Electronic Colloquium on Computational Complexity, Report No. 177 (2025)

Sampling Permutations with Cell Probes is Hard

Yaroslav Alekseev
Technion

Mika Göös
EPFL

Konstantin Myasnikov
EPFL

Artur Riazanov
EPFL

Dmitry Sokolov
EPFL & Université de Montréal

November 14, 2025

Abstract

Suppose we are given an infinite sequence of input cells, each initialized with a uniform random symbol from $[n]$. How hard is it to output a sequence in $[n]^n$ that is close to a uniform random permutation? Viola (SICOMP 2020) conjectured that if each output cell is computed by making d probes to input cells, then $d \geq \omega(1)$. Our main result shows that, in fact, $d \geq (\log n)^{\Omega(1)}$, which is tight up to the constant in the exponent. Our techniques also show that if the probes are *nonadaptive*, then $d \geq n^{\Omega(1)}$, which is an exponential improvement over the previous nonadaptive lower bound due to Yu and Zhan (ITCS 2024). Our results also imply lower bounds against succinct data structures for storing permutations.

Contents

1 Introduction 1
2 Techniques 5
3 Proof of the Main result 9
4 Lipschitz Decision Forests 17
5 Containment Lemma 22
6 Collision Lemma 25
7 Open questions 29
References 30

ISSN 1433-8092

1 Introduction

Randomly shuffling the elements of an array is one of the most basic primitives in randomized algorithms. It is a simple programming exercise to implement this in linear time [Dur64]. Doing it much faster, with a parallel algorithm, has been studied extensively [Rei85, MV91, CK00, Czu15]. In particular, array shuffling is possible even in constant-time in a parallel RAM model [Hag91].

An analogous problem in probability theory is the question of *card shuffling*, which dates back to Markov [Mar06]. For example, one of the long-standing challenges in card shuffling has been to determine how many *Thorp shuffles* (explained in Figure 1 below) are sufficient to produce a nearly uniform permutation over n cards. A sequence of works [Mor08, MT06, Mor09, Mor13] has culminated in a result showing that $O(\log^3 n)$ shuffles are enough. An interesting feature of this shuffle (which has found applications, e.g., in cryptography [MRS09]) is its *obliviousness*: the final position of each card can be computed by *accessing only a few bits of randomness* (formally, the shuffle is given by a shallow "switching network").

Cell-probe model. A widely-studied computational model that captures oblivious shuffling (and much more) is the *cell-probe model* [Yao81]. In this model, we are given a sequence of s input cells, each storing a symbol from $[n]$. A cell-probe algorithm then produces an output sequence in $[n]^m$ where each output cell is computed by making d probes (queries) to input cells. That is, the algorithm computes a function $f\colon [n]^s \to [n]^m$, where the i-th output cell f_i is computed by a depth-d arity-n decision tree. Such cell-probe algorithms are also called depth-d *decision forests*.

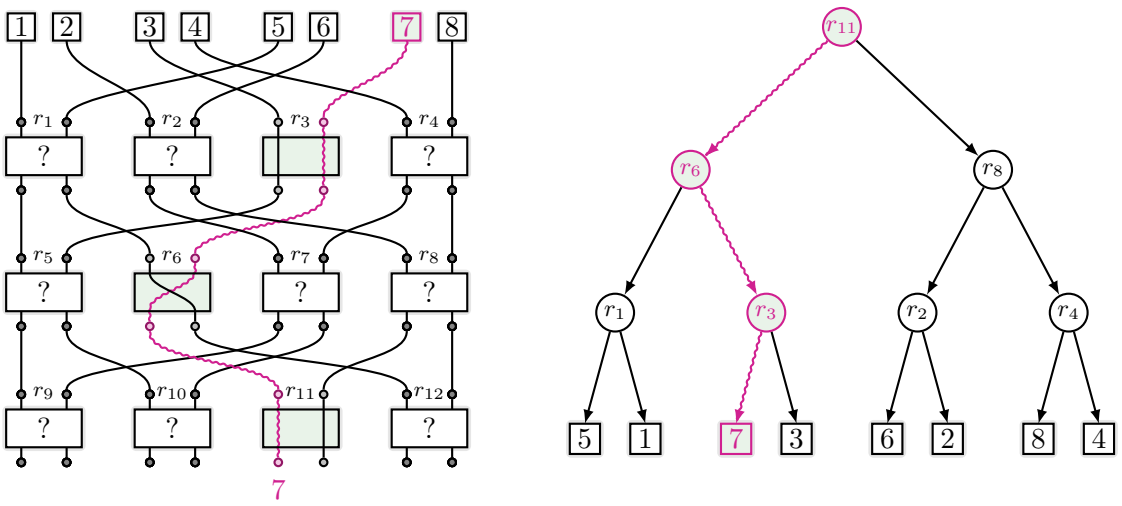

Figure 1: *(Left):* Three iterations of the Thorp shuffle [Tho73] as computed by a network of switches. In a single iteration, we take cards i and $n/2+i$ and place them in positions $2i-1$ and $2i$ in the order determined by a coin toss $r_j \in \{0,1\}$. *(Right):* The shuffle can be simulated by a cell-probe algorithm (aka decision forest) that probes the coin tosses r_j [Vio12b, Lemma 6.4]. Drawn here is the decision tree that finds the 5th output element.

Sampling permutations. We study cell-probe algorithms that solve *sampling* problems. Here we are given a uniform random input $\boldsymbol{u} \sim [n]^s$ (even for infinite $s = \mathbb{N} \coloneqq \{0, 1, 2, \ldots\}$) and the goal is to produce a prescribed output distribution $f(\boldsymbol{u})$. In this paper, we focus on producing an output distribution that is close to a uniform random permutation. More formally, we denote the set of permutations by $S_n \subseteq [n]^n$, a uniform random permutation by $\boldsymbol{\pi} \sim S_n$, and the statistical (total variation) distance between random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ by $\Delta(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \max_E |\Pr[\boldsymbol{x} \in E] - \Pr[\boldsymbol{y} \in E]|$.

Our research question becomes:

Question 1. What is the smallest depth d of a decision forest f such that $\Delta(\boldsymbol{\pi}, f(\boldsymbol{u})) \leq 1\%$?

Upper bounds on d follow from oblivious shuffles. Indeed, Figure 1 illustrates how d iterations of the Thorp shuffle can be simulated by a depth-d decision forest. It then follows from the aforementioned work on Thorp shuffle convergence that $d \leq O(\log^3 n)$ suffices. Better still, using an oblivious shuffle constructed by Czumaj [Czu15] one can obtain $d \leq O(\log^2 n)$. (In fact, Czumaj conjectures that even $d \leq O(\log n)$ is possible.)

Lower bounds on d are our main focus. The lower-bound question for sampling permutations with cell-probes was first raised by Viola [Vio20, §5], who conjectured that $d \geq \omega(1)$ is necessary. In an accompanying seminar talk [Vio18], Viola points out that, while $d = 1$ is easy to rule out, existing techniques in the sampling literature (surveyed in Section 1.2) do not even rule out $d = 2$, surprisingly enough. Our main result is to confirm Viola's conjecture by proving the first non-trivial lower bounds on d. Moreover, our lower bound turns out to match the upper bounds from oblivious shuffles up to polynomial factors.

Theorem 1 (Main result)**.** *Suppose that $f\colon [n]^{\mathbb{N}} \to [n]^n$ is a decision forest of depth $(\log n)^{1/2-\varepsilon}$ for some constant $\varepsilon > 0$. Then for $\boldsymbol{u} \sim [n]^{\mathbb{N}}$ and $\boldsymbol{\pi} \sim S_n$ we have*

$$\Delta(\boldsymbol{\pi}, f(\boldsymbol{u})) \geq 1 - \exp(n^{-\Omega(1)}).$$

The conclusion here gives a robust impossibility result, saying that the output of a shallow decision forest is extremely far from a uniform permutation. Such $1 - o(1)$ distance bounds are known to imply lower bounds against succinct data structures for storing permutations. We discuss these corollaries in Section 1.3.

Nonadaptive algorithms. En route to Theorem 1 we develop new lower-bound techniques that are also able to show improved lower bounds against *nonadaptive* algorithms. We say that a function $f\colon [n]^{\mathbb{N}} \to [n]^n$ is *k-local* if every output cell depends on at most k input cells. That is, every f_i makes at most k *nonadaptive* probes to the input cells. Viola [Vio20] proved that permutation sampling requires $\Omega(\log \log n)$ nonadaptive probes, and this was subsequently improved to $\tilde{\Omega}(\log n)$ by Yu and Zhan [YZ24]. Our second result gives another exponential improvement:

Theorem 2. *Suppose that $f\colon [n]^s \to [n]^n$ is a n^{ε}-local function for a constant $\varepsilon \leq 10^{-3}$. Then for $\boldsymbol{u} \sim [n]^s$ and $\boldsymbol{\pi} \sim S_n$ we have*

$$\Delta(\boldsymbol{\pi}, f(\boldsymbol{u})) \geq 1 - \exp(-n^{1-O(\varepsilon)}).$$

In particular, permutation sampling exhibits an $(\log n)^{O(1)}$-vs-$n^{\Omega(1)}$ separation between adaptive and nonadaptive cell-probe complexities. The only previous such separation for sampling problems was $O(1)$-vs-$\tilde{\Omega}(\log n)$ by Yu and Zhan [YZ24]. The sampling problem they considered was somewhat artificial (defined for the sole purpose of obtaining a separation), whereas permutation sampling is an extremely natural problem. It remains open to prove an $O(1)$-vs-$n^{\Omega(1)}$ separation.

1.1 Cell-probes vs. bit-probes

For sampling permutations, it is important that we allow the full power of the cell-probe model, with input cells coming from a large alphabet $[n]$ corresponding to $O(\log n)$-bit word length. If we only allowed *bit-probes*—that is, we restricted the input cells to be bits $\{0,1\}$—then sampling lower bounds would not meaningfully translate to data structure lower bounds (as discussed in Section 1.3). In fact, bit-probe lower bounds for permutations are trivial to prove: Intuitively, each symbol in a random permutation is uniform over $[n]$ and so requires $\Omega(\log n)$ bits to generate, and this intuition is easy to turn into a formal proof of an $\Omega(\log n)$ bit-probe lower bound.[1]

Many prior works have studied the bit-probe model. For example, by now, it is known that most distributions over $\{0,1\}^n$ that are *symmetric* (invariant under permuting coordinates) cannot be sampled with $O(1)$ bit-probes [Vio23, FLRS23, KOW24, KOW25]. The key difference between bit-probes and cell-probes is that the output of a cell-probe algorithm need not be k-local for any nontrivial k. Indeed, an n-ary decision tree of depth d may depend on n^{d-1} many input cells. This is why locality lower bounds even as high as $k \geq n^{\Omega(1)}$ do not rule out cell-probe algorithms with $d = 2$. Our results contribute new tools that get beyond this "locality barrier"; such techniques are quite rare as only the prior work [Vio23] (discussed below in Section 1.2) has specifically targeted the (adaptive) cell-probe model. Finally, we note that while there are even more powerful techniques to show sampling lower bounds against AC^0-circuits (a model stronger than cell-probes), these do not apply for permutations as they are easy to sample for AC^0 [Vio12b].

1.2 Other related work

Our results contribute to the systematic study of the complexity of sampling distributions. Classically, computational complexity seeks hard-to-compute functions. In a seminal work, Viola [Vio12b] proposed a program to find *hard-to-sample distributions* for various computational models. This program has been extremely fruitful for many areas, such as pseudorandom generators [Vio12a, LV12, BIL12], randomness extractors [Vio14, CZ16, CS16], error-correcting codes [SS24], and data-structure lower bounds [Vio12b, LV12, BIL12, Vio20, CGZ22, Vio23, YZ24, KOW24, KOW25]. Especially for quantum–classical separations, a distribution is a natural witnessing object, and a line of work [WP23, KOW24, GKM+25] has shown quantum advantage for sampling problems.

