Electronic Colloquium on Computational Complexity, Revision 1 of 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

October 4, 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 obtain the first nontrivial sampling lower bounds against oblivious read-once branching programs (ROBPs).

In our first main result, 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. Previously, such a result was only known for sampling in AC^0 (Lovett and Viola, CCC'11; Beck, Impagliazzo and Lovett, FOCS'12). As an application of our result, a known connection implies new data structure lower bounds for storing codewords.

In our second main result, we prove a direct product theorem for sampling with ROBPs. Previously, no direct product theorems were known for the task of sampling, for any computational model. A key ingredient in our proof is a simple new lemma about amplifying statistical distance between sequences of somewhat-dependent random variables. Using this lemma, we also obtain a simple new proof of a known lower bound for sampling disjoint sets using two-party communication protocols (Göös and Watson, RANDOM'19).

[*] 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.

Following the earlier works of Ambainis, Schulman, Ta-Shma, Vazirani and Wigderson [AST$^+$03] and Goldreich, Goldwasser, and Nussboim [GGN10], Viola was the first to launch a systematic study on the *complexity of sampling distributions* [Vio12a]. Since then, this new area of complexity theory has seen an exciting wave of interest [Vio12b, LV12, BIL12, Vio14, DW12, JSWZ13, Aar14, Wat14, 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, little is known about the complexity of sampling with *limited memory*, while this remains a fundamental model in other areas of complexity.

In this work, we aim to fill this gap, and initiate a study of the *space* complexity of sampling. Our model will correspond to the streaming model of computation, an active area of research. Our work makes progress on the research program initiated by Viola, who has advocated for the pursuit of sampling lower bounds against every model for which we already have classical lower bounds (including branching programs, Turing machines, and polynomials) [Vio14].

1.1 Key questions

Before we formally introduce 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 Above, we saw that for the parity function $b : \{0,1\}^n \to \{0,1\}$, it holds that $(x, b(x))$ is hard to compute in AC^0, yet $(\mathbf{U}_n, b(\mathbf{U}_n))$ is easy to sample in AC^0. Given this observation, Viola raised the challenge [Vio12a] of finding a distribution of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$ that is *hard* to sample in AC^0. In a recent paper [Vio20], Viola provided a strong solution to this challenge, by 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.

Assuming Question 1 can be answered positively, it is natural to ask whether a similar result holds for sampling in the limited memory setting:

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 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). A complementary question, suggested by Viola [Vio12a], is to find other natural distributions $\mathbf{Q} \sim \{0,1\}^n$ 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 data structure lower bounds for storing codewords.

Given these results, we would like to know:

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

Direct product theorems Thus far, our questions have asked for small-space analogs of key results known for the complexity of sampling in AC^0. For our final question, we ask for a type of result that has yet to be studied in the complexity of sampling, but which has been well-explored within classical complexity. In particular, we ask for a *direct product theorem*.

In classical complexity, direct product theorems (e.g., Yao's XOR Lemma [Yao82]) are used for hardness amplification: such a result roughly says that if a function f is somewhat hard to compute for a given computational model, then t independent copies of f are *very* hard to compute for that same model. Direct product theorems offer a concrete way to (i) construct simple functions with strong (average-case) lower bounds, and thus (ii) establish strong (average-case) complexity separations between complexity classes.

It is natural to ask whether such direct product theorems can also be established for the task of sampling. In the context of sampling, a direct product theorem can be defined as a result which asserts the following: if a distribution $\mathbf{Q} \sim \{0,1\}^n$ has statistical distance δ from any distribution sampled by some computational model, then t independent copies of $\mathbf{Q}$ (concatenated together) has statistical distance $\gg \delta$ from any distribution sampled by that same model. Our final question is as follows.

Question 4. *Can a direct product theorem be established for distributions sampled in limited memory?*

In this paper, we make progress on these four questions. Before presenting our results, we must discuss our model for sampling distributions with limited memory.

1.2 Sampling in small space using oblivious ROBPs

To model sampling with limited memory, we will use the classic model of *oblivious read-once branching programs* (ROBPs). This 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.

1.3 Summary of our main results

With our questions in mind and our model formally defined, we are ready to state our results. Qualitatively, we provide positive answers to all four questions from Section 1.1.

Question 1 and Question 2 are straightforward to answer by applying known (or easy-to-prove) lower bounds from communication complexity. We provide formal statements and proofs for these results in Appendix A. On the other hand, Question 3 and Question 4 are more challenging to resolve, and our two main contributions are positive answers to these questions. We go into more detail below.

1.3.1 Sampling lower bounds against codes

In our first main theorem, we show that it is hard to sample good codes using ROBPs. More generally, we obtain the following sampling lower bounds against any (n, k, d) code, which are nearly tight.

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

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

Remark 1. *We show that Theorem 1 is nearly 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 $2^{\widetilde{O}(\frac{kd}{n})}$. For a more formal statement of this result, we refer the reader to Remark 2.*

As a corollary, we immediately get 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 1. *For any good code $\mathbf{Q} \sim \{0,1\}^n$, there is a constant $c > 0$ such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width at most 2^{cn},*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 12 \cdot 2^{-cn}.$$

Note that this is tight up to the constants c and 12, since statistical distance $\leq 1 - 2^{-n}$ is easily achieved by a width 1 ROBP that is constant over a single codeword. Finally, we note that we actually obtain a more general version of Theorem 1, which works for any list-decodable code: we refer the reader to Section 4 for more details.

We remark that these sampling lower bounds against codes for ROBPs are stronger than the best known sampling lower bounds against codes for AC^0. In particular, the best sampling lower bounds against good codes for AC^0 are of the form $1 - 2^{-n^{\Omega(1)}}$ [BIL12], and the authors leave as an open problem whether similar lower bounds for AC^0 can be obtained against (n, k, d) codes with $kd \geq n^{1+\Omega(1)}$. On the other hand, our sampling lower bounds against good codes are of the form $1 - 2^{-\Omega(n)}$ for ROBPs of width $2^{\Omega(n)}$ (Corollary 1), and we obtain lower bounds of the form $1 - 2^{-n^{\Omega(1)}}$ against (n, k, d) codes with $kd \geq n^{1+\Omega(1)}$, for ROBPs of width $2^{n^{\Omega(1)}}$ (Theorem 1).

Applications to data structure lower bounds 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 2. *For any good code $Q \subseteq \{0,1\}^n$ of dimension k, there is a constant $c > 0$ such that the following holds. If 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 at most 2^{cn}, then we must have redundancy $r \geq \lfloor cn \rfloor$.*

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.

We remark that these data structure lower bounds for storing codes and retrieving them using ROBPs is stronger than the best known data structure lower bounds for storing codes and retrieving them using AC^0 circuits. In particular, for ROBPs of exponential width $2^{\Omega(n)}$, we show that $r \geq \Omega(n)$ bits of redundancy are necessary, whereas the best known result for AC^0 requires $r \geq n^{\Omega(1)}$ bits of redundancy.

1.3.2 A direct product theorem

Our second main theorem is 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 2 (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 $1 - 2^{-\Omega(t)}$.

A simple new lemma on amplifying statistical distance A key ingredient in the proof of our direct product theorem is a simple new lemma on amplifying statistical distance between sequences of somewhat-dependent random variables. To the best of our knowledge, no such lemma was previously known, and we believe it may be of independent interest:

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

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

The proof of Lemma 1 is not difficult: we simply prove an analogous result over a more amenable notion of distance, known as the *Bhattacharyya coefficient*, and use estimates on statistical distance (in terms of the Bhattacharyya coefficient) to obtain the desired result. Despite its simple proof, we believe that it could be a useful tool for proving lower bounds. In particular, we discuss one such application, below.

Applications to sampling with two-party communication protocols As an application of the above lemma, we obtain a simple new proof of a known result on sampling lower bounds for two-party communication protocols [GW20]. In particular, in Section 6, we provide a short, self-contained proof that for any distribution $\mathbf{X} \sim \{0,1\}^n \times \{0,1\}^n$ sampled by two-party communication protocols with $\Omega(n)$ bits of communication, it holds that $\mathbf{X}$ has statistical distance $1 - 2^{-\Omega(n)}$ from the distribution $\mathbf{Q} \sim \{0,1\}^n \times \{0,1\}^n$ that is uniform over pairs of disjoint strings.

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 obtain our sampling lower bounds against codes, proving Theorem 1. Next, in Section 5, we prove our direct product theorem for sampling with ROBPs (Theorem 2). Finally, in Section 6, we show how a key ingredient of our direct product theorem can be used to obtain a simple new proof for a known result on sampling using communication protocols. We wrap up with some future directions in Section 7.

In our appendix, we include several useful results that are not too difficult to prove. In Appendix A, we show how known (or easy-to-prove) communication complexity lower bounds can be used to answer Question 1 and Question 2. Then, in Appendix B, we prove several equivalence theorems between various models for sampling and computing with limited memory. These theorems are first stated in the preliminaries (Section 3), and are used throughout the paper to streamline our proofs.

2 Overview of our techniques

In this section, we give a detailed overview of the techniques that go into proving our two main theorems: namely, our sampling lower bounds against codes (Theorem 1), and our direct product theorem (Theorem 2). Before we dive into these proofs, we briefly discuss an important tool that we use throughout the paper, which helps streamline our arguments about sampling with ROBPs.

An equivalence between two small-space samplers 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 with this model directly, we will actually prove many of our results using a *different model* for sampling with limited memory, known as a *small-space source*. This model is much easier to work with formally, as its definition is simpler. Surprisingly, it also turns out that sampling with this model is roughly equivalent to sampling with ROBPs. This means that we can largely focus on proving results about the simpler model. We go into more detail below.

Small-space sources were introduced by Kamp, Rao, Vadhan and Zuckerman [KRVZ11]. We henceforth refer to this model as the *KRVZ sampler*, and it is defined as a certain type of branching program that receives no input. More formally, a KRVZ 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 KRVZ 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.

Given this definition, we prove an equivalence theorem which says that a distribution $\mathbf{Q} \sim \{0,1\}^n$ is samplable by a KRVZ 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 4). The more challenging direction of this proof is that KRVZ 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 KRVZ 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 KRVZ samplers imply lower bounds against sampling with ROBPs; and (2) The existence of KRVZ samplers for a distribution implies that such a distribution can be sampled with ROBPs. As we will see now, both directions of this equivalence theorem will be useful in proving our main results.

2.1 Sampling lower bounds against codes

With our equivalence theorem in hand, we are ready to sketch the proof of our first main theorem (Theorem 1) and the proof of its tightness (Remark 1). Recall that Theorem 1 roughly says that for any distribution $\mathbf{Q} \sim \{0,1\}^n$ that is uniform over an (n, k, d) code with good parameters (i.e., k, d, large), it holds that any distribution sampled by an ROBP (whose width is not too large) will have statistical distance close to 1 from $\mathbf{Q}$. Using our equivalence theorem, it suffices to prove this theorem for KRVZ samplers, instead.

