F = E'' \cup E'$. Finally, merge layers $V_{\ell''}$ and $V'_{\alpha-1}$ by identifying each $u_s \in V_{\ell''}$ with $v'_s \in V'_{\alpha-1}$. It is straightforward to verify that $F(\mathbf{U}_\ell) = \sum_{s\in[w]} q_s \cdot \mathbf{Y}_s$. Furthermore, notice that F has width w and length $\ell = \ell' + \beta - \alpha + 1 \le \ell' + \ell^* = 3w\log(w/\varepsilon) + \ell^*$.

Thus, all that remains is to show $|\sum_{s\in[w]} q_s \cdot \mathbf{Y}_s - \sum_{s\in[w]} p_s \cdot \mathbf{Y}_s| \le \varepsilon$. Using the definition of statistical

distance, we have

$$\begin{aligned}
|\sum_s q_s \mathbf{Y}_s - \sum_s p_s \mathbf{Y}_s| &= \max_{T \subseteq \{0,1\}^{j-i+1}} (\Pr[\sum_s q_s \mathbf{Y}_s \in T] - \Pr[\sum_s p_s \mathbf{Y}_s \in T]) \\
&= \max_T \sum_s \Pr[\mathbf{Y}_s \in T] \cdot (q_s - p_s) \\
&\leq \max_T \sum_{s \,:\, q_s - p_s \geq 0} \Pr[\mathbf{Y}_s \in T] \cdot (q_s - p_s) \\
&\leq \sum_{s \,:\, q_s - p_s \geq 0} (q_s - p_s) \\
&= |f(\mathbf{U}_{\ell'}) - \mathbf{A}| \\
&\leq \varepsilon,
\end{aligned}$$

as desired. □

At last, all that remains is to prove Lemma 12. We do so, below.

Proof of Lemma 12. We would like to show that for any distribution $\mathbf{X} \sim [w]$ and $\varepsilon > 0$, there exists a Σ-ROBP $f : \{0,1\}^\ell \to [w]$ of width w and length $\ell = 3w \log(w/\varepsilon)$ such that $|f(\mathbf{U}_\ell) - \mathbf{X}| \leq \varepsilon$. Without loss of generality, we assume that $\Pr[\mathbf{X} = i] > 0$ for each $i \in [w]$.

First, suppose that for any distribution $\mathbf{Y} \sim \{0,1\}$ and $\varepsilon' > 0$, there exists an ROBP $f' : \{0,1\}^{\ell'} \to \{0,1\}$ of width 2 and length $\ell' = \lceil \log(1/\varepsilon') \rceil$ such that $|f'(\mathbf{U}_{\ell'}) - \mathbf{Y}| \leq \varepsilon'$. We will first show how this can be used to obtain the desired result, and then we will show how to construct such an ROBP.

Thus let us return to the original distribution $\mathbf{X} \sim [w]$ that we would like to sample with error ε. For each $i \in [w-1]$, define a random variable $\mathbf{A}_i \sim \{0,1\}$ as follows. We set $\mathbf{A}_i = 0$ with probability $\Pr[\mathbf{X} = i \mid \mathbf{X} \geq i]$ and we set $\mathbf{A}_i = 1$ with probability $\Pr[\mathbf{X} > i \mid \mathbf{X} \geq i]$. Observe that

$$\Pr[\mathbf{X} = i] = \begin{cases} \Pr[\mathbf{A}_i = 0] \cdot \prod_{j<i} \Pr[\mathbf{A}_j = 1] & \text{if } i < w, \\ \Pr[\mathbf{A}_{i-1} = 1] \cdot \prod_{j<i-1} \Pr[\mathbf{A}_j = 1] & \text{if } i = w. \end{cases} \tag{10}$$

Now, for every $i \in [w-1]$, let $f'_i : \{0,1\}^{\ell'} \to \{0,1\}$ be an ROBP of width 2 and length $\ell' = \lceil \log(1/\varepsilon') \rceil$ such that $|f'_i(\mathbf{U}_{\ell'}) - \mathbf{A}_i| \leq \varepsilon'$. For convenience, we let $\mathbf{B}_i := f'_i(\mathbf{U}_{\ell'})$ so that $|\mathbf{A}_i - \mathbf{B}_i| \leq \varepsilon'$. Now, let $G^i = (V^i, E^i)$ denote the underlying graph of f'_i with layers $V^i = V^i_0 \cup \cdots \cup V^i_{\ell'}$. Let $\mathsf{start}^i \in V^i_0$ denote its start state, let $\mathsf{accept}^i \in V^i_{\ell'}$ denote its accept state, and let $\mathsf{reject}^i \in V^i_{\ell'}$ denote its reject state.