The complexity landscape for sampling looks quite different from that for computation. Optimistically, one might hope to relate the complexity of computing $f\colon \{0,1\}^n \to \{0,1\}$ in some circuit class $\mathcal{C}$ to the complexity of sampling the input–output pair $(\boldsymbol{u}, f(\boldsymbol{u}))$ for $\boldsymbol{u} \sim \{0,1\}^n$ with a (multi-output) circuit from $\mathcal{C}$. Sampling input–output pairs is never harder than computing the function, but sometimes it can be dramatically easier. The classical example is $\textsc{Xor}(x) \coloneqq x_1 \oplus \cdots \oplus x_n$. It is known since [FSS84] that Xor is hard to compute with an AC^0-circuit. On the other hand, $(\boldsymbol{u}, \textsc{Xor}(\boldsymbol{u}))$ can be sampled with an NC^0-circuit [Bab87], in fact, with a 2-local function

$$y_1, \ldots, y_n \mapsto y_1, y_1 \oplus y_2, y_2 \oplus y_3, \ldots, y_{n-1} \oplus y_n, y_n.$$

Kane, Ostuni, and Wu [KOW25] show that $(\boldsymbol{u}, \textsc{Xor}(\boldsymbol{u}))$ is the only non-trivial symmetric distribution that can be sampled with an NC^0-circuit. On the other hand, Viola [Vio12a] building on the permutation-sampling algorithm by [MV91, Hag91] shows that for every symmetric f input–output pairs can be approximately sampled with an AC^0-circuit.

The class AC^0 is the frontier for sampling lower bounds: The works [LV12, BIL12] show that sampling a uniform codeword from a good error-correcting code is hard for an AC^0-sampler. Vi-

[1] In a depth-d *binary* decision forest, each tree has range 2^d. Thus the support size of the output of a forest is only 2^{dn}, which is negligible compared to $|S_n| = n! = 2^{\Theta(n \log n)}$ when $d \leq o(\log n)$.

ola [Vio20] exhibits a function f whose input–output pairs cannot be sampled in AC^0 with distance significantly smaller than $1/2$.

Despite all this lower-bound progress for AC^0-sampling, our understanding of the class remains quite coarse. Studying cell-probe algorithms, a model intermediate between NC^0 and AC^0, can give us a more refined picture. The only prior work showing lower bounds specifically against (adaptive) cell-probe samplers is by Viola [Vio23]. He proves a *separator theorem*, which states that any cell-probe sampler f such that $f(\boldsymbol{u})$ is close to a uniform distribution over some set, we can restrict the domain of f to some set R such that $\boldsymbol{y} \coloneqq (f(\boldsymbol{u}) \mid \boldsymbol{u} \in R)$ retains most of its entropy, and the coordinates of $\boldsymbol{y}$ are *almost pairwise independent* (the marginal distribution of every two output cells is close to the product of the individual margins). He used this theorem to separate AC^0 and cell-probe algorithms for sampling problems. However, the separator theorem cannot immediately be used to prove lower bounds for sampling permutations: the coordinates of a permutation are almost pairwise independent, so there is no contradiction to $\boldsymbol{y}$ enjoying the same property.

1.3 Succinct data structure lower bounds

Our results have direct implications for *succinctly storing* permutations. A succinct data structure stores an object (for us, a permutation $\pi \in S_n$) with the number of bits close to the information theoretic minimum (for us, $\log n!$ bits), while supporting interesting queries. For permutations, it is natural to support querying the values $\pi(i)$ and possibly the inverses $\pi^{-1}(i)$. The paper [MRRS12] constructs a data structure for permutations with $\log n! + n/\log^{2-o(1)} n$ bits of memory that can answer both $\pi(i)$ and $\pi^{-1}(i)$-queries with $\tilde{O}(\log n)$ adaptive cell probes. For such a data structure, [Gol09] gives an almost matching lower bound.

Are there succinct data structures with better space/cell-probe complexity that only support $\pi(i)$-queries? The best lower bound for this problem before this work followed from the sampling lower bound of [YZ24]. They proved that every data structure for supporting $\pi(i)$-queries with $o(\log n/\log\log n)$ *nonadaptive* probes must use at least $\log n! + n^{1-o(1)}$ bits. By a very simple reduction from [Vio12b] (also in [Vio20]), our Theorems 1 and 2 imply the following.

Corollary 3. *Every data structure storing a permutation $\pi \in S_n$ and supporting $\pi(i)$-queries with*

- $(\log n)^{1/2-\varepsilon}$ *adaptive cell probes must use* $\log n! + n^{\Omega(1)}$ *bits of space;*
- n^{ε} *nonadaptive cell probes must use* $\log n! + n^{1-O(\varepsilon)}$ *bits of space.*

We emphasize that such a lower bound is not obviously true at the outset, as surprising constructions of succinct data structures exist. For example, for the *dictionary problem* of storing a set of key–value pairs, there exists a data structure that stores only a polylogarithmically many bits above the information-theoretic minimum with constant-time access [Yu20].

1.4 Structure

The rest of the paper is organized as follows: In Section 2 we review our techniques and give a full proof of Theorem 2. In Section 3 we give the proof of Theorem 1 modulo three important technical tools. The three subsequent sections establish those tools: Section 4 handles average Lipschitzness, the ingredient that differentiates Theorem 1 from Theorem 2; in Section 5 we prove a *containment lemma* that brings the statistical distance bounds exponentially close to 1; in Section 6 we prove a *collision lemma*, the main technical differentiator between Theorem 2 and the previous work [Vio20, YZ24]. We conclude with open question in Section 7.

2 Techniques

In this section we discuss the proof of our main result. We start by giving in Section 2.1 a proof of Theorem 2 with *constant statistical distance bound*, and introducing the technical ideas shared between both of the proofs. In Section 2.2 we overview the additional ingredients needed for Theorem 1. Finally in Section 2.3 we show how to boost the statistical distance bound to be exponentially close to 1 in both cases.

2.1 Proof of Theorem 2

In this section, we give a proof of Theorem 2 modulo two technical lemmas. The informal idea for both our proofs is the following dichotomy: if the (Shannon) entropy of $f(\boldsymbol{u})$ is low, then a random permutation $\boldsymbol{\pi} \sim S_n$ lands in the support of $f(\boldsymbol{u})$ with low probability, and if the entropy of $f(\boldsymbol{u})$ is high, then whp two of its symbols coincide, which never happens with a permutation. Here Shannon entropy of a random variable $\boldsymbol{x}$ is $\mathrm{H}(\boldsymbol{x}) \coloneqq \sum_{x \in \mathrm{supp}(\boldsymbol{x})} \Pr[\boldsymbol{x} = x] \log(1/\Pr[\boldsymbol{x} = x])$.

Low entropy case. If we only shoot for a *constant* statistical distance bound, we can get away with a very simple proof in this case without using any properties of the sampler apart from the entropy of $f(\boldsymbol{u})$.

Lemma 4. *Suppose that* $\mathrm{H}(\boldsymbol{x}) \le k$. *Then there exists a set* E *of size* 2^{2k} *such that* $\Pr[\boldsymbol{x} \in E] \ge 1/2$.

Proof. Let $p(x) \coloneqq \Pr[\boldsymbol{x} = x]$ be the probability function of $\boldsymbol{x}$ so that $\mathrm{H}(\boldsymbol{x}) = \mathbb{E}[\log(1/p(\boldsymbol{x}))]$. We get from Markov's inequality that $\Pr[\log(1/p(\boldsymbol{x})) \ge 2k] \le 1/2$. Thus for $E \coloneqq \{x \mid \log(1/p(x)) < 2k\}$ we have $\Pr[\boldsymbol{x} \in E] \ge 1/2$. Observe that $x \in E$ iff $p(x) > 2^{-2k}$, then since the total probability is at most 1 we get $|E| \le 2^{2k}$. □

We can now conclude the low-entropy case almost immediately. If $\mathrm{H}(f(\boldsymbol{u})) \le (n \log n)/4$, then by Lemma 4 we find E of size $n^{n/2}$ such that $\Pr[f(\boldsymbol{u}) \in E] \ge 1/2$. On the other hand $\Pr[\boldsymbol{\pi} \in E] \le |E|/n! = o(1)$, which implies $\Delta(\boldsymbol{\pi}, f(\boldsymbol{u})) \ge 1/2 - o(1)$.

High entropy case. We would like to show that if $\mathrm{H}(f(\boldsymbol{u})) > (n \log n)/4$, then some two symbols of $f(\boldsymbol{u})$ coincide whp. The first step is to show this in the case the coordinates of $f(\boldsymbol{u})$ are independent. We formalize this in the following *collision lemma*:

Lemma 5. *Let* $\boldsymbol{z}_1, \dots, \boldsymbol{z}_m$ *be independent random variables over* $[n]$ *such that* $\mathrm{H}(\boldsymbol{z}_1, \dots, \boldsymbol{z}_m) \ge (m \log n)/8$ *for* $m \ge n^{0.99}$. *Then* $\Pr[\exists i \ne j \in [m] \colon \boldsymbol{z}_i = \boldsymbol{z}_j] \ge 1 - o(1)$.

This lemma (proved in Section 6) and its generalizations will be one of the main technical ingredients in the proof of Theorem 1 as well.[2]

Bounded influence. The remaining piece of the proof is the intermediate notion between local functions and a collection of independent output cells: we say that $f \colon [n]^s \to [n]^n$ is (ℓ, k)-local if it is k-local and *every input cell affects only* ℓ *output cells*. We make two simple observations. The first makes a step from k-locality to (k^2, k)-locality, and the second makes a step from (ℓ, k)-locality to independence.

[2] It might seem that Lemma 5 is a standard birthday paradox. Usually, the proofs of such results go as follows: split $\boldsymbol{z}$ into halves $\boldsymbol{z}_1, \dots, \boldsymbol{z}_{m/2}$ and $\boldsymbol{z}_{m/2+1}, \dots, \boldsymbol{z}_m$, fix $\boldsymbol{z}_{\le m/2} = \alpha$ and show that $\boldsymbol{z}_j \in A = \{\alpha_1, \dots, \alpha_{m/2}\}$ with noticeable probability for many $j > m/2$. Then apply Hoeffding's inequality. Observe that this is false, since it could happen that $\mathrm{supp}(\boldsymbol{z}_j) \cap A = \emptyset$ for all $j > m/2$.

(O1) For every k-local f the distribution $f(\boldsymbol{u})$ is a mixture of $n^{n/k}$ distributions $f^\alpha(\boldsymbol{u})$ where f^α is (k^2, k)-local. Indeed, let $I \subseteq [s]$ be the set of inputs in f that influence more than k^2 output cells. Since the number of input–output cell pairs such that the output cell depends on the input is at most $n \cdot k$, the size of I is at most n/k. Hence for every $\alpha \in [n]^I$, fixing the inputs in I according to α yields f^α.

(O2) For every (ℓ, k)-local function f there is a set $J \subseteq [n]$ of size $n/(\ell k)$ of output cells such that $\{f_j(\boldsymbol{u})\}_{j \in J}$ are independent. Indeed, populate J greedily: add an output cell $j \in [n]$ to J and "delete" all other output cells that share an input with j, and repeat this until all output cells are deleted. At each step at most $k\ell$ output cells are deleted and one of them is added to J, so $|J| \geq n/(\ell k)$.