In order to prove this theorem, we actually prove a more general version that works for list-decodable codes. Recall that a (ρ, L) list-decodable code of dimension k is a subset $Q \subseteq \{0,1\}^n$ of size 2^k such that any Hamming ball of radius $\leq \rho n$ contains at most L points from Q. Note that any (n, k, d) code Q is a $(\rho, 1)$ list-decodable code of dimension k, for any $\rho < \frac{d}{2n}$. Thus it suffices to show that for any distribution $\mathbf{Q}$ uniform over a list-decodable code with good parameters (i.e., k, ρ large, L small), it holds that any distribution sampled by a KRVZ 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) KRVZ 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 $\mathbf{Y} = (\mathbf{Y}_1, \dots, \mathbf{Y}_r) \sim (\{0,1\}^\ell)^r$, where each $\mathbf{Y}_i$ is independent.

The second ingredient, which we will prove, is that product distributions, i.e., those of the form $\mathbf{Y} = (\mathbf{Y}_1, \dots, \mathbf{Y}_r) \sim (\{0,1\}^\ell)^r$ 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 KRVZ 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 $\mathbf{Y} = (\mathbf{Y}_1, \dots, \mathbf{Y}_r) \sim (\{0,1\}^\ell)^r$ 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 $x = (x_1, \dots, x_r) \in (\{0,1\}^\ell)^r$. 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 (i.e., p isn't too small), 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 $\Pr[\mathbf{Y} \in S] \leq p^{1/r} \cdot L$ 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 KRVZ 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 KRVZ samplers are far from list decodable codes? Well, let $\mathbf{X} \sim \{0,1\}^n$ be the KRVZ 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.

On the tightness of our result Above, we sketched the proof that any KRVZ sampler of not-too-large width generates a distribution that is very far from a good (n, k, d) code. More precisely, our result (Theorem 1) says that any KRVZ sampler of width $2^{\Omega(\frac{kd}{n})}$ has statistical distance $1 - 2^{-\Omega(\frac{kd}{n})}$ from any (n, k, d) code. In a complementary result, we show that this is nearly tight.

In more detail, we show that for almost all "valid" n, k, d, there is an (n, k, d) code that can be sampled by a KRVZ sampler (and thus an ROBP sampler) of width $2^{O(\frac{kd}{n} \cdot \log n)}$. More formally, we show this tightness result for all n, k, d for which there exists a linear (n, k, d) code.

Our proof of tightness is split into two cases. In the first case, we consider $k \leq 0.9n$. Here, the general idea is to define some n', k', d' such that there exists an (n', k', d') code Q', and then simply consider the repetition code $Q := Q' \times Q' \times \cdots \times Q'$, where n/n' copies of Q' participate in the Cartesian product. It is straightforward to verify that Q will be an $(n, k'n/n', d')$ code. Furthermore, it is not hard to see that a KRVZ sampler of width w can sample any distribution with support size w, and that the product distribution of two distributions, each samplable by a KRVZ sampler of width w, is also samplable by a KRVZ sampler of width w. Thus, Q will be samplable by a KRVZ sampler of width $2^{k'}$. Thus, if we can find a constant C and an (n', k', d') code with $n' = Cd, k' = Ckd/n, d' = d$, we are done with this case. Since $k \leq 0.9n$, the Gilbert-Varshamov bound guarantees this is always possible.

In the second case, we consider $k > 0.9n$. Here, the general idea is that $n - k$ will now be small. We consider two subcases: $k \geq n - 4d \log n$ and $k < n - 4d \log n$. We focus on the first subcase in this overview, as it is not too hard to extend the argument to work for the second subcase. In the first subcase, note that $n - k \leq 4d \log n$. Now, the main idea is that ROBPs can check membership of a k dimensional subspace $Q \subseteq \mathbb{F}_2^n$ using width 2^{n-k}, simply by keeping track of the $n - k$ parity checks that define Q. Furthermore, it is not too hard to show that for any ROBP of width w, the uniform distribution over its accepting strings can be generated by a KRVZ sampler of width w. Thus in this case, we can simply take any linear (n, k, d) code and uniformly sample from it using a KRVZ sampler of width $2^{n-k} \leq 2^{4d \log n} \leq 2^{\frac{5kd}{n} \cdot \log n}$, where the last inequality follows from the current case $k > 0.9n$.

2.2 A direct product theorem

We now begin our sketch of our second main theorem, Theorem 2. 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 w. Here, recall that $\mathbf{Q}^{\otimes t}$ refers to a sequence of t independent copies of $\mathbf{Q}$.

In order to prove the above direct product theorem, we start by proving an analogous result for KRVZ samplers. Then, we use our equivalence theorem to obtain a direct product theorem for sampling with ROBPs. However, since our equivalence theorem (Theorem 4) has some loss in parameters (width), this will only yield a *weak* direct product theorem: in such a result, the statistical distance still blows up from δ to $1 - 2^{-\Omega(t\delta^2)}$, *but only if* we also require the width to *decrease* from w to $w/14$.

We would really like a *strong* direct product theorem, in the sense that the statistical distance blows up even if the ROBP is allowed to keep all of its width w. At the end of this subsection, we show how to build some extra machinery to make this happen.

A direct product theorem for KRVZ samplers We now proceed to sketch the proof of our direct product theorem for KRVZ samplers.

Let $\mathbf{X} \sim \{0,1\}^{nt}$ be a distribution generated by a KRVZ 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 for any KRVZ 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 KRVZ sampler of width w. Indeed, even though it represents a sequence of bits generated in the middle of the KRVZ sampler $\mathbf{X}$, it is easy to create a new KRVZ sampler $\mathcal{B}$ of the same width that only generates $\mathbf{X}_i$, simply by: (1) copying the KRVZ 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.

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 (that every KRVZ sampler of width w is far from $\mathbf{Q}$). 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 KRVZ 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 $(\mathbf{Q}^{\otimes}_{-i})$ are identical.[1] 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, 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 $i \in [t]$ 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 (2) even true? If we can answer both questions in the affirmative, then we are done with our direct product theorem for KRVZ samplers.

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 KRVZ 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 our direct product theorem for KRVZ samplers is complete.

[1] 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.

A direct product theorem for sampling with ROBPs It is now easy to obtain a direct product theorem for ROBP samplers, in a black-box manner, by combining the above direct product theorem with our equivalence theorem between KRVZ samplers and ROBP samplers (Theorem 4). However, as discussed at the beginning of this section, this will only yield a weak direct product theorem, since our equivalence theorem suffers a slight loss in parameters (i.e., width). If we want to obtain a *strong* direct product theorem for ROBP samplers, we must dig into the black box.

Looking back at the previous discussion, it is not too difficult to see that if one wants a strong direct product theorem for ROBP samplers (that suffers no loss in width), then it suffices to show the following: for any distribution $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_t) \sim (\{0,1\}^n)^t$ sampled by an ROBP of width w, and any $i \in [t], x \in (\{0,1\}^n)^{i-1}$, the distribution $(\mathbf{X}_i \mid \mathbf{X}_1, \dots, \mathbf{X}_{i-1} = x)$ can be sampled by an ROBP of width w.

In order to show the above, our key ingredient is the following: for any w and probability distribution $p : [w] \to \mathbb{R}_{\geq 0}$, we construct an ROBP of width w such that a random walk over it hits the i^{th} vertex in the last layer with probability $p(i) + \gamma$, where $\gamma > 0$ can be arbitrarily small. A first attempt at constructing such an ROBP might use the "multi-thresholding" discussed at the beginning of Section 2, which was used in our equivalence theorem. However, our construction of such a function required width $2w$ (instead of the desired w), and we show that this is tight up to additive constants.

To get the width down to w, we start by showing that for any biased coin $\mathbf{A} \sim \{0,1\}$, there is an ROBP of width 2 that samples a distribution arbitrarily close to it. At a high level, this argument works as follows: for any binary string $b \in \{0,1\}^\ell$, we show how to use its bits as "instructions" to construct a certain ROBP of length ℓ and width 2 in a layer-by-layer fashion. The constructed ROBP then guarantees that it accepts a random string with probability b, where b is interpreted as the binary representation of a number in $[0,1]$. Finally, it is not too hard to bootstrap such an object to create our desired ROBP of width w that hits the vertices in its last layer with probabilities close to $\{p(i)\}_{i \in [w]}$. As a result, we get our strong direct product theorem for sampling with ROBPs.

A key ingredient of our direct product theorem, and a simple new proof of [GW20] A key ingredient of the above proofs is a simple new lemma on amplifying statistical distance between sequences of somewhat-dependent random variables. In particular, we show that for any random variables $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)$ and $\mathbf{Q} = (\mathbf{Q}_1, \dots, \mathbf{Q}_n)$ over the same domain, if it holds that $|(\mathbf{X}_i \mid \mathbf{X}_{<i} = x) - (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = x)| \geq \delta$ for each i, x, then $|\mathbf{X} - \mathbf{Q}| \geq 1 - 2^{-\Omega(n\delta^2)}$. The proof is not difficult, and we believe it could be a useful new tool for proving lower bounds.

As an application, we give a simple new proof of a result by Göös and Watson [GW20] that any distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^n$ sampled by two-party communication protocols (with communication $b = \Omega(n)$) is far from the distribution $(\mathbf{A}, \mathbf{B}) \sim \{0,1\}^n \times \{0,1\}^n$ that is uniform over pairs of disjoint strings. We present it below:

1. First, use the standard observation (made in, e.g., [AST+03]) that $(\mathbf{X}, \mathbf{Y})$ is a convex combination of 2^b product distributions $(\mathbf{X}', \mathbf{Y}')$.

2. Use a standard data processing inequality to observe that
$$|(\mathbf{X}', \mathbf{Y}') - (\mathbf{A}, \mathbf{B})| \geq |(\mathbf{X}'_1, \mathbf{Y}'_1, \dots, \mathbf{X}'_n, \mathbf{Y}'_n) - (\mathbf{A}_1, \mathbf{B}_1, \dots, \mathbf{A}_n, \mathbf{B}_n)|.$$

3. Using a straightforward calculation, observe that $|(\mathbf{X}'_i, \mathbf{Y}'_i) - (\mathbf{A}_i, \mathbf{B}_i)| \geq \Omega(1)$ for each i, since $\mathbf{X}'_i, \mathbf{Y}'_i$ are independent and $(\mathbf{A}_i, \mathbf{B}_i)$ is uniform over $\{(0,0), (0,1), (1,0)\}$. Observe that this still holds even if you condition on any fixing of the random variables earlier on in the sequence, since this doesn't break the independence of $\mathbf{X}'_i, \mathbf{Y}'_i$, nor does it change the distribution of $(\mathbf{A}_i, \mathbf{B}_i)$.