The goal now is to combine these ROBPs into one large ROBP that computes $f : \{0,1\}^\ell \to [w]$. To do so, we construct its underlying graph $G = (V, E)$ as follows. First, we concatenate all of the above ROBPs in a series configuration. That is, we set

$$V = (V^1_0 \cup \cdots \cup V^1_{\ell'}) \cup (V^2_0 \cup \cdots \cup V^2_{\ell'}) \cup \cdots \cup (V^{w-1}_0 \cup \cdots \cup V^{w-1}_{\ell'}). \tag{11}$$

Next, we add all the edges $\bigcup_{i \in [w-1]} E^i$ to E. Then, for every $i \in [w-2]$, we draw two edges from accept^i to start^{i+1}, and give them input labels $0, 1$, respectively. Next, for every $i \in [w-2]$, we do the following: for every layer W appearing (strictly) to the right of layer $V^i_{\ell'}$ in Equation (11), add a node called bucket^i_W. Then, draw two edges (labeled 0 and 1) from reject^i to bucket^i_W, where W is the layer immediately following $V^i_{\ell'}$. Then, for any consecutive layers W, W' that contain nodes $\mathsf{bucket}^i_W, \mathsf{bucket}^i_{W'}$, draw two edges (labeled 0 and 1) from bucket^i_W to $\mathsf{bucket}^i_{W'}$. Finally, for each $i \in [w-2]$, give the node

$\mathsf{bucket}^i_{V^{w-1}_{\ell'}}$ the output label i; give the node reject^{w-1} the output label $w-1$; and give the node accept^{w-1} the output label w. This completes the construction of $G=(V,E)$ and $f:\{0,1\}^\ell \to [w]$.

Notice that the widest layer in G is $V^{w-1}_{\ell'}$, and it has width $2+(w-2)=w$. Thus, G has width w and length $\ell = \ell' + (1+\ell')(w-2)$. In fact, notice we can contract the "trivial" edges between layers $V^i_{\ell'}, V^{i+1}_0$ for every $i \in [w-2]$, without changing the output distribution, thereby making the overall length $\ell = \ell' \cdot (w-1) \leq \ell' \cdot w$.

Now, set $\mathbf{X}' = f(\mathbf{U}_\ell)$ and observe via the above construction that

$$\Pr[\mathbf{X}'=i] = \begin{cases} \Pr[\mathbf{B}_i=0] \cdot \prod_{j<i} \Pr[\mathbf{B}_j=1] & \text{if } i<w, \\ \Pr[\mathbf{B}_{i-1}=1] \cdot \prod_{j<i-1} \Pr[\mathbf{B}_j=1] & \text{if } i=w. \end{cases} \tag{12}$$

Our goal now is to upper bound $|\mathbf{X}'-\mathbf{X}|$ using Equations (10) and (12). Towards this end, suppose we have a sequence of probabilities $p_1,\dots,p_i$ and another sequence of probabilities $q_1,\dots,q_i$ such that each q_j is at most $p_j+\varepsilon'$. It is then straightforward to use a hybrid-type argument to verify that $q_1 \cdot q_2 \cdots q_i \leq p_1 \cdot p_2 \cdots p_i + i \cdot \varepsilon'$. Thus, recalling that each $|\mathbf{A}_i - \mathbf{B}_i| \leq \varepsilon'$, we know by Equations (10) and (12) that for every $i \in [w]$,

$$\Pr[\mathbf{X}'=i] - \Pr[\mathbf{X}=i] \leq w \cdot \varepsilon'.$$

Thus, we have

$$\begin{aligned} |\mathbf{X}'-\mathbf{X}| &= \max_{T\subseteq[w]} \Pr[\mathbf{X}'\in T] - \Pr[\mathbf{X}\in T] \\ &= \max_{T\subseteq[w]} \sum_{i\in T} (\Pr[\mathbf{X}'=i] - \Pr[\mathbf{X}=i]) \\ &\leq \max_{T\subseteq[w]} \sum_{i\in T} w\varepsilon' \\ &\leq w^2 \cdot \varepsilon'. \end{aligned}$$