Proof of Theorem 2 with constant distance. We will show that in the setting of Theorem 2 we have $\Delta(f(\boldsymbol{u}), \boldsymbol{\pi}) \geq 1/2 - o(1)$. We first apply (O1) to get a set $I \subseteq [s]$ of size $n^{1-\varepsilon}$ and a collection $\{f^\alpha\}_{\alpha \in [n]^I}$ of $(n^{2\varepsilon}, n^\varepsilon)$-local functions such that $(f(\boldsymbol{u}) \mid \boldsymbol{u}_I = \alpha) \equiv f^\alpha(\boldsymbol{u})$. Consider an arbitrary $\alpha \in [n]^I$. The function $f^\alpha \colon [n]^{[s] \setminus I} \to [n]^n$ defined by restricting the inputs in I according to α is still n^ε-local and each input cell affects at most $n^{2\varepsilon}$ output cells.

We now apply the entropy dichotomy for $f^\alpha(\boldsymbol{u})$:

1. *Low entropy case:* $\mathrm{H}(f^\alpha(\boldsymbol{u})) \leq (n \log n)/4$. Then by Lemma 4 there exists an event E_α such that $\Pr[f^\alpha(\boldsymbol{u}) \in E_\alpha] \geq 1/2$ with $|E_\alpha| \leq n^{n/2}$.

2. *High entropy case:* $\mathrm{H}(f^\alpha(\boldsymbol{u})) \geq (n \log n)/4$. Then using the chain rule for Shannon entropy, we have $\sum_{i \in [n]} \mathrm{H}(f^\alpha_i(\boldsymbol{u})) \geq \mathrm{H}(f^\alpha(\boldsymbol{u})) \geq (n \log n)/4$. Therefore, there exists a set $J \subseteq [n]$ of size $n/8$ such that for each $j \in J$ we have $\mathrm{H}(f^\alpha_j(\boldsymbol{u})) \geq \log n/8$. Now apply (O2) to find J' of size at least $n^{1-3\varepsilon}/8 \geq n^{0.99}$ such that $f^\alpha_j(\boldsymbol{u})$ are independent for $j \in J'$. Applying Lemma 5 to $f^\alpha_{J'}(\boldsymbol{u})$ we get that $\Pr[\exists i \neq j \in J' \colon f_i(\boldsymbol{u}) = f_j(\boldsymbol{u}) \mid \boldsymbol{u}_I = \alpha] \geq 1 - o(1)$.

Finally we are ready to define an event that witnesses the statistical distance between $f(\boldsymbol{u})$ and $\boldsymbol{\pi}$. Suppose $L \subseteq [n]^I$ is the set of assignments α such that the entropy of $f^\alpha(\boldsymbol{u})$ is low. Then define $F \coloneqq ([n]^n \setminus \mathrm{supp}(\boldsymbol{\pi})) \cup \bigcup_{\alpha \in L} E_\alpha$. That is, F is the event that the output sequence has a collision or belongs to one of the container events for the low-entropy case. Then $\Pr[\boldsymbol{\pi} \in F] \leq n^{|I|} \cdot n^{n/2}/n! = o(1)$. On the other hand by the total probability law

$$\begin{aligned}\Pr[f(\boldsymbol{u}) \in F] \geq{}& \Pr[\boldsymbol{u}_I \in L] \cdot \Pr\left[f(\boldsymbol{u}) \in \bigcup_{\alpha \in L} E_\alpha \mid \boldsymbol{u}_I \in L\right] \\ &+ \Pr[\boldsymbol{u}_I \notin L] \cdot \Pr[f(\boldsymbol{u}) \text{ has a collision} \mid \boldsymbol{u}_I \notin L] \\ \geq{}& 1/2.\end{aligned}$$

Comparison to [Vio20]. The idea of low-vs-high entropy dichotomy was originally introduced in [Vio20] to get $\Omega(\log \log n)$ locality lower bound for sampling permutations. The main reason our lower bound is much stronger is Lemma 5. In [Vio20] the dichotomy was established for a much stronger notion of entropy. That caused the low-entropy case to be much more complicated, rendering Lemma 4 inapplicable.

2.2 What is missing for the adaptive case?

The proof of Theorem 1 follows the same high-level recipe, but the steps are much more involved. The proof of the nonadaptive case has three steps:

(Step 1) Fix some of the input cells so that the rest affect few output cells.
(Step 2) If the output has low entropy, argue that it is significantly contained in a small set.
(Step 3) If the output has high entropy, argue that there is likely a collision by greedily choosing output cells that do not have common inputs.

The constant-probability version of (Step 2) does not suffer from the introduction of adaptivity, however the distance-boosting (see Section 2.3) in the adaptive case gets more involved. (Step 1) and (Step 3) break completely even in the constant-distance regime. The issue is that even a depth-2 decision forest can have arbitrary locality, and every input cell may influence every output cell. So, the key change is to devise an alternative intermediate notion between bounded-depth and independence.

Average Lipschitzness. Although low-depth decision trees are not local, they are local for every particular input. Similarly we can ask that a decision forest has bounded influence *in expectation*:

Definition 1 ([BIL12]). Let $f\colon [n]^s \to [n]^n$ be a decision forest. Let $\boldsymbol{\theta}_j$ denote the number of trees in f that query j on the input $\boldsymbol{u} \sim [n]^s$. We then say that f is *average-μ-Lipschitz* if $\mathbb{E}[\boldsymbol{\theta}_j] \leq \mu$ for every $j \in [s]$. We say that f is (μ, δ)-Lipschitz if $\Pr[\boldsymbol{\theta}_j > \mu] \leq \delta$ for every $j \in [s]$.

The high-level plan of the proof of Theorem 1 is to implement (Step 1) and (Step 3) above with average Lipschitzness replacing bounded influence. Both adaptations come with challenges: In (Step 1) fixing input cells that have high average Lipschitzness might *increase* the average Lipschitzness for other input cells. In (Step 3) the greedy choice of independent output cells as in (O2) fails completely[3], since average Lipschitzness offers no *global* independence properties.

Average Lipschitzness implies Lipschitzness almost everywhere. At first, it seems that (μ, δ)-Lipschitzness (or Lipschitzness *almost everywhere*) is a much stronger property than average Lipschitzness. Remarkably, it turns out that these properties are almost equivalent:

Lemma 6. *If $f\colon \Lambda^s \to \Sigma^m$ is an average-μ-Lipschitz depth-d decision forest, then for every $\varepsilon > 0$ the forest f is $(3\mu d^2 \log(1/\varepsilon), \varepsilon)$-Lipschitz.*

This lemma constitutes the crucial structural property of Lipschitz forests that we exploit in our proofs. It implies that many functions of the output of a decision forest concentrate well around their expectations. In particular, we show this for the conditional entropy (Section 4.2), and the Hamming distance to a set (Section 5).

The Lipschitz property of decision forests (with a different notation) was used very successfully by Beck, Impagliazzo, and Lovett [BIL12][4]. They prove [BIL12, Theorem 1.7] a result very similar (but incomparable) to Lemma 6. Unfortunately, we cannot use their result directly, as it only implies $(\omega(\sqrt{s}), \cdot)$-Lipschitzness and we need the first parameter to be polynomially small in n.

Establishing average Lipschitzness. In the adaptive case we solve the issue with (Step 1) by fixing the inputs *adaptively*. We exhaustively fix inputs that violate the condition $\mathbb{E}[\boldsymbol{\theta}_j] \leq \mu$ and observe that *in expectation over the value we fix the j-th input cell to*, this reduces the *average depth* of the whole decision forest by μ. Then we analyze the stopping time of the input fixing process to say that on average it terminates in nd/μ steps for a depth-d decision forest. See Section 4.3 for the proof.

[3] We remark that the greedy approach *must fail*, since we do have depth-$O(\log^2 n)$ decision forests sampling permutations [Czu15, Vio20].

[4] They asked in Open Problem 1 if their results could be used elsewhere. To the best of our knowledge, our work is the first such usecase.

Entropy-retaining depth reduction. For the nonadaptive case in (Step 3) we used that after the restriction the input–output cell dependency graph has low degree. In the adaptive case, even after establishing Lipschitzness almost everywhere, we still have potentially pairwise-dependent output cells. We resolve this by further restricting the inputs *and removing some output cells* in order to reduce the depth of the forest. The key challenge here is to establish that after such restriction the *entropy rate* of the remaining output cells does not decrease much. Specifically we will show that such procedure only decreases the entropy rate by a factor $1 - O(1/d)$ when reducing the depth of the forest from d to $d - 1$. Hence, after we are done, we are going to have some m output cells that are computed with a depth-1 decision forest and with entropy $\Omega(m \log n/d)$. Depth-1 decision forest is nonadaptive, so we can again choose independent output cells greedily. Our procedure shrinks the number of output cells by a $\mathrm{poly}(\log n)$-factor at each step, so in order to apply (generalized) Lemma 5 we need $\mathrm{poly}(\log n)^d = o(n^{1/d})$, hence we choose $d = o(\sqrt{\log n}/\log\log n)$.

In order to show that whp over the input restrictions the entropy is retained we show that conditional Shannon entropy $\mathrm{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{\alpha})$ concentrates around $\mathrm{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I)$ wrt the random choice of $\boldsymbol{\alpha} \sim [n]^I$ when f is an average-Lipschitz decision forest. The corresponding property is almost immediate for *min-entropy* regardless of the structure of the random variable $f(\boldsymbol{u})$. However, for us it is crucial to work with Shannon entropy, since assuming that min-entropy is low does not imply any *global* properties of the distributions, so the analogue of Lemma 4 for min-entropy is completely false.

2.3 Boosting distance

Theorem 2 and Theorem 1 both claim that the statistical distance from a cell-probe sampler to $\boldsymbol{\pi}$ is exponentially close to 1. This bound is crucial for the application to succinct data structures. In this section, we show how to boost the distance in the nonadaptive case.

In the high-entropy case we actually do not need any changes, since Lemma 5 implies that probability of *not* seeing a collision in a high-entropy collection of independent variables is exponentially low. Thus, it remains to address the low-entropy case where we want to show that for a $(n^{2\varepsilon}, n^{\varepsilon})$-local function g there exists an event E such that $|E| \le n^{3n/4}$ and $\Pr[g(\boldsymbol{u}) \in E] \ge 1 - \exp(-n^{1-10\varepsilon})$.

Let F be the event given by Lemma 4: $\Pr[g(\boldsymbol{u}) \in F] \ge 1/2$ and $|F| \le n^{n/2}$. The main idea is to consider a *neighborhood* of F as the new witnessing event: $\mathcal{N}_k(F) \coloneqq \{x \mid \min_{y \in F} \mathrm{dist}(x, y) \le k\}$, where $\mathrm{dist}(\cdot,\cdot)$ is the (n-ary) Hamming distance. The probability bound $\Pr[g(\boldsymbol{u}) \in F] \ge 1/2$ is equivalent to $|g^{-1}(F)| \ge n^s/2$. If $x \in \mathcal{N}_k(g^{-1}(F))$ then $g(x) \in \mathcal{N}_{k \cdot n^{2\varepsilon}}(F)$, since changing one symbol in the input to g changes at most $n^{2\varepsilon}$ output cells. We then need to choose k such that $\mathcal{N}_k(g^{-1}(F))$ contains almost all points of $[n]^s$, yet $E \coloneqq \mathcal{N}_{k \cdot n^{2\varepsilon}}(F)$ is small as shown in Figure 2.

For the former, we use a fact from combinatorics, that is essentially due to Harper [Har66].