4. Use our new lemma on amplifying statistical distance to conclude $|(\mathbf{X}', \mathbf{Y}') - (\mathbf{A}, \mathbf{B})| \geq 1 - 2^{-\Omega(n)}$.

5. Using the fact that the convex combination of a few far distributions is still far (Lemma 2), conclude that $|(\mathbf{X}, \mathbf{Y}) - (\mathbf{A}, \mathbf{B})| \geq 1 - 2^{b-\Omega(n)}$, which is $1 - 2^{-\Omega(n)}$ for some $b = \Omega(n)$, as desired.

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 $\{1, 2, \dots, n\}$. 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 $x_{i \to j} := (x_i, \dots, x_j) \in \{0,1\}^{j-i+1}$, 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 $x^\pi := (x_{\pi(1)}, \dots, x_{\pi(n)})$.

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 $\mathbf{X}^\pi := (\mathbf{X}_{\pi(1)}, \dots, \mathbf{X}_{\pi(n)})$. 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 $\mathbf{Y}_1, \dots, \mathbf{Y}_k$ 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, it will be very useful to have the following lemma, which says that for any distribution $\mathbf{Q}$, a convex combination of not-too-many distributions, each far from $\mathbf{Q}$, is itself far from $\mathbf{Q}$. This slightly generalizes a result from [Vio20], where it was shown to hold for convex combinations of the form $\sum_i p_i \mathbf{Y}_i$, where each p_i is the same.

Lemma 2. *Let* $\mathbf{X}, \mathbf{Q}$ *be any two random variables over the same discrete space* V*. Suppose that* $\mathbf{X}$ *is a convex combination of* t *distributions* $\mathbf{X} = \sum_{i \in [t]} p_i \mathbf{Y}_i$*, where for each* $\mathbf{Y}_i$ *we have* $|\mathbf{Y}_i - \mathbf{Q}| \geq 1 - \delta$*. Then*

$$|\mathbf{X} - \mathbf{Q}| \geq 1 - t\delta.$$

Proof. Let Q denote the support of $\mathbf{Q}$. By definition of statistical distance, we know that for each $i \in [t]$ there is a set S_i such that $|\mathbf{Y}_i - \mathbf{Q}| = \Pr[\mathbf{Q} \in Q - S_i] - \Pr[\mathbf{Y}_i \in Q - S_i] = 1 - (\Pr[\mathbf{Q} \in S_i] + \Pr[\mathbf{Y}_i \in Q - S_i])$. And by the lemma hypothesis, we know that $\Pr[\mathbf{Q} \in S_i] + \Pr[\mathbf{Y}_i \in Q - S_i] \leq \delta$.

Now, define $S = \bigcup_i S_i$. By definition of statistical distance, $|\mathbf{X} - \mathbf{Q}| \geq 1 - (\Pr[\mathbf{Q} \in S] + \Pr[\mathbf{X} \in Q - S])$. Thus it suffices to show $\Pr[\mathbf{Q} \in S] + \Pr[\mathbf{X} \in Q - S] \leq t\delta$ to complete the proof. Towards this end, observe that we can write 1 as $\sum_j p_j$ to obtain

$$\Pr[\mathbf{Q} \in S] \leq \sum_{i \in [t]} \Pr[\mathbf{Q} \in S_i] = \sum_{i,j \in [t]} p_j \Pr[\mathbf{Q} \in S_i].$$

Next, observe that

$$\Pr[\mathbf{X} \in Q - S] = \sum_{j \in [t]} p_j \Pr[\mathbf{Y}_j \in Q - S] \leq \sum_{i,j \in [t]} p_j \Pr[\mathbf{Y}_j \in Q - S_i].$$

Thus we have

$$\Pr[\mathbf{Q} \in S] + \Pr[\mathbf{X} \in Q - S] \leq \sum_{i,j \in [t]} p_j \cdot (\Pr[\mathbf{Q} \in S_i] + \Pr[\mathbf{Y}_j \in Q - S_i]) \leq \sum_{i,j \in [t]} p_j \cdot \delta \leq t\delta,$$

as desired. □

Bias, correlation, and covariance We will prove several results that allow us to convert sampling lower bounds into worst-case and average-case lower bounds for computation. For this, we 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 3 ([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 $\mathsf{0}$, and the other labeled $\mathsf{1}$. 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 $x_1, x_2, \dots, x_n$. 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 $\mathsf{0}$, and the other labeled with the input bit $\mathsf{1}$. 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 $x_1, x_2, \dots, x_n$. 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]. We call it a KRVZ sampler, and will show that it is equivalent (up to a small loss in parameters) to the ROBP sampler.

Definition 10 (KRVZ sampler)**.** *A* KRVZ 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 $\mathsf{0}$, and the rest labeled $\mathsf{1}$. 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.*

Throughout the paper, we also prove sampling *upper bounds*, to show the tightness of our results. To obtain our upper bounds, we will construct KRVZ samplers that come close to sampling certain distributions. As it turns out, our upper bounds will actually be even stronger than this, and we will show it often suffices to use a weaker model of sampling, which we call a *simple sampler*. It is identical to the KRVZ 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 $\mathsf{0}$, and the other labeled $\mathsf{1}$. 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.*

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 KRVZ samplers will be important when we convert simple and KRVZ samplers into ROBP samplers.

Definition 12. *A KRVZ 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 KRVZ 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 by Fact 4, but we give a different proof which allows for more flexibility when designing *granular* KRVZ samplers.

Fact 5. *Any distribution* $\mathbf{X} \sim \{0,1\}^n$ *can be sampled by a KRVZ 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 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 Equivalence theorems

Throughout the paper, we make use of several *equivalence theorems*, which establish relationships between various models for sampling and computing in limited memory. These equivalence theorems will help us focus on proving lower bounds for the model with the simplest definition, which can then be lifted to lower bounds against the model of interest. We prove these equivalence theorems in Appendix B, and record the most important ones, below.

First, we show that any distribution that can be sampled by a low-width KRVZ sampler can also be sampled by a low-width ROBP sampler (up to a small loss in parameters).

Theorem 4. *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 KRVZ sampler of width* $2w$ *that samples* $\mathbf{X}$.
- *If there exists a KRVZ 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* $\mathbf{X}$.

If the KRVZ sampler in the second bullet of the above theorem is *granular* (Definition 12), then we can design an ROBP sampler that exactly samples the same distribution:

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

As its name suggests, this lemma (combined with a result of Kamp, Rao, Vadhan and Zuckerman [KRVZ11]) is actually used to prove the second (and more challenging) bullet in Theorem 4. We also prove a different version of this lemma, which focuses on minimizing the randomness used by the ROBP sampler, at the expense of introducing slightly more width.

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

We will use both bullets of Theorem 4 in the paper, although we will often opt to use Lemma 3 or Lemma 4 instead of the second bullet. Next, we will prove a perhaps surprising equivalence between *sampling* with simple samplers and *computing* with ROBPs.

Theorem 5. *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.*

One direction of this theorem offers a way to translate sampling lower bounds against simple (and thus KRVZ and ROBP) samplers into hard functions for ROBPs. In fact, we actually prove the following stronger result, which offers a way to translate such lower bounds into correlation bounds against ROBPs.

Lemma 5. *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 $|\operatorname{corr}(f, b)| < \varepsilon$.*

Finally, we prove a second equivalence between sampling with simple samplers and computing with ROBPs.

Theorem 6. *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 $\mathbf{Q}$.*

The second bullet of the above theorem is quite useful: for example, it gives an easy way to sample from affine spaces (by computing parity checks in low-width), and furthermore it offers a way to translate lower bounds against simple samplers (and thus KRVZ and ROBP) samplers into hard functions for ROBPs. In fact, we actually prove the following stronger result, which offers a way to translate such lower bounds into covariance bounds against ROBPs.

Lemma 6. *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 $|\operatorname{cov}(f, 1_S)| < \varepsilon$.*

We are now ready to proceed to the technical portion of the paper.

4 Sampling lower bounds against codes

In this section, we will prove our first main theorem (Theorem 1) on sampling lower bounds against codes. We start by proving the most general version of this result in Section 4.1: namely, we prove sampling lower bounds against list-decodable codes for KRVZ samplers. Then, in Section 4.2, we show how this result specializes to sampling lower bounds against (n, k, d) codes, which are shown to be nearly tight. In Section 4.3, by combining the above results with our equivalence theorem (Theorem 4), we obtain the corresponding results for sampling with ROBPs. Finally, in Section 4.4, we show how these sampling lower bounds imply new data structure lower bounds, using a known connection.

4.1 Sampling lower bounds against list-decodable codes

We start by proving the following sampling lower bounds against list-decodable codes for KRVZ samplers, which is our main result of the section.

Theorem 7. *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over a (ρ, L) list decodable code of dimension k. Then for any KRVZ 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. We start by writing our KRVZ sampler as a convex combination of random variables with nice structure. Let $\mathbf{W} = (\mathbf{W}_1, \dots, \mathbf{W}_n)$ 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 $\mathbf{W}^* = (\mathbf{W}_\ell, \mathbf{W}_{2\ell}, \dots, \mathbf{W}_{r\ell})$, 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 $\mathbf{X}^{(W)} := (\mathbf{X}_1^{(W)}, \mathbf{X}_2^{(W)}, \dots, \mathbf{X}_r^{(W)})$, where each $\mathbf{X}_i^{(W)} \sim \{0,1\}^\ell$ is independent. Thus the KRVZ 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 $\mathbf{Y}=(\mathbf{Y}_1,\dots,\mathbf{Y}_r)$ 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 $(q_1,q_2,\dots,q_r)\in(\{0,1\}^\ell)^r$, 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 $x=(x_1,x_2,\dots,x_r)\in(\{0,1\}^\ell)^r$, we let $x_{-i}:=(x_1,\dots,x_{i-1},x_{i+1},\dots,x_r)$ 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[2] $t = \frac{r}{r+1} \cdot (k - r\log w + \log L)$ 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

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1}. \tag{3}$$

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 (3), 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 (3) 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. □

[2] 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.

4.2 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 7. 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 8. *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over an (n, k, d) code. Then for any KRVZ 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 KRVZ 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 8 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 KRVZ sampler of width $w = 2^{\widetilde{O}(\frac{kd}{n})}$. More formally, we show the following.