Finally, we see that if we set $\varepsilon' = \varepsilon/w^2$, then we have a Σ-ROBP $f:\{0,1\}^\ell \to [w]$ of width w and length $\ell \leq \ell' \cdot w = \lceil \log(1/\varepsilon') \rceil \cdot w = \lceil \log(w^2/\varepsilon) \rceil \cdot w \leq 3w \log(w/\varepsilon)$ such that $|f(\mathbf{U}_\ell) - \mathbf{X}| \leq \varepsilon$, as desired.[10]

All that remains is to show the claim we assumed at the beginning of this proof. Namely, that for any distribution $\mathbf{Y} \sim \{0,1\}$ and $\varepsilon' > 0$, there exists an ROBP $f':\{0,1\}^{\ell'} \to \{0,1\}$ of width 2 and length $\ell' = \lceil \log(1/\varepsilon') \rceil$ such that $|f'(\mathbf{U}_{\ell'}) - \mathbf{Y}| \leq \varepsilon'$.

To prove the above, we specify the underlying graph $G'=(V',E')$ of f' as follows. The graph will consist of layers $V' = V'_0 \cup V'_1 \cup \cdots \cup V'_{\ell'}$, where each V'_i consists of vertices labeled u_i, v_i. We label u_0 as the start vertex, $u_{\ell'}$ as the reject state (outputs 0), and $v_{\ell'}$ as the accept state (outputs 1). Define $p := \Pr[\mathbf{Y}=0]$ and assume without loss of generality that $0<p<1$. Next, we specify the edges E' of G'.

Let $b \in \{0,1\}^{\ell'}$ denote a parameter that we will specify later. Using b, we construct the edges entering each layer V'_i as follows. For every $i \in [\ell']$, do the following: if $b_i = 0$, draw two parallel edges of the form (v_{i-1}, v_i) and give them input labels $0,1$; then, draw edges (u_{i-1}, u_i) with input label 0 and (u_{i-1}, v_i) with input label 1. On the other hand, if $b_i = 1$, then draw two parallel edges of the form (u_{i-1}, u_i) and give them input labels $0,1$; then, draw edges (v_{i-1}, v_i) with input label 0 and (v_{i-1}, u_i) with input label 1.

[10] We remark that the w inside the log can be removed through a slightly more technical construction, where the ROBPs f'_i's are selected to take into account the errors made by the earlier f'_i's.

Consider now a (uniform) random walk over G', starting at the start vertex u_0. For every $i \in [\ell']$, let q_i denote the probability that this random walk hits vertex u_i. Of course, the probability that the random walk hits v_i is then $1 - q_i$. Define $q_0 = 1$, and observe via our construction that the following holds for all $i \in [\ell']$:

$$q_i = \begin{cases} \frac{1}{2} \cdot q_{i-1} & \text{if } b_i = 0, \\ q_{i-1} + \frac{1}{2} \cdot (1 - q_{i-1}) = \frac{1}{2} \cdot (q_{i-1} + 1) & \text{if } b_i = 1, \end{cases}$$

which can be expressed more concisely as

$$q_i = \frac{1}{2} \cdot (q_{i-1} + b_i).$$

Recalling that $q_0 = 1$, it is then straightforward to show via induction that for any $i \in [\ell']$,

$$q_i = 2^{-i} + \sum_{h \in [i]} b_{i+1-h} \cdot 2^{-h}.$$

Since $u_{\ell'}$ is the reject state, we have

$$\Pr[f'(\mathbf{U}_{\ell'}) = 0] = q_{\ell'} = 2^{-\ell'} + \sum_{h \in [\ell']} b_{\ell'+1-h} \cdot 2^{-h}.$$