Theorem 7. *For an arbitrary set $S \subseteq [n]^s$ and $\boldsymbol{u} \sim [n]^s$ we have*

$$\Pr[\boldsymbol{u} \in \mathcal{N}_k(S)] \ge 1 - \frac{\exp(-k^2/(2s \log n))}{\Pr[\boldsymbol{u} \in S]}.$$

Proof. For the boolean alphabet ($n = 2$) the claim is shown in [McD89, Proposition 7.7]. We give a simple reduction to this case. Consider the natural bijection $b\colon [n]^s \to \{0,1\}^{s \log n}$ that encodes every $[n]$-symbol as $\log n$ bits. Then $\Pr[\boldsymbol{u} \in \mathcal{N}_k(S)] \ge \Pr[b(\boldsymbol{u}) \in \mathcal{N}_k(b(S))]$, which implies the claim. □

In order to achieve high probability in Theorem 7 we choose $k = n^{1-4\varepsilon}$. Wlog we may assume

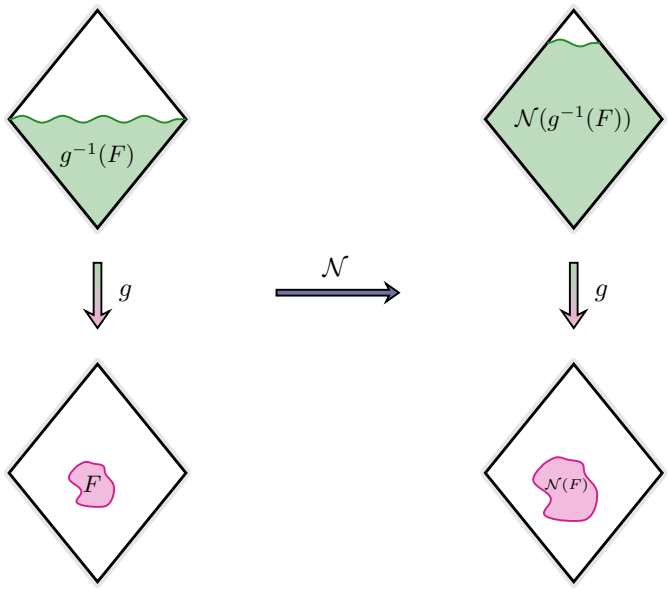

Figure 2: The picture illustrates the process of boosting the distance from 1/2 to almost 1 by expanding the witnessing event to its neighborhood.

that $s \le n^{1+\varepsilon}$ for a n^{ε}-local function. Hence we get $\Pr[\boldsymbol{u} \in \mathcal{N}_k(g^{-1}(F))] \ge 1 - \exp(-n^{1-9\varepsilon}/2 \log n)/2$. On the other hand $|\mathcal{N}_{k \cdot n^{2\varepsilon}}(F)| \le |F| \cdot \binom{s}{k \cdot n^{2\varepsilon}} n^{kn^{2\varepsilon}} \le n^{n/2} \cdot \exp(3k \cdot n^{2\varepsilon} \log s) \le n^{n/2} \cdot n^{3n^{1-2\varepsilon}} \ll n^{3n/4}$.

Boosting distance in the adaptive case: containment lemma. The problem with the proof of (Step 2) above is that in the adaptive case the input space might have the dimension up to n^{d+1}, which prevents us from using Theorem 7 to boost the error probability: the radius of the neighborhood that we would have to use is at least $\sqrt{n^{d+1}}$, which is *larger than the dimension of the output space*. Thus we prove a dimension-free version of Theorem 7 for sets that can be recognized by bounded-depth decision trees:

Theorem 8 (simplified version of Lemma 30)**.** *Suppose that $T\colon [n]^s \to \{0,1\}$ is a decision tree of depth d such that $|T^{-1}(1)| \ge \mu \cdot n^s$. Then*

$$\mathop{\mathbb{E}}_{\boldsymbol{u} \sim [n]^s} [\operatorname{dist}(\boldsymbol{u}, T^{-1}(1))] = O(\sqrt{d \log(1/\mu)}).$$

For every depth-d decision forest *every property* of its output can be computed with a depth-nd decision tree, so with Theorem 8 we can reduce the effective dimension to nd. On the other hand, the expectation bound we get is not quite enough to establish the exponentially low probability of being outside the neighborhood. Thus, in Section 5 we boost Theorem 8 using average Lipschitzness.

3 Proof of the Main result

In this section, we give the proof of the lower bound for the adaptive case and formally introduce all the technical tools needed.

3.1 Warm-up: the case of bucketed queries

We start by proving the theorem in the special case where the queries of the decision forest are structured: every tree takes its first query from a set I_1, the second query from the set I_2, and so

on $[s] = I_1 \sqcup \cdots \sqcup I_d$. We say that such a forest is *bucketed*. The goal of this section is to present the high-level structure of the general proof and introduce the primary technical tools that will be utilized throughout. We remark that instead of the domain $[n]^{\mathbb{N}}$ used in the intro, we use a domain $[n]^s$ for some integer s: a depth-d decision forest (bucketed or not) consisting of n trees can query at most n^{d+1} distinct input cells, so we may assume wlog that the indices of these cells are $[s]$.

Theorem 9. *Suppose $f\colon [n]^s \to [n]^n$ is a bucketed depth-$o(\log n/\log\log n)$ decision forest with buckets $I_1, \dots, I_d$ and let $\boldsymbol{u}_i \sim [n]^{I_i}$ for each $i \in [d]$. Let $\boldsymbol{\pi} \sim S_n$ be the uniform random permutation. Then*

$$\Delta(f(\boldsymbol{u}_1, \dots, \boldsymbol{u}_d), \boldsymbol{\pi}) \geq 1 - \exp(-n^{\Omega(1)}).$$

As announced in Section 2.2, on the high level, this proof follows the plan of the proof of Theorem 2 with bounded influence replaced with bounded average Lipschitzness (see Definition 1). Let us first consider two cases: when $\mathrm{H}(f(\boldsymbol{u})) \geq n\log n/4$ and when $\mathrm{H}(f(\boldsymbol{u})) \leq n\log n/4$.

High entropy case. Suppose that $\mathrm{H}(f(\boldsymbol{u})) \geq n\log n/4$. Ideally, in this case we want to show that probability of a collision is

$$\Pr[\exists i \neq j \in [n]\colon f_i(\boldsymbol{u}) = f_j(\boldsymbol{u})] \geq 1 - \exp(-n^{\Omega(1)}),$$

and thus, almost always, we will not generate a permutation. The main idea is to reduce the problem to the case where the output cells of f are independent, and apply a collision lemma similar to Lemma 5. Here we will use the following version:

Lemma 10. *Let $\boldsymbol{u} \sim [n]^s$ and $f\colon [n]^s \to [n]^n$ be a depth-1 decision forest. Suppose that $\mathrm{H}(f(\boldsymbol{u})) \geq 4(n\log\log n)$. Then $\Pr[\exists i \neq j \in [n]\colon f_i(\boldsymbol{u}) = f_j(\boldsymbol{u})] \geq 1 - \exp(-\Omega(n/\mathrm{poly}(\log n)))$.*

Observe that with a stronger guarantee on the entropy this would follow from Theorem 2, since depth-1 decision forests are 1-local functions. We prove this lemma in Section 6.2.

In the bucketed case the reduction from a depth-d to a depth-1 decision forest is quite straightforward: *fix all input buckets, except for one.* Thus, we need to find a bucket I_i such that after fixing inputs in all other buckets the output of the resulting forest *typically* retains some entropy. The following simple fact says that after such conditioning the entropy is retained *in expectation*:

Fact 1. *Suppose $\boldsymbol{x}_1, \dots, \boldsymbol{x}_\ell$ are independent and $\boldsymbol{y} = f(\boldsymbol{x}_1, \dots, \boldsymbol{x}_\ell)$ then $\mathrm{H}(\boldsymbol{y}) \leq \sum_{i\in[\ell]} \mathrm{H}(\boldsymbol{y} \mid \boldsymbol{x}_{[\ell]\setminus i})$.*

The proof of this fact is a simple chain rule computation and can be found in Section 3.3. We then find that there exists $i \in [d]$ such that

$$\mathrm{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_{[d]\setminus i}) \geq (n\log n/4)/d \gg n\log\log n. \tag{1}$$

As observed before, for any $\beta \in [n]^s$ and $j \in [d]$, function $x \mapsto f(\beta_{[s]\setminus I_j}, x_{I_j})$ can be computed with a depth-1 decision forest, hence if we proved that

$$\Pr_{\boldsymbol{\beta}\sim[n]^s}\left[\mathrm{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_{[d]\setminus i} = \boldsymbol{\beta}_{[s]\setminus I_i}) \geq 4(n\log\log n)\right] \geq 1 - \exp(-n^{\Omega(1)}), \tag{2}$$

we would be able to immediately apply Lemma 10 and finish the proof in the high-entropy case.

Unfortunately, (1) does not imply the probability lower bound by itself. However, for an *average Lipschitzness* forest f, retaining a high entropy under restriction *typically* is implied by having a high conditional entropy:

Lemma 11. *Let $f\colon [n]^s \to \Sigma^m$ be an average-$n^{0.3}$-Lipschitz depth-$\log^{O(1)} n$ decision forest. Suppose that $m \le n$ and $|\Sigma| = O(n)$. Let $I \subseteq [s]$ be a subset of the input cells. Then for $\boldsymbol{r}$ and $\boldsymbol{u}$ uniformly distributed over $[n]^s$ we have*

$$\Pr\left[\left|\mathrm{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{r}_I) - \mathrm{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I)\right| > n^{0.9}\right] \le \exp(-n^{\Omega(1)}).$$

We prove this lemma in Section 4.2. Combined with (1) this lemma implies (2) finishing the proof in the high-entropy case for average Lipschitz functions. We will see how to get rid of this assumption shortly.

Low entropy case. Now suppose that $\mathrm{H}(f(\boldsymbol{u})) \le n \log n/4$. Observe that here, analogously to the nonadaptive case in Theorem 2 the would have been resolved with Lemma 4 had we aimed for the constant distance bound. For the strong bounds we need to boost the distance. To this end, we prove the following lemma in Section 5:

Lemma 12. *Suppose $f\colon [n]^s \to [n]^m$ is an average-$n^{0.1}$-Lipschitz depth-d decision forest with $m \ge n^{0.99}$, $d = \log^{O(1)} n$, $\boldsymbol{u} \sim [n]^s$, and $\mathrm{H}(f(\boldsymbol{u})) \le m \log n/4$. Then there exists a set $F \subseteq [n]^m$ of size $n^{3m/4}$ such that*

$$\Pr[f(\boldsymbol{u}) \in F] \ge 1 - \exp(-n^{\Omega(1)}).$$

Final step: enforcing average Lipschitzness. The only obstacle that remains for the proof of Theorem 9 is the failure of average Lipschitzness for f. Similar to the nonadaptive case we enforce it by restricting the values of input cells that are queried too many times in expectation.

We need some additional notation to state this result. Each leaf of a decision tree can be naturally identified with a partial assignment $\ell \in ([n] \cup \{\star\})^s$, where the non-$\star$ symbols correspond to the input cells queried in the path to the leaf and $\star$-symbols correspond to all the remaining symbols. A random leaf $\boldsymbol{\ell}$ of a decision tree is defined as the leaf reached by the computation on the random input $\boldsymbol{u} \sim [n]^s$. For a decision forest $f\colon [n]^s \to [n]^m$ and a partial assignment ℓ, $f|_\ell$ denotes the decision forest f after restricting the input cells according to ℓ.