Theorem 9. *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 KRVZ 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 3), 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 6. 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 $v^{(1)}, \dots, v^{(n-k)}$ 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 $v^{(1)}, \dots, v^{(n-k)}$ 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 $(\langle x, v^{(1)} \rangle, \dots, \langle x, y^{(n-k)} \rangle)$. 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 6 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 3), 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 $v^{(1)}, \dots, v^{(n-k')}$ 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 $v^{(1)}, \dots, v^{(n-k')}$. Without loss of generality, assume they are $\hat{e}^{(1)}, \dots, \hat{e}^{(k'-1)}$. Notice now that $t \leq k' - 1$ (since we may assume $k \geq 1$), and consider the subspace $\widetilde{Q}$ spanned by basis vectors $v^{(1)}, \dots, v^{(n-k')}, \hat{e}^{(1)}, \dots, \hat{e}^{(t)}$. 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 $v \in \{v^{(1)}, \dots, v^{(n-k')}, \hat{e}^{(1)}, \dots, \hat{e}^{(t)}\}$. 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 6.

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 $v^{(1)}, \dots, v^{(n-k')}$, while the remaining two parts will keep track of the parity checks $\hat{e}^{(1)}, \dots, \hat{e}^{(t)}$ 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 $v^{(1)}, \dots, v^{(n-k')}$. 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 $\hat{e}^{(1)}, \dots, \hat{e}^{(t)}$ 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 $v^{(1)}, \dots, v^{(n-k')}, \hat{e}^{(1)}, \dots, \hat{e}^{(t)}$. 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 6, 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. □

4.3 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 10. *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 7 with the first bullet of Theorem 4. □

Next, by Fact 2, we obtain the following specialization of Theorem 10:

Corollary 3 (Theorem 1, restated)**.** *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 KRVZ samplers against (n,k,d) codes are almost tight (Theorem 9). In fact, the tightness is actually witnessed by a simple sampler that is 2^{-n}-granular. Thus, combining Theorem 9 with Lemma 4, we obtain the following.

Remark 2 (Remark 1, formal version)**.** *Corollary 3 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. First, by applying Theorem 6, we immediately get that ROBPs of exponential width cannot test membership of a good code. In fact, using Lemma 6, we immediately get the following (stronger) 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, we show how our sampling lower bounds against codes yield data structure lower bounds against storing codewords succinctly and retrieving them using ROBPs.

4.4 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 5 (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 5. 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 10, 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 6. *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 $2^{\Omega(n)}$.

5 A direct product theorem

In this section, we present our direct product theorems. Our main result will be the following strong direct product theorem for sampling with ROBPs.

Theorem 11 (Theorem 2, 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 $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq \delta$. 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*

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

We start by proving a simple new lemma on amplifying statistical distance, in Section 5.1. Then, in Section 5.2, we show how this lemma can be combined with a basic fact about KRVZ samplers in order to obtain a strong direct product theorem for KRVZ samplers. Finally, in Section 5.3, we start by showing how this result can be combined with our KRVZ-ROBP equivalence theorem (Theorem 4) in order to obtain a weak direct product theorem for sampling with ROBPs. Then, we show how to turn this into a strong direct product theorem by employing some more involved tools, ultimately proving Theorem 11.

5.1 A simple lemma on amplifying statistical distance

We start with a simple lemma on amplifying statistical distance, which will be a key ingredient in the proof of our direct product theorems.

Lemma 7. *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*. Given two random variables $\mathbf{X}, \mathbf{Y}$ over some discrete space V, the squared Hellinger distance between $\mathbf{X}, \mathbf{Y}$ is denoted $H^2(\mathbf{X}, \mathbf{Y})$ and defined as follows:

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

We would now like to express Lemma 7 using this more well-behaved notion of distance. For this, we can use 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 7. *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 7 into a statement about squared Hellinger distance. However, it turns out that there is another related measure of distance/similarity that will make Lemma 7 even easier to prove. It is known as the *Bhattacharyya coefficient*. Given two random variables $\mathbf{X}, \mathbf{Y}$ over some discrete space V, the Bhattacharyya coefficient between $\mathbf{X}$ and $\mathbf{Y}$ is denoted $\mathsf{BC}(\mathbf{X}, \mathbf{Y})$ and defined as follows:

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

It is easy to verify, via the definitions of squared Hellinger distance and Bhattacharyya coefficient, that $1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y}) = H^2(\mathbf{X}, \mathbf{Y})$. We can combine this observation with Fact 7 to obtain the following estimates on statistical distance in terms of the Bhattacharyya coefficient.

Fact 8. *For any discrete random variables* $\mathbf{X}, \mathbf{Y} \sim V$*,*

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

The goal now is to prove a version of Lemma 7 that uses the Bhattacharyya coefficient. We can then combine this result with Fact 8 to prove Lemma 7.

Lemma 8. *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* $v \in V^{i-1}$*,*

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

Then

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

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

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

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

We now combine the above lemma with our estimates from Fact 8 to prove Lemma 7.

Proof of Lemma 7. By combining the lemma hypothesis with Fact 8, we know that for any i, v,

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

Thus by Lemma 8 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 8 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. □

5.2 A direct product theorem for KRVZ samplers

We are now ready to prove a strong direct product theorem for KRVZ samplers.

Theorem 12. *Let $\mathbf{Q} \sim \{0,1\}^n$ be a distribution such that for any KRVZ sampler $\mathbf{X} \sim \{0,1\}^n$ of width w, it holds that $|\mathbf{X} - \mathbf{Q}| \geq \delta$. Then for any KRVZ sampler $\mathbf{X}^* \sim \{0,1\}^{nt}$ of width w,*

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

To prove this lemma, we will simply combine our lemma on amplifying statistical distance (Lemma 7) with the following fact, which is straightforward to verify using the definition of KRVZ sampler.

Fact 9. *Let $\mathbf{X} \sim \{0,1\}^n$ be a KRVZ 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 KRVZ sampler of width w.

We can now formally prove Theorem 12.

Proof of Theorem 12. Parse $\mathbf{X}^*$ as $(\mathbf{X}_1, \ldots, \mathbf{X}_t) \sim (\{0,1\}^n)^t$, and parse $\mathbf{Q}^{\otimes t}$ as $(\mathbf{Q}_1, \ldots, \mathbf{Q}_t) \sim (\{0,1\}^n)^t$. By definition, each $\mathbf{Q}_i$ is an independent copy of $\mathbf{Q}$. And by Fact 9, we know that for any fixing of $\mathbf{X}_{<i}$, the random variable $\mathbf{X}_i$ is a KRVZ sampler of width w. Thus for any $i \in [t]$ and $v \in (\{0,1\}^n)^{i-1}$, we have

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

by the hypothesis $|\mathbf{X} - \mathbf{Q}| \geq \delta$. Applying Lemma 7 completes the proof. □

5.3 A direct product theorem for sampling with ROBPs

Finally, we are ready to prove our main direct product theorem, which we restate below for convenience.

Theorem 13 (Theorem 11, 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 $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq \delta$. 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*

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

In order to prove this result, we would like to apply our KRVZ-ROBP equivalence theorem (Theorem 4) with our direct product theorem from KRVZ samplers (Theorem 12). Such a proof would look roughly as follows: first, if every ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w has $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq \delta$, then by the second bullet of Theorem 4, every KRVZ sampler $\mathbf{X} \sim \{0,1\}^n$ of width $w/7$ has roughly $|\mathbf{X} - \mathbf{Q}| \geq \delta$. Then, by Theorem 12, we know that every KRVZ sampler $\mathbf{X}^* \sim \{0,1\}^{nt}$ of width $w/7$ has roughly $|\mathbf{X}^* - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}$. Finally, by the first bullet of Theorem 4, every ROBP $F : \{0,1\}^{\ell^*} \to \{0,1\}^{nt}$ of width $w/14$ must have $|F^*(\mathbf{U}_{\ell^*}) - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}$.

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, the goal will be to prove a version of Fact 9 for sampling using ROBPs. Just as in the proof of our direct product theorem for KRVZ samplers, we will then be able to combine this result with our lemma on amplifying statistical distance from Section 5.1 in order to prove Theorem 13. Thus, we proceed towards proving such a result, which will be significantly more challenging to prove than Fact 9 (which was easy to verify by the definition of KRVZ sampler).

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 $1 \leq i \leq j \leq n$ 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*

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

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 $\mathbf{X}_{<i} = x$ by prepending another ROBP. The exact type of ROBP we would like to prepend should compute a function $g : \{0,1\}^{\ell'} \to [w]$ such that $g(\mathbf{U}_{\ell'}) = \alpha$ with roughly the same probability that F^* hits vertex α in layer i, conditioned on $\mathbf{X}_{<i} = x$.

To make things more formal, we introduce a slightly different type of ROBP called a Σ-ROBP, which can be viewed as an intermediate model between single-output ROBPs (Definition 7) and multi-output ROBPs (Definition 8). A Σ-ROBP 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 Σ-ROBP will then compute a function $f_{\mathcal{B}} : \{0,1\}^n \to \Sigma$ in the natural way: on input $x \in \{0,1\}^n$, it traces the path with edge labels corresponding to x, and outputs the label (in Σ) of the final vertex on this path.

Given this definition, we would like to construct a Σ-ROBP of width w that computes the g described above. Towards this end, one idea is to define g to compute a "multi-tresholding function", which roughly splits the hypercube $\{0,1\}^{\ell'}$ into w consecutive buckets of an appropriate size, and on input x outputs the label of the bucket it falls into. Such a construction can indeed be used to compute the appropriate probabilities, and we use this in Appendix B.1 to prove the equivalence between KRVZ and ROBP samplers. However, our Σ-ROBP from that section requires width $2w$, and we show that this is tight. To compute g using width w, we need a new idea. Before presenting this new idea and proving Claim 1, let us show how it can be combined with Lemma 7 to prove Theorem 13.

Proof of Theorem 13. Let $\mathbf{X}^* = F^*(\mathbf{U}_{\ell^*})$. Parse $\mathbf{X}^*$ as $(\mathbf{X}_1, \ldots, \mathbf{X}_t) \sim (\{0,1\}^n)^t$, and parse $\mathbf{Q}^{\otimes t}$ as $(\mathbf{Q}_1, \ldots, \mathbf{Q}_t) \sim (\{0,1\}^n)^t$. 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

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

Furthermore, by the theorem hypothesis, we know that $|F(\mathbf{U}_\ell) - \mathbf{Q}_i| \geq \delta$. Thus by the triangle inequality we have $|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - \mathbf{Q}_i| \geq \delta/\sqrt{2}$. 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

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

By applying Lemma 7, we immediately get that $|\mathbf{X}^* - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}$, 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 Appendix B.1, but using less width. In particular, we prove the following.

Lemma 9 (Key ingredient for Theorem 13)**.** *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*

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

Before we prove Lemma 9, 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 $1 \leq i \leq j \leq n$ 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 $\alpha \leq \beta \in [\ell^*]$ 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 $\sum_{h\in[\alpha]} \gamma_h \geq i$ and β is the smallest integer such that $\sum_{h\in[\beta]} \gamma_h \geq j$.