But observe that this quantity is simply a decimal written in binary using $\{b_1, b_2, \ldots, b_{\ell'}\}$. In particular, if we consider any $0 < \tau < 1$ that can be written as $\tau = 2^{-\ell'} \cdot K$ for some $K \in \mathbb{N}$, then we can always find some $b \in \{0,1\}^{\ell'}$ such that the right hand side above evaluates to exactly τ. Now, recall that $p = \Pr[\mathbf{Y} = 0]$. Pick the smallest $K \in \mathbb{N}$ such that $2^{-\ell'} K \geq p > 0$, and pick $b \in \{0,1\}^{\ell'}$ such that $\Pr[f'(\mathbf{U}_{\ell'}) = 0] = 2^{-\ell'} K$. Then we must have

$$\Pr[\mathbf{Y} = 0] \leq \Pr[f'(\mathbf{U}_{\ell'}) = 0] < \Pr[\mathbf{Y} = 0] + 2^{-\ell'}.$$

In other words,

$$|f'(\mathbf{U}_{\ell'}) - \mathbf{Y}| < 2^{-\ell'}. \tag{13}$$

Finally, recall that the ROBP we constructed for f' had width 2, and thus we may set $\ell' = \lceil \log(1/\varepsilon') \rceil$ to upper bound Equation (13) by ε' and complete the proof. □

A separation between sampling with ROBPs and sampling with AC^0 circuits Finally, we show how to apply our direct product theorem to obtain a strong complexity separation between ROBPs and AC^0 circuits for the task of sampling. We prove the following.

Theorem 29 (Corollary 6, restated)**.** *There is a universal constant $c > 0$ such that for all sufficiently large $n \in \mathbb{N}$, the following holds. There exists an AC^0 circuit $Q : \{0,1\}^\ell \to \{0,1\}^n$ such that for every ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width $w \leq 2^{c \cdot n / \log(1/\varepsilon)}$,*

$$|F(\mathbf{U}_\ell) - Q(\mathbf{U}_\ell)| \geq 1 - \varepsilon.$$

Proof. Let $k, t \in \mathbb{N}$ be parameters to be set later, such that $(2k+1)t = n$. By the proof of Theorem 20, we know that for the inner product function $\mathsf{IP} : \mathbb{F}_2^k \times \mathbb{F}_2^k \to \mathbb{F}_2$, and any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width 2^{ck} with $c > 0$ a sufficiently small constant, it holds that

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_{2k}, \mathsf{IP}(\mathbf{U}_{2k}))| \geq \frac{1}{2} - 2^{-ck}.$$

Now, define $\mathbf{Q} = (\mathbf{U}_{2k}, \mathsf{IP}(\mathbf{U}_{2k}))$ and consider the distribution $\mathbf{Q}^{\otimes t}$. By our direct product theorem (Theorem 28), we know that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width 2^{ck}, it holds that

$$|F(\mathbf{U}_\ell) - \mathbf{Q}^{\otimes t}| \geq 1 - 2^{-c't}.$$

However, by [IN96], we know that $\mathbf{Q}$ can be sampled by AC^0 circuits, and thus so can $\mathbf{Q}^{\otimes t}$ (simply by running several of these circuits in parallel). Thus, setting $t = O(\log(1/\varepsilon))$ and $k = ((n/t) - 1)/2$ completes the proof. □

Two particularly interesting settings of Theorem 29 are $\varepsilon = 0.01$ and $\varepsilon = 2^{-\sqrt{n}}$. In the first setting, we get a distribution $\mathbf{Q}$ samplable by AC^0 circuits but such that $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 0.99$ for any ROBP F of width $2^{\Omega(n)}$. In the latter setting, we get much stronger bounds of the form $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 2^{\Omega(-\sqrt{n})}$ for any ROBP F of width $2^{\Omega(\sqrt{n})}$.