Lemma 13. *Suppose $f\colon [n]^s \to [n]^m$ is an arbitrary depth-d decision forest. There exists a $2nd/\mu \cdot \log(1/\varepsilon)$-depth decision tree T querying symbols of $[n]^s$ such that for a random leaf $\boldsymbol{\ell}$ of T, $f|_{\boldsymbol{\ell}}$ is average-μ-Lipschitz with probability $1 - \varepsilon$.*

We then can derive a simple corollary (along the lines of the proof of Theorem 2) that immediately implies Theorem 9.

Corollary 14. *Suppose that $f\colon [n]^s \to [n]^n$ is such that for any partial assignment $\ell \in ([n] \cup \{\star\})^s$ that makes $f|_\ell$ average-$n^{0.1}$-Lipschitz, and for $\boldsymbol{u} \sim [n]^s$ we have either*

- (Collision): $\Pr[\exists i \ne j \in [n]\colon (f|_\ell)_i(\boldsymbol{u}) = (f|_\ell)_j(\boldsymbol{u})] \ge 1 - \exp(-n^{\Omega(1)})$.
- (Containment): *There is a set F_ℓ of size $n^{3n/4}$ such that* $\Pr[f|_\ell(\boldsymbol{u}) \in F_\ell] \ge 1 - \exp(-n^{\Omega(1)})$.

Then $\Delta(f(\boldsymbol{u}), \boldsymbol{\pi}) \ge 1 - \exp(-n^{\Omega(1)})$, where $\boldsymbol{\pi} \sim S_n$.

Proof. We apply Lemma 13 with $\varepsilon = \exp(-n^{.01})$ and $\mu = n^{.1}$ to get a depth-$n^{.99}$ decision tree. Consider a leaf ℓ of this tree. By the assumption for each ℓ such that $f|_\ell$ is average-μ-Lipschitz, we either get a collision with probability $1 - \exp(-n^{\Omega(1)})$, or there exists a set F_ℓ of size at most $n^{3n/4}$ such that $\Pr[f(\boldsymbol{u}) \in F_\ell \mid \boldsymbol{u} \in \ell] \ge 1 - \exp(-n^{\Omega(1)})$. The total number of leaves is bounded by $n^{n^{.99}} \ll n^{n/20}$. Thus, the size of the union of all F_ℓ over all leaves that fall in the containment case is at most $n^{n/20} \cdot n^{3n/4} \le n^{4n/5}$.

Finally, we see that with probability $1-\exp(-n^{\Omega(1)})$ a uniformly random $\boldsymbol{u}$ lands in a leaf $\boldsymbol{\ell}$ such that $f|_{\boldsymbol{\ell}}$ is average-$n^{.1}$-Lipschitz. If that happens, then either the sequence that we sample does not constitute a permutation, or it belongs to $\bigcup F_\ell$. Given that $\Pr[\boldsymbol{\pi} \in \cup_\ell F_\ell] \le n^{4n/5}/n! = \exp(-\Omega(n))$, this immediately implies that $\Delta(f(\boldsymbol{u}), \boldsymbol{\pi}) \ge 1 - \exp(-n^{\Omega(1)})$. □

3.2 The general proof

In this section we finalize the proof of Theorem 1. The only missing structural piece is the treatment of the high-entropy case. It is formalized as follows:

Lemma 15. *Suppose $f\colon [n]^s \to [n]^m$ is a depth-d average-μ-Lipschitz decision forest such that $d = o(\sqrt{\log n}/\log\log n)$, $\mu = n^{0.1}$, and $m = \Omega(n)$. Suppose that $\mathrm{H}(f(\boldsymbol{u})) \ge m \log n/4$. Then $\Pr[\exists i \ne j \in [m]\colon f_i(\boldsymbol{u}) = f_j(\boldsymbol{u})] \ge 1 - \exp(-n^{\Omega(1)})$.*

Assuming this lemma, we can finish the proof.

Proof of Theorem 1. Consider a partial assignment $\ell \in ([n]\cup\{\star\})^s$, suppose that $f|_\ell$ is average-$n^{0.1}$-Lipschitz. Then either $\mathrm{H}(f|_\ell(\boldsymbol{u})) \ge n\log n/4$, in which case we have a collision whp by Lemma 15, or $\mathrm{H}(f|_\ell(\boldsymbol{u})) \le n\log n/4$, so we have a small container set by Lemma 12. We now apply Corollary 14 and finish the proof. □

We now proceed to the proof of Lemma 15.

3.2.1 Sharper tools

So far, we have stated two simplified versions of our collision lemma: Lemma 5 and Lemma 10. The general adaptive case requires a stronger version (proved in Section 6):

Lemma 16. *Let $\boldsymbol{u} \sim [n]^s$ and $f\colon [n]^s \to ([n]\cup\{\perp\})^m$ be a depth-1 decision forest. Suppose that $\mathrm{H}(f(\boldsymbol{u})) \ge \delta \cdot m \log n$, and $(\delta^2/4)m \ge n^{1-\varepsilon}$ for $\delta = \delta(n) \ge \max(4\log\log n/\log n, 8\varepsilon)$. Then*

$$\Pr[\exists i \ne j \in [m]\colon f_i(\boldsymbol{u}) = f_j(\boldsymbol{u}) \ne \perp] \ge 1 - \exp(-\Omega(\delta^4 m^3/n^2)).$$

In the warm-up section we hid under the rug the need to use (μ,δ)-Lipschitzness, instead of average Lipschitzness. Although by Lemma 6 we can always assume average Lipschitzness, operating with almost-everywhere Lipschitzness directly comes in handy in the general proof, because the latter is preserved under restrictions:

Lemma 17. *Suppose $f\colon \Lambda^s \to \Sigma^m$ is a (μ,δ)-Lipschitz decision forest. Let T be a decision tree querying symbols of a string in Λ^s. For $\boldsymbol{\alpha}$ be a leaf of T where a uniformly random assignment lends, and let $f|_{\boldsymbol{\alpha}}$ be the forest with the input cells restricted according to $\boldsymbol{\alpha}$. Then*

$$\Pr[\ f|_{\boldsymbol{\alpha}} \text{ is } (\mu,\sqrt{\delta})\text{-Lipschitz}\ ] \ge 1 - \sqrt{\delta}.$$

Proof. Fix some $j \in [s]$. Let $Q(\alpha, x)$ be the number of trees in $f|_\alpha$ that query the cell j on the input x. Let E be the event "$f|_{\boldsymbol{\alpha}}$ is *not* $(\mu,\sqrt{\delta})$-Lipschitz". Suppose for contradiction that $\Pr[E] > \sqrt{\delta}$. Consequently we get

$$\Pr_{\boldsymbol{\alpha},\boldsymbol{u}}[Q(\boldsymbol{\alpha},\boldsymbol{u}) > \mu] \ge \Pr[E]\cdot\Pr[Q(\boldsymbol{\alpha},\boldsymbol{u}) > \mu \mid E] > \sqrt{\delta}\cdot\sqrt{\delta} = \delta.$$

Since $Q(\alpha, x)$ is also the number of trees in f that query the cell j on the joint input (α, x) we have a contradiction with (μ,δ)-Lipschitzness of f. □

A helpful trick for dealing with Lipschitz decision forests is to turn an unlikely undesirable event (say an input cell is queried too many times) into an impossible event by terminating the computation of a tree if it is about to do something undesirable (e.g. query that too popoular input cell). The following fact helps to argue that such terminations do not affect entropy too much.

Fact 2. *Let $\boldsymbol{x}$ be a random variable supported over $(\Sigma \cup \{\perp\})^n$ and let $b\colon (\Sigma \cup \{\perp\})^n \to \mathbb{Z}_{\geq 0}$ be the function that counts non-$\perp$ elements in the input. Then $\mathrm{H}(\boldsymbol{x}) \leq \log(n+1) + \mathbb{E}[b(\boldsymbol{x})] \log(n|\Sigma|)$.*

Proof. Observe that $\mathrm{H}(\boldsymbol{x} \mid b(\boldsymbol{x}) = \ell) \leq \ell \log(n|\Sigma|)$, since the support size of $(\boldsymbol{x} \mid b(\boldsymbol{x}) = \ell)$ is at most $(n|\Sigma|)^\ell$. We then write by the chain rule:

$$\begin{aligned}
\mathrm{H}(\boldsymbol{x}) &= \mathrm{H}(b(\boldsymbol{x}), \boldsymbol{x}) \\
&= \mathrm{H}(b(\boldsymbol{x})) + \mathrm{H}(\boldsymbol{x} \mid b(\boldsymbol{x})) \\
&= \mathrm{H}(b(\boldsymbol{x})) + \mathop{\mathbb{E}}_{\boldsymbol{\ell} \sim b(\boldsymbol{x})} [\mathrm{H}(\boldsymbol{x} \mid b(\boldsymbol{x}) = \boldsymbol{\ell})] \\
&\leq \mathrm{H}(b(\boldsymbol{x})) + \mathop{\mathbb{E}}_{\boldsymbol{\ell} \sim b(\boldsymbol{x})} [\boldsymbol{\ell} \log(n|\Sigma|)] \\
&\leq \log(n+1) + \mathbb{E}[b(\boldsymbol{x})] \cdot \log(n|\Sigma|).
\end{aligned}$$

□

3.2.2 Handling the high-entropy case.

In this section, we prove Lemma 15. The proof proceeds by repeatedly reducing the depth of the forest until the depth is 1, so we are in a position to apply Lemma 16.

In what sense do we reduce the depth? We will find a set $I \subseteq [s]$ and a set $J \subseteq [m]$ such that whp over the assignment to I the projection f_J after assigning the input cells in I has high entropy and depth at most $d-1$. This is formalized in the following key lemma:

Lemma 18. *Suppose $f\colon [n]^s \to ([n] \cup \{\perp\})^m$ is a depth-d average-μ-Lipschitz decision forest with $m \geq n^{0.99}$, and $\mu \leq n^{0.3}$. Suppose that for $\boldsymbol{u} \sim [n]^s$ we have $\mathrm{H}(f(\boldsymbol{u})) \geq cm \log n$ with $c = \omega(1/\log n)$. Then there exists a set $I \subseteq [s]$, a set $J \subseteq [m]$ of size $|J| \geq m/\log^6(n)$ and a forest $g\colon [n]^s \to ([n] \cup \{\perp\})^J$ such that for every $j \in J$ and all $x \in [n]^s$ we have $g_j(x) \in \{f_j(x), \perp\}$ and one of the following conditions holds:*

(C1) With probability $1 - \exp(-n^{\Omega(1)})$ over $\boldsymbol{\beta} \sim [n]^I$, $\mathrm{H}(g(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{\beta}) \geq (1 - 3/d)c|J| \log n$ and g has depth $d-1$ after any assignment to the input cells in I.

(C2) With probability $1 - \exp(-n^{\Omega(1)})$ over $\boldsymbol{\beta} \sim [n]^I$, $\mathrm{H}(g(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{\beta}) \geq (1/(3d))c|J| \log n$ and g has depth 1 after any assignment to the input cells in I.

Proof of Lemma 15 given Lemma 18. We apply Lemma 6 to get that for every $\delta > 0$ we have that f is $(3\mu d^2 \log(1/\delta), \delta)$-Lipschitz. We take $\delta := \exp(-\mu)$, so f is $(3(\mu d)^2, \delta)$-Lipschitz. Then we will iterate Lemma 18 until the restricted f is depth-1 according to the following algorithm.