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 $\sum_{h\in[\alpha-1]} \gamma_h \leq i - 1 < \sum_{h\in[\alpha]} \gamma_h$. Let $r := (i-1) - \sum_{h\in[\alpha-1]} \gamma_h \geq 0$ 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 9, 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

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

as desired. □

At last, all that remains is to prove the key ingredient Lemma 9. We do so, below.

Proof of Lemma 9. 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 $|f(\mathbf{U}_\ell) - \mathbf{X}| \le \varepsilon$. 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

$$\Pr[\mathbf{X} = i] = \begin{cases} \Pr[\mathbf{A}_i = 0] \cdot \prod_{j<i} \Pr[\mathbf{A}_j = 1] & \text{if } i < w, \\ \Pr[\mathbf{A}_{i-1} = 1] \cdot \prod_{j<i-1} \Pr[\mathbf{A}_j = 1] & \text{if } i = w. \end{cases} \tag{4}$$

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

$$V = (V^1_0 \cup \cdots \cup V^1_{\ell'}) \cup (V^2_0 \cup \cdots \cup V^2_{\ell'}) \cup \cdots \cup (V^{w-1}_0 \cup \cdots \cup V^{w-1}_{\ell'}). \tag{5}$$

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 (5), 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

$$\Pr[\mathbf{X}' = i] = \begin{cases} \Pr[\mathbf{B}_i = 0] \cdot \prod_{j<i} \Pr[\mathbf{B}_j = 1] & \text{if } i < w, \\ \Pr[\mathbf{B}_{i-1} = 1] \cdot \prod_{j<i-1} \Pr[\mathbf{B}_j = 1] & \text{if } i = w. \end{cases} \tag{6}$$

Our goal now is to upper bound $|\mathbf{X}' - \mathbf{X}|$ using Equations (4) and (6). Towards this end, suppose we have a sequence of probabilities $p_1, \ldots, p_i$ and another sequence of probabilities $q_1, \ldots, q_i$ 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 (4) and (6) 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.[3]

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.

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

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

[3] 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.

But observe that this quantity is simply a decimal written in binary using $\{b_1, b_2, \dots, b_{\ell'}\}$. 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,

$$|f'(\mathbf{U}_{\ell'}) - \mathbf{Y}| < 2^{-\ell'}. \tag{7}$$

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 (7) by ε' and complete the proof. □

6 A simple new proof of sampling lower bounds against disjoint sets

In the previous section, we proved our direct product theorems. A key ingredient we used was a simple new lemma about amplifying statistical distance between sequences of somewhat-dependent random variables (Lemma 7). In this section, we will show how this lemma can be used to yield a simple new proof of a known result on the complexity of sampling disjoint sets using two-party communication protocols.

Sampling using two-party communication protocols Thus far, we have discussed sampling distributions using ROBPs and AC^0 circuits. In the works [AST+03, GW20], the authors consider sampling distributions using two-party communication protocols. Here, Alice and Bob receive private randomness $\mathbf{A}$ and $\mathbf{B}$, and no other input. Then, Alice and Bob take turns communicating bits to one another, until the protocol ends. We let $\Pi(\mathbf{A}, \mathbf{B})$ denote the sequence of bits communicated (i.e., the transcript). At the end, Alice uses $\mathbf{A}$ and $\Pi(\mathbf{A}, \mathbf{B})$ to output some string $\mathbf{X} \sim \{0,1\}^n$, while Bob uses $\mathbf{B}$ and $\Pi(\mathbf{A}, \mathbf{B})$ to output some string $\mathbf{Y} \sim \{0,1\}^n$. The distribution generated by the protocol is $(\mathbf{X}, \mathbf{Y})$.

In the work [GW20], Göös and Watson study the complexity of generating disjoint subsets using two-party communication protocols. More formally, they ask for the minimum number of bits communicated by any two-party protocol that generates the distribution $(\mathbf{S}, \mathbf{T}) \sim \{0,1\}^n \times \{0,1\}^n$ that is uniform over all strings $(s, t) \in \{0,1\}^n \times \{0,1\}^n$ with $s \wedge t = 0^n$. Their main result is the following.

Theorem 14 ([GW20])**.** *For any distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^n$ sampled by a two-party protocol using $\Omega(n)$ bits of communication,*

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{S}, \mathbf{T})| \geq 1 - 2^{-\Omega(n)}.$$

Below, we provide a simple new proof of this result, by applying our new lemma on amplifying statistical distance (Lemma 7). We reminder the reader that this lemma has a simple, self-contained proof in Section 5.1.

New proof of Theorem 14. Suppose Alice and Bob have private randomness $\mathbf{A}$ and $\mathbf{B}$, respectively, and that they generate the distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^n$ using a protocol that exchanges b bits of communication. A simple observation (made in [AST+03]) is that if we condition on any fixing of the communication transcript $\Pi(\mathbf{A}, \mathbf{B})$, then $\mathbf{X}, \mathbf{Y}$ become independent. In other words, $(\mathbf{X}, \mathbf{Y})$ is a convex combination of $\leq 2^b$ distributions of the form $(\mathbf{X}', \mathbf{Y}') \sim \{0,1\}^n \times \{0,1\}^n$, where $\mathbf{X}', \mathbf{Y}'$ are independent.

We now seek to lower bound each $|(\mathbf{X}', \mathbf{Y}') - (\mathbf{S}, \mathbf{T})|$. Towards this end, define random variables $\mathbf{W} = (\mathbf{W}_1, \dots, \mathbf{W}_n)$ and $\mathbf{R} = (\mathbf{R}_1, \dots, \mathbf{R}_n)$, where each $\mathbf{W}_i := (\mathbf{X}'_i, \mathbf{Y}'_i)$ and each $\mathbf{R}_i := (\mathbf{S}_i, \mathbf{T}_i)$. Since $\mathbf{W}$ simply permutes the bits of $(\mathbf{X}', \mathbf{Y}')$, and $\mathbf{R}$ permutes the bits of $(\mathbf{S}, \mathbf{T})$ in the same way, we have via the data processing inequality (Fact 1) that

$$|(\mathbf{X}', \mathbf{Y}') - (\mathbf{S}, \mathbf{T})| \geq |\mathbf{W} - \mathbf{R}| = |(\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_n) - (\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_n)|.$$

Now, consider any $i \in [n]$, let v be in $(\{0,1\}^2)^{i-1}$, and consider the random variables $(\mathbf{W}_i \mid \mathbf{W}_{<i} = v)$ and $(\mathbf{R}_i \mid \mathbf{R}_{<i} = v)$. Notice that $(\mathbf{W}_i \mid \mathbf{W}_{<i} = v)$ is of the form $((\mathbf{X}'_i, \mathbf{Y}'_i) \mid (\mathbf{X}'_1, \mathbf{Y}'_1, \dots, \mathbf{X}'_{i-1}, \mathbf{Y}'_{i-1}) = v)$. Since $\mathbf{X}', \mathbf{Y}'$ are independent, $\mathbf{X}'_i, \mathbf{Y}'_i$ are independent, and fixing the first few bits of each random variable doesn't change this fact. Thus $(\mathbf{W}_i \mid \mathbf{W}_{<i} = v)$ is a random variable consisting of two independent bits. Let p denote the probability that its first bit is 1, and q denote the probability that its second bit is 1.

On the other hand, since $(\mathbf{S}, \mathbf{T})$ is uniform over pairs $(x, y) \sim \{0,1\}^n \times \{0,1\}^n$ of disjoint strings, it is straightforward to verify that each $\mathbf{R}_i \sim \{0,1\}^2$ is an independent random variable that is uniform over the strings $\{(0,0), (0,1), (1,0)\}$. Thus, $(\mathbf{R}_i \mid \mathbf{R}_{<i} = v)$ is uniform over the same set of strings. Now that we have specified the distributions of $(\mathbf{W}_i \mid \mathbf{W}_{<i} = v)$ and $(\mathbf{R}_i \mid \mathbf{R}_{<i} = v)$, it is a straightforward calculation to verify that there is a universal constant $\delta > 0.1$ such that no matter the values of p, q,

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

Thus, applying our lemma on amplifying statistical distance (Lemma 7), we get

$$|(\mathbf{X}', \mathbf{Y}') - (\mathbf{S}, \mathbf{T})| \geq 1 - e^{-n\delta^2/2}.$$

To conclude, recall that $(\mathbf{X}, \mathbf{Y})$ is a convex combination of $\leq 2^b$ distributions of the form $(\mathbf{X}', \mathbf{Y}')$, and thus applying Lemma 2 we get

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{S}, \mathbf{T})| \geq 1 - 2^b \cdot e^{-n\delta^2/2}.$$

Thus, as long as Alice and Bob communicated $b \leq \frac{n\delta^2}{4} \log_2 e = \Omega(n)$ bits, we get statistical distance at least $1 - 2^{-\frac{n\delta^2}{4} \log_2 e} = 1 - 2^{-\Omega(n)}$, as desired. □

7 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*, and prove the first nontrivial sampling lower bounds against oblivious read-once branching programs.

The overall study of the complexity of sampling is still a very new area, and many open questions remain. Below, we outline three such questions on the complexity of sampling with limited memory.

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, $\ell = \text{poly}(n)$)?

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 $\ell \leq \text{poly}(n)$.

Sampling lower bounds for more general branching programs On the other hand, if we keep the output bits layered, and the randomness unlimited, it is natural to ask if there exist more general types of branching programs for which sampling lower bounds can be established.

One such generalization might be to unknown-order ROBPs [FK18], which are allowed to read their input bits and write their output bits in any unknown order. As it turns out, many of our results can be easily extended to this more general setting: for example, our results on sampling codes can be extended to the unknown-order setting using the fact that distance is preserved under permutating coordinates; and our sampling lower bounds against input-output pairs $(\mathbf{U}_n, b(\mathbf{U}_n))$ (from Appendix A.2) can be extended by replacing the two-source extractor with an extractor for interleaved sources (e.g., like the one from [RY11]).

Perhaps a more interesting generalization would be to read-k branching programs, where the branching program is allowed to read each input bit k times. It seems much more challenging to break the correlation between the output bits of a read-k branching program - thus, establishing sampling lower bounds for this more general setting would seem to require significantly new techniques.

A separation between sampling with ROBPs and AC^0 circuits It is not too hard to show that there exist simple distributions samplable by AC^0 circuits that cannot be sampled by ROBPs, even given exponential width (see Appendix A.3). Can one find a distribution that cannot be sampled in 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, 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 [CG21], new extractors are needed.

8 Acknowledgements

We thank William Hoza and anonymous reviewers for extremely helpful comments.