9 Future directions

Recently, there have been a number of exciting works that study the complexity of sampling. In this area of complexity, one seeks to understand the power of classical computational models for the task of sampling from distributions. In this paper, we initiate a study of the complexity of sampling in limited *space*. We prove that ROBPs are more powerful for sampling than computing, yet we demonstrate explicit distributions that are very difficult to sample using ROBPs. Furthermore, we provide a direct product theorem for this setting, and use it to demonstrate a strong complexity separation between ROBPs and AC^0 circuits for the task of sampling.

As this area of complexity is very new, many open questions remain. Below, we outline a few such questions on the complexity of sampling with ROBPs.

A stronger separation between sampling and computing with ROBPs In this paper, we gave an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be sampled by ROBPs of width $O(n)$, but such that ROBPs of width $2^{\Omega(n)}$ fail to compute b on roughly $1/4$ if its inputs. Can one strengthen these average-case bounds? Namely, can one find a function $b : \{0,1\}^n \to \{0,1\}$ such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be sampled by ROBPs of linear width, but such that ROBPs of exponential width fail to compute b on roughly $1/2$ of its inputs?

Sampling lower bounds using limited randomness In this paper, we demonstrated distributions that cannot be sampled by ROBPs of limited width, given *unlimited* random bits as input. Can one demonstrate sampling lower bounds for ROBPs against a richer class of distributions if the ROBP is only provided with a limited number of random input bits (say, $\ell = \mathrm{poly}(n)$)?

This question is especially interesting if one considers a more powerful model of multi-output ROBPs whose output bits need not be layered (as in Definition 8). It can be shown that such ROBPs can come arbitrarily close to sampling *any distribution* $\mathbf{X} \sim \{0,1\}^n$ using just width 3. However, the most natural way

to construct such an ROBP requires $\ell \gg |\text{support}(\mathbf{X})|$ bits of randomness, which could be exponentially large in n. Thus, it would be interesting to understand the power of these ROBPs under the restriction $\ell \leq \text{poly}(n)$.

A separation between sampling with ROBPs and AC^0 circuits, in the other direction In this paper, we demonstrated a distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ that cannot be sampled by ROBPs (even on average), but can be sampled by AC^0 [IN96]. Can one find a distribution that cannot be sampled by AC^0, but can be sampled by ROBPs? This is possible if one can construct an extractor for AC^0 sources (distributions generated in AC^0), which can be computed by small width ROBPs. In fact, even a disperser $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ for min-entropy $n-1$ would suffice, as this would imply that both $\mathsf{Disp}^{-1}(1)$ and $\mathsf{Disp}^{-1}(0)$ can be generated in small space (via our equivalence theorems), yet one of these has min-entropy $n-1$ and thus cannot be generated in AC^0 (by definition of a disperser). Since all known extractors for AC^0 sources [Vio14] also work for distributions that can be generated in small space [CG20], new extractors are needed.

10 Acknowledgements

We thank William Hoza for helpful comments.

References

[Aar14] Scott Aaronson. The equivalence of sampling and searching. *Theory of Computing Systems*, 55(2):281–298, 2014.

[ASTS+03] Andris Ambainis, Leonard J Schulman, Amnon Ta-Shma, Umesh Vazirani, and Avi Wigderson. The quantum communication complexity of sampling. *SIAM Journal on Computing*, 32(6):1570–1585, 2003.

[Bab87] László Babai. Random oracles separate pspace from the polynomial-time hierarchy. *Information Processing Letters*, 26(1):51–53, 1987.

[BC82] Allan Borodin and Stephen Cook. A time-space tradeoff for sorting on a general sequential model of computation. *SIAM Journal on Computing*, 11(2):287–297, 1982.

[BCS16] Itai Benjamini, Gil Cohen, and Igor Shinkar. Bi-lipschitz bijection between the boolean cube and the hamming ball. *Israel Journal of Mathematics*, 212(2):677–703, 2016.

[Bea89] Paul Beame. A general sequential time-space tradeoff for finding unique elements. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 197–203, 1989.

[BFK+79] Allan Borodin, Michael J Fischer, David G Kirkpatrick, Nancy A Lynch, and Martin Tompa. A time-space tradeoff for sorting on non-oblivious machines. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 319–327. IEEE, 1979.