Input: $f\colon [n]^s \to ([n] \cup \{\perp\})^m$
Output: $f'\colon [n]^R \to ([n] \cup \{\perp\})^J$ of depth 1. A partial assignment $\boldsymbol{\alpha}$ to $[n]^s$

1: Let $f' \leftarrow f$ with $R \leftarrow [s]$ and $J \leftarrow [m]$.
2: $\boldsymbol{\alpha}$ be an empty assignment.
3: **while** Depth of f' is larger than 1 **do**
4: Apply Lemma 18 to get $g\colon [n]^R \to ([n] \cup \{\perp\})^{J'}$, $J' \subseteq J$ and $I \subseteq R$.
5: Sample $\boldsymbol{\beta} \sim [n]^I$.
6: $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} \cup \boldsymbol{\beta}$.
7: Update $f' \leftarrow g|_{\boldsymbol{\beta}}$; $R \leftarrow R \setminus I$; $J \leftarrow J'$.

8: **if** f' is not $((3\mu d)^2, \sqrt{\delta})$-Lipschitz **then**
9: **fail**
10: **end if**
11: **end while**

By Lemma 17 the line (9) is executed at any point in the algorithm with probability at most $\sqrt{\delta}$ over $\boldsymbol{\alpha}$. With probability $(1-\exp(-n^{\Omega(1)}))^d$ over the random assignment $\boldsymbol{\alpha}$ the entropy of $f'(\boldsymbol{u})$ satisfies for some $j \in [d]$

$$\frac{\mathrm{H}(f'(\boldsymbol{u}))}{|J|\log n} \geq \prod_{i=j}^{d}(1-O(1/i)) \cdot \Omega(1/j) \geq \Omega(1/d) = \omega(\log\log n/\sqrt{\log n}).$$

On the other hand $|J| \geq m(\log^6 n)^{-d} = n \cdot 2^{o(\sqrt{\log n})}$.

Now we reduced the problem to the case $d = 1$, so we can apply Lemma 16 to get that f_J has a non-$\perp$ collision with probability $1-\exp(-n^{\Omega(1)})$ over $\boldsymbol{u} \sim [n]^R$. Hence with probability $(1-\sqrt{\delta})(1-\exp(-n^{\Omega(1)}))$ there is a collision in $f(\boldsymbol{\alpha}, \boldsymbol{u})$ as required. □

3.2.3 Proof of Lemma 18

The proof goes as follows:

1. We are going to identify sets $I \subseteq [s]$ and $J \subseteq [m]$ such that the *first queries* in trees f_J always come from I and only some $o(|J|)$ trees query I after their first query whp over the input.
2. We will prune the trees f_J into g_J such that g_J *never* query I after the first query at the expense of sometimes returning $\perp$.
3. By Fact 1 we have that either $\mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_I) \geq \mathrm{H}(g_J(\boldsymbol{u})) \cdot (1-1/d)$ or $\mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_{[s]\setminus I}) \geq \mathrm{H}(g_J(\boldsymbol{u})) \cdot (1/d)$, so in the former case (C1) is satisfied and in the latter (C2) is.

In the steps (1) and (2) we need to make sure that the entropy *rate* of $g_J(\boldsymbol{u})$ does not drop too much compared to the entropy *rate* of $f(\boldsymbol{u})$. In the step (3) the key is to use the fact that the conditional entropy *concentrates* i.e. not only conditioning on a *random variable* $\boldsymbol{u}_I$ does not reduce it too much, but conditioning on the event $\boldsymbol{u}_I = \boldsymbol{\beta}$ also does not reduce it too much whp over $\boldsymbol{\beta} \sim \boldsymbol{u}_I$.

First step: isolating first queries. First, we are going to choose some trees in f_i such that the set of input cells I that are queried first by f_i is unlikely to be queried as a non-first query by *any* of the chosen trees.

Let us partition $f_1, \ldots, f_m$ into subsets $J_1 \sqcup \cdots \sqcup J_\ell = [m]$ such that trees indexed with J_i first query the input cell i (wlog we may assume that the first queries form the set $[\ell]$). By the Lipschitzness assumption we have that $|J_i| \leq 2\mu$ for every $i \in [\ell]$, hence $\ell \geq m/(2\mu)$. Now let $\boldsymbol{I} \subseteq [\ell]$ be a random set where each element from $[\ell]$ is included independently with probability $\alpha = 1/\log^6 n$. Let $\boldsymbol{J} \coloneqq \bigcup_{i \in \boldsymbol{I}} J_i$ be the set of output cells that first query an input cell from $\boldsymbol{I}$.

We would like to argue that the expectation over $\boldsymbol{I}$ of the expectation over $\boldsymbol{u}$ of the number of non-first queries $f_{\boldsymbol{J}}$ make to $\boldsymbol{I}$ is low, see Figure 3 for the illustration. Let $p_{ij} \coloneqq \Pr[f_i(\boldsymbol{u}) \text{ queries } j]$. Then

$$\mathbb{E}\left[\sum_{a \neq b \in \boldsymbol{I}} \sum_{(i,j) \in J_a \times J_b} p_{ij}\right] = \sum_{a \neq b \in [\ell]} \Pr[a \in \boldsymbol{I}] \Pr[b \in \boldsymbol{I}] \sum_{(i,j) \in J_a \times J_b} p_{ij} \leq \alpha^2 md.$$

We now argue that there exists $I \subseteq [\ell]$ and corresponding $J \coloneqq \bigcup_{i \in I} J_i$ satisfying three conditions:

(R1) $\mathrm{H}(f_J(\boldsymbol{u})) \geq (1-1/d) \cdot \alpha\, \mathrm{H}(f(\boldsymbol{u}))$.

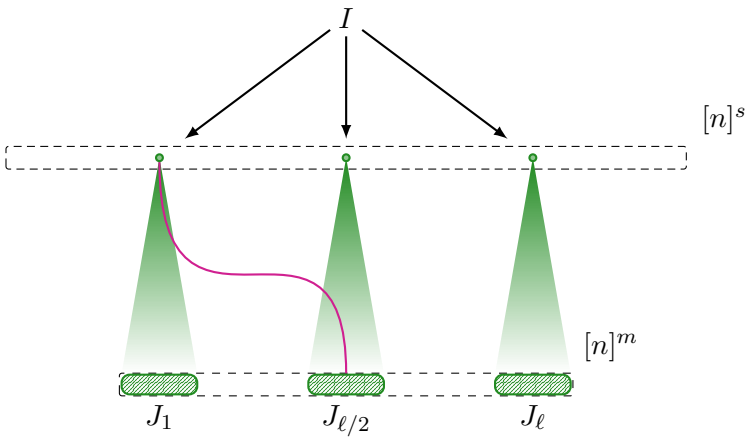

Figure 3: The picture illustrates the choice of $\boldsymbol{J}$. We first sample the set $\boldsymbol{I} \subseteq [\ell]$ of input cells and take to $\boldsymbol{J}$ only the trees that query a symbol in $\boldsymbol{I}$ as their first query. The undesirable events for us are **non-first** queries from $\boldsymbol{J}$ to $\boldsymbol{I}$, it is represented as the red line in the picture. The main point is that for a fixed $i \in [\ell]$ and $j \in [m]$ this happens with probability α^2, but the expected size of $\boldsymbol{J}$ is an α-fraction of $[m]$. Hence, the subsampling procedure sparsifies the undesirable events.

(R2) $|J| \le (1 + 1/d) \cdot \alpha m$.
(R3) $\mathbb{E}[$ number of non-first queries to I that f_J make $] \le \alpha^{-1/2} d \cdot \alpha^2 m d$.

Claim 19. *Conditions (R1)-(R3) are satisfied by $\boldsymbol{I}$ and $\boldsymbol{J}$ with positive probability.*

Proof. By Markov's inequality $\boldsymbol{I}$ satisfies the condition (R3) with probability $1 - \sqrt{\alpha}/d$.

Let us compute the probability that (R2) is not satisfied by $\boldsymbol{J}$. For an event A, let $[\![A]\!]$ denote the random variable that is 1 if A occurs and 0 otherwise. Then

$$\begin{aligned}
\Pr[|\boldsymbol{J}| > (1+1/d)\alpha m] &= \Pr\left[\sum_{i\in[\ell]} |J_i| [\![i \in \boldsymbol{I}]\!] > (1+1/d)\alpha m\right] \\
&= \Pr\left[\sum_{i\in[\ell]} |J_i| [\![i \in \boldsymbol{I}]\!] - \mathbb{E}[|\boldsymbol{J}|] > 1/d \cdot \alpha m\right] \\
\text{(Hoeffding's inequality)} \quad &\le \exp(-\Omega((\alpha/d)^2 \cdot m^2 / \sum_{i\in[\ell]} |J_i|^2)) \\
&= \exp(-\Omega(\alpha^2 m/(\mu d)^2)) \\
\text{(since } m \ge n^{0.99}\text{, and } \mu \le n^{0.3}\text{)} \quad &= \exp(-n^{\Omega(1)}).
\end{aligned}$$

Now we turn to (R1). The key step is to show that the entropy is retained in expectation: $\mathbb{E}_{\boldsymbol{J}}[\mathrm{H}(f_{\boldsymbol{J}}(\boldsymbol{u}))] \ge \alpha\, \mathrm{H}(f(\boldsymbol{u}))$. This holds by Shearer's inequality:

Lemma 20 (Shearer's Inequality [CGFS86])**.** *Suppose $\boldsymbol{S} \subseteq [n]$ is a distribution over subsets of $[n]$ such that for every $i \in [n]$ $\Pr[i \in \boldsymbol{S}] \ge \kappa$. Then for any random variable $\boldsymbol{x} \sim \Sigma^n$ we have*

$$\mathrm{H}(\boldsymbol{x}) \le \frac{1}{\kappa}\, \mathbb{E}[H(\boldsymbol{x}_{\boldsymbol{S}})].$$

On the other hand

$$\begin{aligned}\mathbb{E}_{\boldsymbol{J}}[\mathrm{H}(f_{\boldsymbol{J}}(\boldsymbol{u}))] \leq \quad & \Pr[\mathrm{H}(f_{\boldsymbol{J}}(\boldsymbol{u})) < (1-1/d)\cdot\alpha\,\mathrm{H}(f(\boldsymbol{u}))]\cdot(1-1/d)\cdot\alpha\,\mathrm{H}(f(\boldsymbol{u})) \\ & + \Pr[\mathrm{H}(f_{\boldsymbol{J}}(\boldsymbol{u})) \geq (1-1/d)\cdot\alpha\,\mathrm{H}(f(\boldsymbol{u})) \wedge |\boldsymbol{J}| \leq 2\alpha m]\cdot 2\alpha m\log n \\ & + \Pr[|\boldsymbol{J}| > 2\alpha m]\cdot\mathrm{H}(f(\boldsymbol{u})) \\ \leq\ & (1-1/d)\cdot\alpha\,\mathrm{H}(f(\boldsymbol{u})) + p\cdot 2\alpha m\log n + \exp(-n^{\Omega(1)}),\end{aligned}$$

where $p \coloneqq \Pr[\mathrm{H}(f_{\boldsymbol{J}}(\boldsymbol{u})) \geq (1-1/d)\cdot\alpha\,\mathrm{H}(f(\boldsymbol{u}))]$. By rearranging we get

$$\frac{\mathrm{H}(f(\boldsymbol{u}))/d - \exp(-n^{\Omega(1)})}{2m\log n} \leq p.$$

Since $\mathrm{H}(f(\boldsymbol{u})) = cm\log n$ we then get $p \geq c/3d = \omega(1/(d\log n))$, so since $\omega(1/(d\log n)) - \sqrt{\alpha}/d - \exp(-n^{\Omega(1)}) = \omega(1/(d\log n)) > 0$ there exists I satisfying (R1), (R2), (R3). □

Second step: pruning the trees. Having $I \subseteq [\ell]$ and $J \subseteq [m]$ satisfying the conditions (R1)-(R3) we now define trees $g_j \colon [n]^s \to [n] \cup \{\perp\}$ for each $j \in J$ as follows: g_j follows the behavior of f_j until it is about to query an input cell from I in which case it returns $\perp$.