References

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

[AST$^+$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.

[CG21] Eshan Chattopadhyay and Jesse Goodman. Improved extractors for small-space sources. To Appear in the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2021.

[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.

[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.

[Kil88] Joe Kilian. Founding cryptography 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.

[Yao82] Andrew C Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pages 80–91. IEEE, 1982.

A ROBP sampling results from lower bounds in communication complexity

As it turns out, many natural questions about sampling with ROBPs can be answered using known (or easy-to-prove) results in communication complexity. Using this connection, it is straightforward to show that (i) sampling is easier than computing for ROBPs (answering Question 1); and (ii) there exist explicit boolean functions $b : \{0,1\}^n \to \{0,1\}$ such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ is hard to sample for ROBPs (answering Question 2). Moreover, this connection can also be leveraged to exhibit very simple distributions $\mathbf{Q} \sim \{0,1\}^n$ that are *extremely* hard to sample for ROBPs.

We give formal statements and proofs of the above results in Appendix A.1, Appendix A.2, and Appendix A.3, respectively. We will also observe how these sampling lower bounds imply complexity separations between sampling with ROBPs and sampling with AC^0 circuits. But before we proceed with this, we review the connection between ROBPs and communication protocols, and outline why communication lower bounds imply lower bounds against ROBPs.

Simulating ROBPs with two-party communication protocols It is well known that ROBPs of width 2^s can be simulated by two-party protocols that communicate s bits. In fact, *one-way* communication protocols suffice. In a one-way two-party communication protocol, Alice and Bob are given inputs $x \in \{0,1\}^a, y \in \{0,1\}^b$ respectively, and they try to compute some function $f : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}$. Alice is allowed to send a single message to Bob, after which Bob must immediately determine the value $f(x,y)$ (using just his input y and the message from Alice).

Observe that for any ROBP $f : \{0,1\}^n \to \{0,1\}$ of width 2^s, and any $a, b \in \mathbb{N}$ that sum to n, it holds that a one-way communication protocol (where Alice is given a bits, Bob is given b bits, and s bits are communicated) can exactly compute f. To see why, simply note that Alice can use her a bits to simulate the first a steps of the ROBP, send the state at which she arrives to Bob (using s bits), which he can then use to complete the computation.

Thus, if a function $g : \{0,1\}^n \to \{0,1\}$ cannot be computed by one-way protocols that communicate s bits, then it also cannot be computed by ROBPs of width 2^s. More generally, if all one-way protocols over s bits fail to compute g on a $1/2-\varepsilon$ fraction of its inputs, then the same must be true for all ROBPs of width 2^s.

Analogously, it is not difficult to show that if a distribution $\mathbf{Q} \sim \{0,1\}^n$ can be sampled by an ROBP of width 2^s, then it can also be sampled by a two-party communication protocol that communicates $s+1$ bits. Recall (from Section 6) that in such a protocol, Alice is given private randomness $\mathbf{A}$, and Bob is given private randomness $\mathbf{B}$, which they use to communicate $s+1$ bits between one another. At the end of the protocol, Alice outputs some $\mathbf{X} \sim \{0,1\}^{n/2}$, Bob outputs some $\mathbf{Y} \sim \{0,1\}^{n/2}$, and the overall output of the protocol is defined as $(\mathbf{X}, \mathbf{Y})$.

To see why such communication protocols can sample the same distributions as ROBPs, first recall that any distribution $\mathbf{Q} \sim \{0,1\}^n$ sampled by a width 2^s ROBP can also be sampled by a KRVZ sampler (Definition 10) of width 2^{s+1}, as per Theorem 4. It is then easy to simulate a KRVZ sampler using two-party protocols: Alice can use her randomness $\mathbf{A}$ to simulate the first $n/2$ steps of the random walk over the KRVZ sampler, send the state at which she arrives to Bob (using $s+1$) bits, at which point Bob can simulate the remaining $n/2$ steps of the random walk using his randomness $\mathbf{B}$. At the end, Alice can output the bits she saw on her random walk, and Bob can do the same.

Thus, if a distribution $\mathbf{Q} \sim \{0,1\}^n$ cannot be sampled by two-party protocols that communicate $s+1$ bits, then it also cannot be sampled by ROBPs of width 2^s. More generally, if any distribution $\mathbf{Q}' \sim \{0,1\}^n$ sampled by such a protocol has statistical distance $\geq \delta$ from $\mathbf{Q}$, then the same must be true for any distribution generated by an ROBP of width 2^s.

A.1 Sampling is easier than computing

Using the above connection, it is now easy to show that sampling is easier than computing for ROBPs: there is some explicit function $b : \{0,1\}^n \to \{0,1\}$ such that ROBPs of exponential width cannot compute b (even on average), yet they can easily sample $(\mathbf{U}_n, b(\mathbf{U}_n))$ using just linear width. The latter part will be straightforward to show, and the former part will follow immediately from known communication lower bounds. We thank an anonymous reviewer for a concise proof of these communication lower bounds, which is included (and slightly optimized) in the proof below.

Theorem 15. *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any $\varepsilon > 0$, the following holds. For every ROBP $F : \{0,1\}^n \to \{0,1\}$ of width at most $2^{\frac{n\varepsilon^2}{9} - \log(1/\varepsilon)}$ it holds that*

$$\Pr_x[F(x) = b(x)] < \frac{1}{2} + \varepsilon,$$

but there exists an ROBP $G : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $2n$ (and length $\ell \le n + \log n + 2$) such that

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

Proof. Let $k \in \mathbb{N}$ be any power of 2, and define $n := k + \log k$. We take $b : \{0,1\}^n \to \{0,1\}$ as the well-known *address* (or *index*) function $\mathsf{address} : \{0,1\}^k \times [k] \to \{0,1\}$, defined as

$$\mathsf{address}(x, i) := x_i.$$

We first prove the second part of the theorem. Towards this end, define $\mathbf{X} \sim \{0,1\}^k$ and $\mathbf{Y} \sim [k]$ as independent random variables that are uniform over their respective domain, and note that $(\mathbf{U}_n, b(\mathbf{U}_n)) = (\mathbf{X}, \mathbf{Y}, \mathbf{X}_\mathbf{Y})$. Now, for any $y \in [k]$ and $b \in \{0,1\}$, define the random variable $\mathbf{X}^{y \leftarrow b} \sim \{0,1\}^k$ to have all of its bits independent and uniform, except for the y^{th} bit, which is fixed to b. Furthermore, define the random variable $\mathbf{Y}^{y \leftarrow b} := (\mathbf{X}^{y \leftarrow b}, y, b)$. It is straightforward to verify that $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be written as the following convex combination:

$$(\mathbf{U}_n, b(\mathbf{U}_n)) = (\mathbf{X}, \mathbf{Y}, \mathbf{X}_\mathbf{Y}) = \sum_{y \in [k], b \in \{0,1\}} \frac{1}{2k} \cdot \mathbf{Y}^{y \leftarrow b}. \tag{8}$$

It is not difficult to show that $\mathbf{Y}^{y \leftarrow b}$ can be sampled by an ROBP of width 1 and length $k + \log k + 1 = n + 1$. Let $\mathcal{B}^{y \leftarrow b}$ be the ROBP that samples it. Then, to sample the entire convex combination written above, we can create an ROBP $\mathcal{B}$ as follows. First, construct a complete binary tree of depth d satisfying $2^d = 2k$ (recall that k is a power of 2). For any node in the tree, assign its outgoing edges the input labels $0, 1$, and assign to them no output labels. Identify the $2k$ leaves of the tree with the set $[k] \times \{0,1\}$, and for each $y \in [k], b \in \{0,1\}$, attach $\mathcal{B}^{y \leftarrow b}$ to the leaf (y, b): more formally, identify leaf (y, b) with the start vertex of $\mathcal{B}^{y \leftarrow b}$. This completes the construction of $\mathcal{B}$.

It is straightforward to verify that $\mathcal{B}$ exactly samples the convex combination from Equation (8), and thus it exactly samples $(\mathbf{U}_n, b(\mathbf{U}_n))$. Furthermore, it has width $2k \le 2n = O(n)$ and length $\ell = d + (n+1) = \log(2k) + n + 1 \le n + \log n + 2 = O(n)$, as desired.

We now prove the first part of the theorem, using (a slightly optimized version of) a proof suggested by an anonymous reviewer. This part of the theorem follows immediately from known lower bounds on computing the address function using one-way communication protocols. We prove these lower bounds, below.

Suppose there is a one-way communication protocol that computes a function $f : \{0,1\}^k \times [k] \to \{0,1\}$ such that $f(x,i) = \mathsf{address}(x,i)$ on at least $\frac{1}{2} + \varepsilon$ fraction of the possible pairs x, i. In other words,

$$\Pr_{\substack{x \sim \{0,1\}^k \\ i \sim [k]}} [f(x,i) = \mathsf{address}(x,i)] \geq \frac{1}{2} + \varepsilon.$$

Furthermore, suppose that in this protocol, Alice is given $x \in \{0,1\}^k$ and Bob is given $i \in [k]$. Let s be the maximum number of bits that Alice sends to Bob on any input. The goal is to lower bound s.

First, note that Alice's message to Bob will always be in the set $\mathcal{M} := \{0,1\} \cup \{0,1\}^2 \cup \cdots \cup \{0,1\}^s$, and thus there are $< 2^{s+1}$ possible messages she could send to Bob. For each message $m \in \mathcal{M}$, let $S_m \subseteq \{0,1\}^k$ denote the set of all strings that cause Alice to send m to Bob. Note that $\{S_m\}_{m \in \mathcal{M}}$ partitions Alice's input space $\{0,1\}^k$.