[BIL12] Chris Beck, Russell Impagliazzo, and Shachar Lovett. Large deviation bounds for decision trees and sampling lower bounds for ac0-circuits. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 101–110. IEEE, 2012.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CG20] Eshan Chattopadhyay and Jesse Goodman. Explicit designs and extractors. *arXiv preprint arXiv:2007.07772*, 2020.

[DW12] Anindya De and Thomas Watson. Extractors and lower bounds for locally samplable sources. *ACM Transactions on Computation Theory (TOCT)*, 4(1):1–21, 2012.

[FK18] Michael A Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 946–955. IEEE, 2018.

[GGN10] Oded Goldreich, Shafi Goldwasser, and Asaf Nussboim. On the implementation of huge random objects. *SIAM Journal on Computing*, 39(7):2761–2822, 2010.

[Gil52] Edgar N Gilbert. A comparison of signalling alphabets. *The Bell system technical journal*, 31(3):504–522, 1952.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on information theory*, 54(1):135–150, 2008.

[GV20] Rohit Gurjar and Ben Lee Volk. Pseudorandom bits for oblivious branching programs. *ACM Transactions on Computation Theory (TOCT)*, 12(2):1–12, 2020.

[GW20] Mika Göös and Thomas Watson. A lower bound for sampling disjoint sets. *ACM Transactions on Computation Theory (TOCT)*, 12(3):1–13, 2020.

[Hås87] Johan Håstad. *Computational limitations of small-depth circuits*. MIT press, 1987.

[IN96] Russell Impagliazzo and Moni Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of cryptology*, 9(4):199–216, 1996.

[JSWZ13] Rahul Jain, Yaoyun Shi, Zhaohui Wei, and Shengyu Zhang. Efficient protocols for generating bipartite classical distributions and quantum states. *IEEE Transactions on Information Theory*, 59(8):5171–5178, 2013.

[JVV86] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical computer science*, 43:169–188, 1986.

[Kil88] Joe Kilian. Founding crytpography on oblivious transfer. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 20–31, 1988.

[KM04] R Koenig and Ueli Maurer. Extracting randomness from generalized symbol-fixing and markov sources. In *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, page 232. IEEE, 2004.

[KM05] Robert Koenig and Ueli Maurer. Generalized strong extractors and deterministic privacy amplification. In *IMA International Conference on Cryptography and Coding*, pages 322–339. Springer, 2005.

[KRVZ11] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small-space sources. *Journal of Computer and System Sciences*, 77(1):191–220, 2011.

[LV12] Shachar Lovett and Emanuele Viola. Bounded-depth circuits cannot sample good codes. *Comput. Complex.*, 21(2):245–266, 2012.

[RY11] Ran Raz and Amir Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *Journal of Computer and System Sciences*, 77(1):167–190, 2011.

[Var57] Rom Rubenovich Varshamov. Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, SSSR*, 117:739–741, 1957.

[Vaz85] Umesh V Vazirani. Towards a strong communication complexity theory or generating quasi-random sequences from two communicating slightly-random sources. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 366–378, 1985.

[Vio12a] Emanuele Viola. The complexity of distributions. *SIAM Journal on Computing*, 41(1):191–218, 2012.

[Vio12b] Emanuele Viola. Extractors for turing-machine sources. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 663–671. Springer, 2012.

[Vio14] Emanuele Viola. Extractors for circuit sources. *SIAM Journal on Computing*, 43(2):655–672, 2014.

[Vio16] Emanuele Viola. Quadratic maps are hard to sample. *ACM Transactions on Computation Theory (TOCT)*, 8(4):1–4, 2016.

[Vio20] Emanuele Viola. Sampling lower bounds: boolean average-case and permutations. *SIAM Journal on Computing*, 49(1):119–137, 2020.

[Wat14] Thomas Watson. Time hierarchies for sampling distributions. *SIAM Journal on Computing*, 43(5):1709–1727, 2014.

[Wat16] Thomas Watson. Nonnegative rank vs. binary rank. *Chicago Journal of Theoretical Computer Science*, 2016(2), February 2016.

[Wat20] Thomas Watson. Communication complexity with small advantage. *computational complexity*, 29(1):1–37, 2020.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il