Claim 21. $\mathrm{H}(g_J(\boldsymbol{u})) \geq \mathrm{H}(f_J(\boldsymbol{u})) - O(\alpha^{3/2}md^2\log n) \geq (1-1/d)\,\mathrm{H}(f_J(\boldsymbol{u}))$.

Proof. The second inequality is satisfied since

$$\alpha^{3/2}md^2\log n = \alpha md^2/\log^2 n \leq \alpha m/\log n = o(\alpha cm\log n/d) = o(\mathrm{H}(f_J(\boldsymbol{u}))/d).$$

We now prove the first inequality. Consider a random variable $\boldsymbol{b} \in ([n]\cup\{\perp\})^J$ such that $\boldsymbol{b}_j = \perp$ if $g_j(\boldsymbol{u}) \neq \perp$ and $\boldsymbol{b}_j = f_j(\boldsymbol{u})$ if $g_j(\boldsymbol{u}) = \perp$. Then $f_J(\boldsymbol{u})$ is uniquely determined by $\boldsymbol{b}$ and $g_J(\boldsymbol{u})$, so $\mathrm{H}(f_J(\boldsymbol{u})) \leq \mathrm{H}(g_J(\boldsymbol{u})) + \mathrm{H}(\boldsymbol{b})$. Then, since the expected number of non-$\perp$ symbols in $\boldsymbol{b}$ is bounded by (R3), we conclude since by Fact 2 $\mathrm{H}(\boldsymbol{b}) \leq \log n(2 + 2\alpha^{3/2}md^2)$. □

Third step: restricting the inputs. By Fact 1 we either have $\mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_I) \geq \mathrm{H}(g_J(\boldsymbol{u}))\cdot(1-1/d)$ or $\mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_{[s]\setminus I}) \geq \mathrm{H}(g_J(\boldsymbol{u}))\cdot(1/d)$. Suppose that the former holds, then, since g is average-μ-Lipschitz and $\mu \leq n^{0.3}$, Lemma 11 implies

$$\begin{aligned}&\Pr_{\boldsymbol{r}\sim\boldsymbol{u}}[\mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{r}_I) \geq (1-2/d)\,\mathrm{H}(g_J(\boldsymbol{u}))] \geq \\ &\Pr_{\boldsymbol{r}\sim\boldsymbol{u}}[|\,\mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{r}_I) - \mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_I)| \leq (1/d)\,\mathrm{H}(g_J(\boldsymbol{u}))] \geq \\ &\Pr_{\boldsymbol{r}\sim\boldsymbol{u}}[|\,\mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{r}_I) - \mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_I)| \leq n^{0.9}] \geq 1 - \exp(-n^{\Omega(1)}).\end{aligned}$$

The second inequality holds since $\mathrm{H}(g_J(\boldsymbol{u}))/d \geq m/\mathrm{poly}(\log n) \gg n^{0.9}$. If $\mathrm{H}(g_J(\boldsymbol{u}) \mid \boldsymbol{u}_{[s]\setminus I}) \geq \mathrm{H}(g_J(\boldsymbol{u}))\cdot(1/d)$ holds we apply Lemma 11 analogously. Then either (C1) or (C2) is satisfied since $\mathrm{H}(g_J(\boldsymbol{u})) \geq (1-1/d)\,\mathrm{H}(f_J(\boldsymbol{u}))$ by Claim 21.

3.3 Entropy after assignment

In this section, we prove Fact 1, the proof is a simple chain rule computation.

Fact 1. *Suppose $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell$ are independent and $\boldsymbol{y} = f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell)$ then* $\mathrm{H}(\boldsymbol{y}) \leq \sum_{i\in[\ell]} \mathrm{H}(\boldsymbol{y} \mid \boldsymbol{x}_{[\ell]\setminus i})$.

We first prove it in the special case of $\ell = 2$:

Fact 3. *Suppose $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{z}$ are independent and $\boldsymbol{y} = f(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{z})$ then*

$$\mathrm{H}(\boldsymbol{y} \mid \boldsymbol{z}) \leq \mathrm{H}(\boldsymbol{y} \mid \boldsymbol{x}_1, \boldsymbol{z}) + \mathrm{H}(\boldsymbol{y} \mid \boldsymbol{x}_2, \boldsymbol{z}).$$

Proof. We write

$$\begin{aligned}
\mathrm{H}(\boldsymbol{x}_1 \mid \boldsymbol{z}) + \mathrm{H}(\boldsymbol{x}_2 \mid \boldsymbol{z}) &= \mathrm{H}(\boldsymbol{x}_1, \boldsymbol{x}_2 \mid \boldsymbol{z}) \\
&= \mathrm{H}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y} \mid \boldsymbol{z}) \\
\text{(chain rule for Shannon entropy)} \quad &= \mathrm{H}(\boldsymbol{y} \mid \boldsymbol{z}) + \mathrm{H}(\boldsymbol{x}_1 \mid \boldsymbol{y}, \boldsymbol{z}) + \mathrm{H}(\boldsymbol{x}_2 \mid \boldsymbol{y}, \boldsymbol{x}_1, \boldsymbol{z}) \\
\text{(entropy decreases with conditioning)} \quad &\leq \mathrm{H}(\boldsymbol{y} \mid \boldsymbol{z}) + \mathrm{H}(\boldsymbol{x}_1 \mid \boldsymbol{y}, \boldsymbol{z}) + \mathrm{H}(\boldsymbol{x}_2 \mid \boldsymbol{y}, \boldsymbol{z}) \\
&= \mathrm{H}(\boldsymbol{y}, \boldsymbol{x}_1 \mid \boldsymbol{z}) + \mathrm{H}(\boldsymbol{y}, \boldsymbol{x}_2 \mid \boldsymbol{z}) - \mathrm{H}(\boldsymbol{y} \mid \boldsymbol{z})
\end{aligned}$$

By applying $\mathrm{H}(\boldsymbol{y}, \boldsymbol{x}_i \mid \boldsymbol{z}) = \mathrm{H}(\boldsymbol{y} \mid \boldsymbol{x}_i, \boldsymbol{z}) + \mathrm{H}(\boldsymbol{x}_i \mid \boldsymbol{z})$ for $i \in \{1, 2\}$ and rearranging we get the claim. □

Now we can derive Fact 1.

Proof of Fact 1. Applying Fact 3 with empty $\boldsymbol{z}$ we get:

$$\mathrm{H}(f(\boldsymbol{x})) \leq \mathrm{H}(f(\boldsymbol{x}) \mid \boldsymbol{x}_1) + \mathrm{H}(f(\boldsymbol{x}) \mid \boldsymbol{x}_{[d] \setminus 1}).$$

Then we continue to rewrite $\mathrm{H}(f(\boldsymbol{x}) \mid \boldsymbol{x}_1) \leq \mathrm{H}(f(\boldsymbol{x}) \mid \boldsymbol{x}_1, \boldsymbol{x}_2) + \mathrm{H}(f(\boldsymbol{x}) \mid \boldsymbol{x}_1, \boldsymbol{x}_{[n] \setminus \{1,2\}})$ again using Fact 3 with $\boldsymbol{z} = \boldsymbol{x}_1$. We then get the claim by a simple induction. □

4 Lipschitz Decision Forests

Two of our technical lemmas are in fact some form of concentration, Lemma 11 is precisely the concentration of conditional entropy, and Lemma 12 can be seen as the concentration of *distance from a random point to a set*. In both cases we reduce the question to the McDiarmid's inequality.

McDiarmid's inequality. A function $f \colon \Lambda^s \to \mathbb{R}$ has *c-bounded differences over* $S \subseteq \Lambda^s$ if for every $x, x' \in S$ that differ in one coordinate we have $|f(x) - f(x')| \leq c$.

The following concentration inequality is well-known and can be found, for example, in [Com15].

Lemma 22 (McDiarmid's inequality)**.** *Suppose $f \colon \Lambda^s \to \mathbb{R}$ has c-bounded differences over S and for $\boldsymbol{u} \sim [n]^s$ we have $\Pr[\boldsymbol{u} \in S] = 1 - \delta$. Then*

$$\Pr[|f(\boldsymbol{u}) - \mathbb{E}[f(\boldsymbol{u}) \mid \boldsymbol{u} \in S]| \geq \lambda + \delta cs] \leq 2(\delta + \exp(-\Omega(\lambda^2/(c^2 s))).$$

The bounded difference property is very similar in spirit to Lipschitzness, the topic of this section. We restate the definition for convenience:

Definition 1 ([BIL12]). Let $f \colon [n]^s \to [n]^n$ be a decision forest. Let $\boldsymbol{\theta}_j$ denote the number of trees in f that query j on the input $\boldsymbol{u} \sim [n]^s$. We then say that f is *average-μ-Lipschitz* if $\mathbb{E}[\boldsymbol{\theta}_j] \leq \mu$ for every $j \in [s]$. We say that f is (μ, δ)-Lipschitz if $\Pr[\boldsymbol{\theta}_j > \mu] \leq \delta$ for every $j \in [s]$.

This concept was studied (with somewhat different notation) by Beck, Impagliazzo, and Lovett [BIL12], who proved a concentration inequality for Lipschitz forests.

Theorem 23 ([BIL12, Theorem 1.7]). *Suppose that $f\colon \{0,1\}^s \to \{0,1\}^m$ is an average-μ-Lipschitz depth-d decision forest. Then*

$$\Pr_{\boldsymbol{u}\sim\{0,1\}^s}\left[\left|\sum_{i\in[n]} f_i(\boldsymbol{u}) - \mathbb{E}\left[\sum_{i\in[n]} f_i(\boldsymbol{u})\right]\right| > d\cdot\sqrt{\mu\cdot n\cdot\log(d^4/\varepsilon)}\right] \le \varepsilon.$$

Unfortunately, we cannot use their results in a black-box way, the first reason is that they are stated only for the binary alphabet, whereas we need it for *exponential-size* alphabet, and the second is because of the $\sqrt{n}$ multiplier in the deviation bound. We would like to apply such concentration bound in the case of low expectation, so this multiplier would be too weak. For these reasons we prove the following simpler deviation bound:

Lemma 24. *Suppose that $f\colon \Lambda^s \to [0,1]^m$ is a depth-d average-μ-Lipschitz decision forest. Then if $\kappa \coloneqq \mathbb{E}\left[\sum_{i\in[m]} f_i(\boldsymbol{u})\right]$, we have for every $\varepsilon > 0$*

$$\Pr\left[\sum_{i\in[m]} f_i(\boldsymbol{u}) \ge 2(\kappa + \log(1/\varepsilon)d\mu)\right] \le \varepsilon.$$

As a corollary of Lemma 24 we get, following the simplified proof of [BIL12, Corollary 1.8] that

Lemma 6. *If $f\colon \Lambda^s \to \Sigma^m$ is an average-μ-Lipschitz depth-d decision forest, then for every $\varepsilon > 0$ the forest f is $(3\mu d^2\log(1/\varepsilon), \varepsilon)$-Lipschitz.*

Proof. As above, for every $j \in [s]$ let $\boldsymbol{\theta}_j$ be number of trees in f querying j on the input $\boldsymbol{u}$. Let $g_i^{\ell,j}$ be the depth-ℓ decision tree that returns 1 if the ℓ-th query of f_i is to j and 0 otherwise. Clearly $g^{\ell,j}$ is obtained by pruning f up to the ℓ-th layer and replacing all j-labels with 1 and all others with 0.