Now, consider any $m \in \mathcal{M}$ such that

$$\Pr_{\substack{x \sim S_m \\ i \sim [k]}} [f(x,i) = \mathsf{address}(x,i)] > \frac{1+\varepsilon}{2}. \tag{9}$$

We will argue that $t := |S_m|$ must be quite small. To see why, first notice that by definition of one-way communication protocols, there must be some deterministic function $g : [k] \to \{0,1\}$ such that $f(x,i) = g(i)$ for all $x \in S_m$: this is because Bob's output only depends on the message he receives and his own input i, and all inputs $x \in S_m$ end up in Bob receiving the same message. Thus we have

$$\Pr_{\substack{x \sim S_m \\ i \sim [k]}} [g(i) = \mathsf{address}(x,i)] > \frac{1+\varepsilon}{2}. \tag{10}$$

Now, define $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_k) \sim \{0,1\}^k$ to be uniform over S_m, and define for each $i \in [k]$ the value

$$\mathsf{bias}(\mathbf{Y}_i) := |\Pr[\mathbf{Y}_i = 1] - \Pr[\mathbf{Y}_i = 0]|.$$

For any fixed i, it is easy to verify that $\Pr_{x \sim S_m}[g(i) = \mathsf{address}(x,i)] \leq (1 + \mathsf{bias}(\mathbf{Y}_i))/2$, since $g(i)$ is a fixed value. Thus we have

$$\frac{1+\varepsilon}{2} < \Pr_{\substack{x \sim S_m \\ i \sim [k]}} [g(i) = \mathsf{address}(x,i)] \leq \sum_{i \in [k]} \frac{1}{k} \cdot \frac{1 + \mathsf{bias}(\mathbf{Y}_i)}{2} = \frac{1 + \sum_i \mathsf{bias}(\mathbf{Y}_i)/k}{2},$$

which implies

$$\sum_{i \in [k]} \mathsf{bias}(\mathbf{Y}_i) > k\varepsilon. \tag{11}$$

Now, letting $H_{\mathsf{Sh}}(\cdot)$ denote Shannon entropy, it is a straightforward calculation to show that for any random variable $\mathbf{Z} \sim \{0,1\}$, it holds that $H_{\mathsf{Sh}}(\mathbf{Z}) \leq 1 - \mathsf{bias}(\mathbf{Z})^2/2$. Using this fact, we have

$$\log t = H_{\mathsf{Sh}}(\mathbf{Y}) \leq \sum_{i \in [k]} H_{\mathsf{Sh}}(\mathbf{Y}_i) \leq \sum_{i \in [k]} \left(1 - \frac{\mathsf{bias}(\mathbf{Y}_i)^2}{2}\right) = k - \frac{1}{2} \sum_{i \in [k]} \mathsf{bias}(\mathbf{Y}_i)^2. \tag{12}$$

Now, note that by Cauchy-Schwarz, it holds that $(\sum_i \mathsf{bias}(\mathbf{Y}_i))^2 \leq k \sum_i \mathsf{bias}(\mathbf{Y}_i)^2$. Combining this with Equations (11) and (12), we have:

$$\log t \leq k - \frac{1}{2}\sum_{i\in[k]} \mathsf{bias}(\mathbf{Y}_i)^2 \leq k - \frac{1}{2k}\left(\sum_i \mathsf{bias}(\mathbf{Y}_i)\right)^2 < k - \frac{1}{2k}(k\varepsilon)^2 = k\cdot(1-\varepsilon^2/2).$$

At last, we can conclude that for any $m \in \mathcal{M}$ such that Equation (9) holds, it must also be true that $|S_m| = t < 2^{k\cdot(1-\varepsilon^2/2)}$. Now, let $p_m := \Pr_{x\sim S_m, i\sim[k]}[f(x,i) = \mathsf{address}(x,i)]$, and observe that

$$\begin{aligned}
\frac{1}{2}+\varepsilon &\leq \Pr_{x\sim\{0,1\}^k, i\sim[k]}[f(x,i) = \mathsf{address}(x,i)] \\
&= \sum_{m\in\mathcal{M}} p_m \cdot \Pr_{x\sim\{0,1\}^k}[x\in S_m] \\
&= \sum_{m\in\mathcal{M}:p_m\leq\frac{1+\varepsilon}{2}} p_m \cdot \Pr_{x\sim\{0,1\}^k}[x\in S_m] + \sum_{m\in\mathcal{M}:p_m>\frac{1+\varepsilon}{2}} p_m \cdot \Pr_{x\sim\{0,1\}^k}[x\in S_m] \\
&\leq \frac{1+\varepsilon}{2}\cdot \sum_{m\in\mathcal{M}:p_m\leq\frac{1+\varepsilon}{2}} \Pr_{x\sim\{0,1\}^k}[x\in S_m] + \sum_{m\in\mathcal{M}:p_m>\frac{1+\varepsilon}{2}} 1\cdot 2^{-k}\cdot 2^{k\cdot(1-\varepsilon^2/2)}. \\
&\leq \frac{1+\varepsilon}{2}\cdot 1 + |\mathcal{M}|\cdot 2^{-k\varepsilon^2/2} \\
&< \frac{1+\varepsilon}{2} + 2^{s+1-k\varepsilon^2/2},
\end{aligned}$$

which implies that

$$s > \frac{k\varepsilon^2}{2} - \log(1/\varepsilon) - 2,$$

Thus, we can finally conclude that any one-way communication protocol that computes address on $\geq \frac{1}{2}+\varepsilon$ of its inputs must use $s > \frac{k\varepsilon^2}{2} - \log(1/\varepsilon) - 2 \geq \frac{n\varepsilon^2}{3} - \log(1/\varepsilon) - 2$ bits of communication. In other words, for any function $f : \{0,1\}^k \times [k] \to \{0,1\}$ computed by a one-way communication protocol over $s \leq \frac{n\varepsilon^2}{3} - \log(1/\varepsilon) - 2$ bits, it holds that

$$\Pr_{x,i}[f(x,i) = \mathsf{address}(x,i)] < \frac{1}{2}+\varepsilon. \tag{13}$$

By our discussion at the beginning of Appendix A, it immediately follows that for any ROBP $f : \{0,1\}^n \to \{0,1\}$ of width $2^s \leq 2^{\frac{n\varepsilon^2}{3}-\log(1/\varepsilon)-2}$, Equation (13) also holds. Furthermore, this must also hold for any $2^s \leq 2^{\frac{n\varepsilon^2}{9}-\log(1/\varepsilon)}$: this trivially holds when $n\varepsilon^2/9 < 1$ (any ROBP of width $2^s < 2$ must have width at most 1, and such ROBPs can only compute the address function with probability half), and when $n\varepsilon^2/9 \geq 1$ we have $2^{\frac{n\varepsilon^2}{9}-\log(1/\varepsilon)} \leq 2^{\frac{n\varepsilon^2}{3}-\log(1/\varepsilon)-2}$. This completes the proof. □

A.2 Sampling lower bounds against input-output pairs

In the previous section, we gave an explicit function $b : \{0,1\}^n \to \{0,1\}$ that is hard to compute (even on average) for ROBPs of exponential width, but such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ is *easy to sample* for ROBPs of just linear width. Next, we show how to find an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ is *hard to sample* for ROBPs. We prove the following.

Theorem 16. *There exists a universal constant $c > 0$ and an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any $\ell = \ell(n)$ and ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width at most 2^{cn},*

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

In particular, for ROBPs of width $2^{\Omega(n)}$, we get sampling lower bounds against input-output pairs of the form $\frac{1}{2} - 2^{-\Omega(n)}$. In order to prove this result, 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 $(\mathbf{U}_n, b(\mathbf{U}_n))$ is hard to sample for two-party communication protocols when b is a two-source extractor. Then, we instantiate this lemma with a well-known explicit two-source extractor (the inner product function), and use the connection between ROBPs and communication protocols to obtain Theorem 16.

Lemma 10. *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 distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^{n+1}$ sampled by a two-party communication protocol with s bits of communication, it holds that*

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| \geq \frac{1}{2} - \varepsilon - 2^{s-n+k+5}$$

Proof. The distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^{2n+1}$ sampled by Alice and Bob 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. But as we have seen, we can fix the message sent from Alice to Bob in order to write $(\mathbf{X}, \mathbf{Y})$ as a convex combination of the form

$$(\mathbf{X}, \mathbf{Y}) = \sum_{i \in \mathcal{M}} p_i \cdot (\mathbf{A}^{(i)}, \mathbf{B}^{(i)}, \mathbf{b}^{(i)}),$$

where i runs over all possible message transcripts $\mathcal{M}$, and each $\mathbf{A}^{(i)} \sim \{0,1\}^n$ is independent from each $(\mathbf{B}^{(i)}, \mathbf{b}^{(i)}) \sim \{0,1\}^n \times \{0,1\}$. Consider now fixing each $\mathbf{b}^{(i)}$ to some $b \in \{0,1\}$. Then we can write

$$(\mathbf{X}, \mathbf{Y}) = \sum_{i \in \mathcal{M}, b \in \{0,1\}} p_{i,b} \cdot (\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b), \tag{14}$$

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 $S \subseteq \{0,1\}^{2n+1}$,

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

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

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

Furthermore, we define the set

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

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 $\Pr[(\mathbf{X}, \mathbf{Y}) \in S]$, 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 $i \in \mathcal{M}, b \in \{0,1\}$,

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

Since the set of messages $\mathcal{M} \subseteq \{0,1\} \times \{0,1\}^2 \times \cdots \times \{0,1\}^s$ has size $< 2^{s+1}$, we get

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

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

$$\Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S] = \Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in \mathsf{Bad}] \leq 2^{-2n} \cdot |\mathsf{Bad}| < 2^{-n+s+t+4}. \quad (15)$$

We now move towards lower bounding $\Pr[(\mathbf{X}, \mathbf{Y}) \in S]$. By Equation (14), 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

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

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

$$p \cdot \Pr[\mathsf{Ext}(\widetilde{\mathbf{A}}^{(i,b)}, \widetilde{\mathbf{B}}^{(i,b)}) \neq b] + (1-p). \quad (16)$$

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 (16) 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, 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 (16) is $\geq 1/2 - \varepsilon$ (given that we set $t := k + 1$). Thus in all cases, Equation (16) 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. By Equation (14), this therefore implies

$$\Pr[(\mathbf{X}, \mathbf{Y}) \in S] \geq \frac{1}{2} - \varepsilon, \tag{17}$$

as long as we set $t = k + 1$. Wrapping everything up, we combine Equation (15) and Equation (17) to get

$$\begin{aligned} |(\mathbf{X}, \mathbf{Y}) - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| &\geq \Pr[(\mathbf{X}, \mathbf{Y}) \in S] - \Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S] \\ &> \frac{1}{2} - \varepsilon - 2^{-n+s+t+4} \\ &= \frac{1}{2} - \varepsilon - 2^{-n+s+k+5} \end{aligned}$$

as desired. □

Now, in order to prove Theorem 16, We will instantiate Lemma 10 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. Let $n = 2\ell$ for some $\ell \in \mathbb{N}$, and let $\mathsf{Ext}\{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}$ be the inner product extractor from Theorem 17 for min-entropy k and error $\varepsilon = 2^{-(2k-\ell-1)/2}$. By Lemma 10 we know that for any distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^\ell \times \{0,1\}^{\ell+1}$ sampled by a two-party communication protocol with s bits of communication, it holds that

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{U}_n, \mathsf{Ext}(\mathbf{U}_n))| \geq \frac{1}{2} - \varepsilon - 2^{s-\ell+k+5} = \frac{1}{2} - 2^{-(2k-\ell-1)/2} - 2^{s-\ell+k+5}.$$

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

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{U}_n, \mathsf{Ext}(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{s-\ell/4+6} = \frac{1}{2} - 2^{s-n/8+6}$$

Finally, by our discussion at the beginning of this section (on the connection between ROBPs and communication protocols), we know that for any $\ell \in \mathbb{N}$ and ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $2^{s'}$,

$$|F(\mathbf{U}_n) - (\mathbf{U}_n, \mathsf{Ext}(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{s'-n/8+7}.$$

Picking a small enough constant $c > 0$ completes the proof. □

A separation between sampling with ROBPs and AC^0 circuits for input-output pairs We conclude this subsection by remarking that the above result gives an input-output distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ that cannot be sampled by exponential width ROBPs (even on average), but which can be sampled by AC^0. In particular, we can take b to be the inner product function, and combine the above result with the known result [IN96] that $(\mathbf{U}_n, \mathsf{IP}(\mathbf{U}_n))$ can be sampled in AC^0.

A.3 Simple distributions that are hard to sample

As a final application of communication lower bounds to ROBP lower bounds, we give a very simple distribution that is very hard to sample for ROBPs.

Theorem 18. *For any $\ell \in \mathbb{N}$ and any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{2n}$ of width at most $2^{n/6}$,*

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, \mathbf{U}_n)| \geq 1 - 16 \cdot 2^{-n/6}.$$

Proof. Let $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^n$ be a distribution sampled by a two-party communication protocol that communicates at most s bits. As we have seen, by fixing the message that Alice sends to Bob, we get that $(\mathbf{X}, \mathbf{Y})$ is a convex combination of $< 2^{s+1}$ distributions of the form $(\mathbf{X}', \mathbf{Y}') \sim \{0,1\}^n \times \{0,1\}^n$, where $\mathbf{X}', \mathbf{Y}'$ are independent. We consider any such $\mathbf{X}', \mathbf{Y}'$, and lower bound $|(\mathbf{X}', \mathbf{Y}') - (\mathbf{U}_n, \mathbf{U}_n)|$.

Towards this end, let t be a parameter we fix later, and let

$$\mathsf{BAD} := \left\{(s, s) \in \{0,1\}^n \times \{0,1\}^n : \Pr[\mathbf{X}' = s] \geq 2^{-t} \text{ or } \Pr[\mathbf{Y}' = s] \geq 2^{-t}\right\}.$$

Notice that $|\mathsf{BAD}| \leq 2 \cdot 2^t$, and furthermore for every $(s,s) \notin \mathsf{BAD}$ it holds that $\Pr[(\mathbf{X}', \mathbf{Y}') = (s,s)] < 2^{-2t}$. We let $S := \{(s,s) : s \in \{0,1\}^n\}$, and note that by definition of statistical distance we have

$$\begin{aligned}|(\mathbf{X}', \mathbf{Y}') - (\mathbf{U}_n, \mathbf{U}_n)| &\geq \Pr[(\mathbf{U}_n, \mathbf{U}_n) \in S - \mathsf{BAD}] - \Pr[(\mathbf{X}', \mathbf{Y}') \in S - \mathsf{BAD}] \\ &> 1 - 2^{-n} \cdot |\mathsf{BAD}| - 2^{-2t} \cdot |S - \mathsf{BAD}| \\ &\geq 1 - 2^{-n+t+1} - 2^{-2t+n}.\end{aligned}$$

We set $t = \frac{2n-1}{3}$ to equalize both exponents to $\frac{-n+2}{3}$, which yields

$$|(\mathbf{X}', \mathbf{Y}') - (\mathbf{U}_n, \mathbf{U}_n)| > 1 - 2^{\frac{-n+5}{3}}.$$

Combining this with our lemma on convex combinations (Lemma 2), we get

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{U}_n, \mathbf{U}_n)| \geq 1 - 2^{s+1} \cdot 2^{\frac{-n+5}{3}} = 1 - 2^{\frac{-n+8}{3}+s} \geq 1 - 2^{-n/3+s+3}. \tag{18}$$

As we have seen in the beginning of Appendix A, for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{2n}$ of width $w = 2^{s'}$, the distribution $F(\mathbf{U}_\ell)$ can be sampled by a two-party communication protocol that uses $s' + 1$ bits of communication. Combining this with Equation (18), we that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{2n}$ of width s', it holds that

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, \mathbf{U}_n)| \geq 1 - 2^{-n/3+s'+4},$$

which is at least $1 - 16 \cdot 2^{-n/6}$ for $s' \leq n/6$, as desired. □

A strong separation between sampling with ROBPs and AC^0 circuits We conclude this section by remarking that the above result provides a very strong separation between AC^0 circuits and ROBPs for the task of sampling, since AC^0 can clearly sample the distribution $(\mathbf{U}_n, \mathbf{U}_n)$.

B Equivalence theorems

In this section, we prove the equivalence theorems claimed in Section 3.3.

B.1 An equivalence theorem between KRVZ 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 KRVZ 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 19 (Theorem 4, restated)**.** *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 KRVZ sampler of width $2w$ that samples $\mathbf{X}$.*
- *If there exists a KRVZ 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 KRVZ 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 19 as the following lemma.

Lemma 11. *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 KRVZ sampler of width $2w$ that samples $\mathbf{X}$.*

To prove this result, we will need to show that KRVZ samplers can efficiently simulate ROBP samplers. In other words, we will need to transform an ROBP sampler into a KRVZ 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 19 will be more challenging to prove. Here, we will need to show that ROBP samplers can efficiently simulate KRVZ samplers; or rather, that any KRVZ 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 KRVZ 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 KRVZ sampler into an ROBP sampler, we would first like to transform the KRVZ sampler into a *granular* KRVZ 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 12 ([KRVZ11, Lemma 8.4])**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be a KRVZ sampler of width w. For any $\alpha = 1/A$ with $A \in \mathbb{N}$, there exists an α-granular KRVZ 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 KRVZ sampler into a granular KRVZ sampler, the next step will be to transform this granular KRVZ 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 19.

Lemma 13 (Lemma 3, restated)**.** *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists a (2^{-t})-granular KRVZ 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 13 is the result mentioned earlier, which strengthens the second bullet of Theorem 19 when the KRVZ sampler is granular. Given Lemmas 11 to 13, we can easily prove Theorem 19 as follows.

Proof of Theorem 19. The first bullet is clearly true by Lemma 11. For the second bullet: let $\mathbf{X} \sim \{0,1\}^n$ be any distribution that can be generated by a KRVZ 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 12, there is a 2^{-t}-granular KRVZ sampler $\mathbf{X}^* \sim \{0,1\}^n$ of width w that is $(2^{-t}nw \leq \varepsilon)$-close to $\mathbf{X}$. By Lemma 13, 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 11 and 13, then we are done. We prove these lemmas in the next two subsections. We start with Lemma 13, since its proof is more interesting than Lemma 11.

B.1.1 Proof of Lemma 13

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 13.

Lemma 14 (Lemma 4, restated)**.** *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists a (2^{-t})-granular KRVZ 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 13, 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 13 is usually more useful than Lemma 14. 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 14 over Lemma 13. For this reason, we also record the general version of Lemma 14, which can be obtained by combining it with Lemma 12 (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 KRVZ 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 19 that uses more width but less randomness.

The plan now is to prove Lemma 14, and then show how we can extend the intuitions developed in this proof to prove Lemma 13. In order to prove Lemma 14, we will transform a KRVZ sampler into an ROBP sampler in a vertex-by-vertex fashion. In particular, we will replace each vertex v in the KRVZ 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 will use a Σ-*ROBP*, as defined in Section 5.3. Next, we need a few more definitions before formalizing multi-threshold functions. 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 14, Lemma 13, and Theorem 19. That is, we construct a (near-optimal) Σ-ROBP for computing t-threshold functions.

Lemma 15 (Key ingredient for Theorem 19)**.** *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 KRVZ 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 5, 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 13 and 14.

We will prove Lemma 15 at the very end of this section. But first, we show how it can be used to prove Lemmas 13 and 14.

Proof of Lemma 14. We must show that if there is a 2^{-t}-granular KRVZ 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 KRVZ 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 $V_1 = \{v_1, \dots, v_w\}$. For each $i \in [w]$ and $b \in \{0,1\}$, let $p_{i,b}$ denote the probability that the KRVZ 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 15, 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 15, 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 KRVZ sampler $\mathcal{B}$ into an ROBP $\mathcal{B}'$. Let $G = (V, E)$ be the underlying graph of the KRVZ 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 13.

Proof of Lemma 13. We must show that if there is a 2^{-t}-granular KRVZ 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 14, we know that for any 2^{-t}-granular KRVZ 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 KRVZ 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 KRVZ sampler, with layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. In the proof to Lemma 14, 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 KRVZ 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 $\mathcal{S}, \mathcal{W}, \mathcal{T}$.[4] 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 KRVZ 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 KRVZ

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

sampler. Using the proof of Lemma 14, 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 KRVZ 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 KRVZ 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 KRVZ 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. □

At last, all that remains is to prove our result on computing multi-threshold functions in low width, Lemma 15. We do so below.

Proof of Lemma 15. 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. □

B.1.2 Proof of Lemma 11

We have now arrived at the final missing piece for Theorem 19. To prove Lemma 11, we must show that KRVZ samplers can simulate ROBP samplers using roughly the same width.

Proof of Lemma 11. 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 KRVZ 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 KRVZ 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 KRVZ 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 *KRVZ 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 KRVZ sampler edges, and delete all vertices that are not in $V_{a_1}, V_{a_2}, \ldots, V_{a_n}$. Thus we obtain a KRVZ 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 KRVZ sampler. Thus we have a KRVZ sampler $\mathcal{B}'$ of width $2w$ that samples $\mathbf{X}$. □

B.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 20 (Theorem 5, restated)**.** *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*.[5]

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

[5] 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. □

B.2.1 An extension to correlation bounds

Theorem 20 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 21 (Lemma 5, restated)**.** *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 20. 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. □

B.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 22 (Theorem 6, restated)**.** *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 $\mathbf{Q}$. □

B.3.1 An extension to covariance bounds

Theorem 22 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 21) since it is possible to get strong sampling lower bounds against $\mathbf{Q}$ even if 1_S is very biased.

Theorem 23 (Lemma 6, restated)**.** *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 $\mathrm{cov}(f, 1_S) \geq \varepsilon$. 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 22. 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 $\mathrm{cov}(f, 1_S) = \mathrm{corr}(f, 1_S) - \mathrm{bias}(f)\,\mathrm{bias}(1_S)$, and using the definitions of correlation and bias, it is a straightforward calculation to obtain

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

Now, notice that for any $x \in \mathrm{support}(\mathbf{X})$, it holds that $\Pr[\mathbf{X} = x] = 2^{-n}/a$. Similarly, for any $q \in \mathrm{support}(\mathbf{Q})$, 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:

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

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

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

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

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

as desired. □