Then $\boldsymbol{\theta}_j = \sum_{\ell\in[d]}\sum_{i\in[n]} g_i^{\ell,j}(\boldsymbol{u})$. The forest $g_i^{\ell,j}$ for fixed j and $\ell \in [d]$, $i \in [n]$ is average-$d\mu$-Lipschitz, since every tree in the forest is obtained by pruning a tree in f and every tree from f corresponds to d trees in the forest. Thus, by Lemma 24 we have

$$\Pr[\boldsymbol{\theta}_j \ge 2(\mu + \log(1/\varepsilon)d^2\mu)] \le \varepsilon,$$

which implies the claim. □

4.1 Proof of Lemma 24

We denote $\tau_i(\boldsymbol{u}) \in (\Lambda\times[s])^d$ the transcript of running f_i on $\boldsymbol{u}$ (the queries and the outcomes). We abuse notation to denote by $f_i(\alpha)$ the value of f_i given the transcript $\alpha \in (\Lambda\times[s])^d$.

We now estimate the k-th moment of $\sum_{i\in[m]} f_i(\boldsymbol{u})$ by induction on k. We are going to prove the statement by induction on k. Let $F(\kappa,k)$ be the maximum of $\mathbb{E}[(\sum_{i\in[m]} h_i(\boldsymbol{u}))^k]$ over all depth-d average-μ-Lipschitz decision forests h with $\mathbb{E}[\sum_{i\in[m]} h_i(\boldsymbol{u})] = \kappa$. The base of induction is $k = 1$ where trivially $F(\kappa,1) = \kappa$.

$$\begin{aligned}\mathbb{E}\left[\Big(\sum_{i\in[m]} f_i(\boldsymbol{u})\Big)^k\right] &= \sum_{i,j_1,\dots,j_{k-1}\in[m]} \mathbb{E}[f_i(\boldsymbol{u})f_{j_1}(\boldsymbol{u})\cdot\dots\cdot f_{j_{k-1}}(\boldsymbol{u})]\\ &= \sum_{i\in[m]}\sum_{\alpha\in(\Lambda\times[s])^d} \Pr[\tau_i(\boldsymbol{u}) = \alpha] f_i(\alpha)\cdot\mathbb{E}\left[\Big(\sum_{j\in[m]} f_j(\boldsymbol{u})\Big)^{k-1}\,\middle|\,\tau_i(\boldsymbol{u}) = \alpha\right]\end{aligned}$$

Let us now estimate $\mathbb{E}\left[(\sum_{j\in[m]} f_j(\boldsymbol{u}))^{k-1} \mid \tau_i(\boldsymbol{u}) = \alpha\right]$ for fixed i, α. Consider a decision forest g^α such that g_i^α is a copy of f_i where all queries to α are replaced with leaves labeled with 1, in other words, when we are about to query something in α, we return 1 instead. Such a transformation will preserve the μ-Lipschitz property, as for every j, the number of queries can only decrease. We then further estimate the expectation

$$\begin{aligned}
\mathbb{E}\left[\Big(\sum_{j\in[m]} f_j(\boldsymbol{u})\Big)^{k-1} \,\middle|\, \tau_i(\boldsymbol{u}) = \alpha\right] &\le \mathbb{E}\left[\Big(\sum_{j\in[m]} g_j^\alpha(\boldsymbol{u})\Big)^{k-1} \,\middle|\, \tau_i(\boldsymbol{u}) = \alpha\right] \\
(\text{as } g_j^\alpha \text{ never query } \alpha) \ &= \mathbb{E}\left[\Big(\sum_{j\in[m]} g_j^\alpha(\boldsymbol{u})\Big)^{k-1}\right] \\
&\le F\left(\mathbb{E}\left[\sum_{j\in[m]} g_j^\alpha(\boldsymbol{u})\right], k-1\right)
\end{aligned}$$

Now let us compute the new expectation:

$$\begin{aligned}
\mathbb{E}\left[\sum_{j\in[m]} g_j^\alpha(\boldsymbol{u})\right] &= \mathbb{E}\left[\sum_{j\in[m]} (g_j^\alpha(\boldsymbol{u}) - f_j(\boldsymbol{u}))\right] + \kappa \\
&\le d\mu + \kappa.
\end{aligned}$$

Here we use that $(g_j^\alpha(\boldsymbol{u}) - f_j(\boldsymbol{u})) \neq 0$ only if f_j queries α, hence the expectation of the sum of these over $j \in [n]$ is bounded by the expected number of queries to α, which is at most $d\mu$. Putting all together we get

$$\begin{aligned}
\mathbb{E}\left[\Big(\sum_{i\in[m]} f_i(\boldsymbol{u})\Big)^{k}\right] &\le F(d\mu+\kappa, k-1) \cdot \sum_{i\in[m];\ \alpha\in(\Lambda\times[s])^d} f_i(\alpha)\Pr[\tau_i(\boldsymbol{u}) = \alpha] \\
&\le F(d\mu+\kappa, k-1)\cdot\kappa \\
&\le (\kappa + kd\mu)^k.
\end{aligned}$$

Finally, by Markov inequality

$$\Pr\left[\sum_{i\in[m]} f_i(\boldsymbol{u}) \ge a\right] = \Pr\left[\Big(\sum_{i\in[m]} f_i(\boldsymbol{u})\Big)^k \ge a^k\right] \le \frac{\mathbb{E}\left[\Big(\sum_{i\in[m]} f_i(\boldsymbol{u})\Big)^k\right]}{a^k} \le \left(\frac{\kappa + kd\mu}{a}\right)^k$$

and we get the desired inequality by substituting $a = 2(\kappa + d\mu\cdot\log(1/\varepsilon))$ and $k = \log(1/\varepsilon)$.

4.2 Conditional entropy concentration

The main property of Lipschitz forests is that conditional entropy concentrates in the following sense:

Lemma 25. *Suppose $f\colon [n]^s \to \Sigma^m$ is an (μ,δ)-Lipschitz depth-d decision forest with $\mu = m^{\Theta(1)}$ and $\delta = \exp(-m^{\Omega(1)})$. Let $I \subseteq [s]$ be a subset of the input cells. Then for $\boldsymbol{r}$ and $\boldsymbol{u}$ uniformly distributed over $[n]^s$ we have*

$$\Pr[|\operatorname{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{r}_I) - \operatorname{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I)| > \lambda\log(m|\Sigma|)\sqrt{\mu m d}] \le \exp(-\Omega(\lambda^2)) + \exp(-m^{\Omega(1)}).$$

Before proceeding to the proof of this lemma, let us derive a simpler form that we use in Section 3:

Lemma 11. *Let $f\colon [n]^s \to \Sigma^m$ be an average-$n^{0.3}$-Lipschitz depth-$\log^{O(1)} n$ decision forest. Suppose that $m \le n$ and $|\Sigma| = O(n)$. Let $I \subseteq [s]$ be a subset of the input cells. Then for $\boldsymbol{r}$ and $\boldsymbol{u}$ uniformly distributed over $[n]^s$ we have*

$$\Pr\left[\left|\operatorname{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I = \boldsymbol{r}_I) - \operatorname{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I)\right| > n^{0.9}\right] \le \exp(-n^{\Omega(1)}).$$

Proof. Wlog we may assume $m = n$ by adding $n - m$ trivial decision trees to f. Then, taking $\lambda = n^{0.1}$, we get $\lambda \log(m|\Sigma|)\sqrt{\mu m d} = \operatorname{poly}(\log n) \cdot n^{0.75} \ll n^{0.9}$. □

The key step in the proof of Lemma 25 is to show that the conditional entropy has bounded differences:

Lemma 26. *Suppose $f\colon \Lambda^s \to \Sigma^m$ is an average-μ-Lipschitz decision forest. Then for $\boldsymbol{y} \sim \Lambda$, $\boldsymbol{z} \sim \Lambda^{s-1}$, and every $y \in \Lambda$ we have with $\boldsymbol{o} \coloneqq f(\boldsymbol{y}, \boldsymbol{z})$ that*

$$\left|\operatorname{H}(\boldsymbol{o} \mid \boldsymbol{y} = y) - \operatorname{H}(\boldsymbol{o})\right| \le \log(m+1) + \mu \log(m|\Sigma|).$$

Proof. Fix some $y \in \Lambda$ and let $\boldsymbol{o}' \coloneqq f(y, \boldsymbol{z})$. Let $\boldsymbol{a} \in (\Sigma \cup \{\perp\})^m$ be the random variable such that for each $i \in [m]$ we have $\boldsymbol{a}_i = \boldsymbol{o}'_i$ if the first input cell was queried by f_i and $\boldsymbol{a}_i = \perp$ if it was not. Then $\operatorname{H}(\boldsymbol{o}') \le \operatorname{H}(\boldsymbol{o}, \boldsymbol{a}) \le \operatorname{H}(\boldsymbol{o}) + \operatorname{H}(\boldsymbol{a})$, since $\boldsymbol{o}'$ is uniquely determined by $\boldsymbol{o}$ and $\boldsymbol{a}$. We then observe that the expected number of non-$\perp$ elements in $\boldsymbol{a}$ is at most μ in expectation: indeed f is average-μ-Lipschitz and fixing the first input cell does not change the expected number of queries to it, since the trees in f query each symbol at most once. Then by Fact 2 we get $\operatorname{H}(\boldsymbol{a}) \le \log(m+1) + \mu \log(m|\Sigma|)$ Hence $\operatorname{H}(\boldsymbol{o} \mid \boldsymbol{y} = y) \le \operatorname{H}(\boldsymbol{o}) + \log(m+1) + \mu \log(m|\Sigma|)$. The reverse direction is proved analogously. □

Corollary 27. *Suppose $f\colon \Lambda^s \to \Sigma^m$ is a (μ, δ)-Lipschitz decision forest, with $\delta = \exp(-m^{\Omega(1)})$, and $\mu = m^{\Theta(1)}$. Let $I \subseteq [s]$ be a subset of input cells, and let a function $h\colon \Lambda^I \to \mathbb{R}$ map $y \mapsto \operatorname{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I = y)$. Then there exists a set $S \subseteq \Lambda^I$ such that $\Pr_{\boldsymbol{r} \sim \Lambda^I}[\boldsymbol{r} \in S] \ge 1 - |I| \cdot \sqrt{\delta}$ and h has $O(\mu \log(m|\Sigma|))$-bounded differences over S.*

Proof. Let us fix some $i \in I$ and show that there exists a set S_i with $\Pr[\boldsymbol{r} \in S_i] \ge 1 - \sqrt{\delta}$ such that h has bounded differences in the i-th coordinate in the set S_i. The claim then follows by the union bound.

Let S_i be set of $r \in \Lambda^I$ such that $f_{r_{I \setminus \{i\}}}$ is $(\mu, \sqrt{\delta})$-Lipschitz, by Lemma 17 the forest $f|_{\boldsymbol{r}_{I \setminus \{i\}}}$ is $(\mu, \sqrt{\delta})$-Lipschitz with probability $1 - \sqrt{\delta}$ over the choice of $\boldsymbol{r} \sim \Lambda^I$. Then $g_r \coloneqq f|_{r_{I \setminus \{i\}}}$ is also average-$(\mu + \sqrt{\delta} m)$-Lipschitz. Since $\sqrt{\delta} m = O(\mu)$, an application of Lemma 26 implies that for every $y \in \Lambda$

$$\left|\operatorname{H}(g_r(\boldsymbol{y}, \boldsymbol{z}) \mid \boldsymbol{y} = y) - \operatorname{H}(g_r(\boldsymbol{y}, \boldsymbol{z}))\right| \le O(\mu \log(m|\Sigma|)),$$

where $\boldsymbol{y} \sim \Lambda^{\{i\}}$ and $\boldsymbol{z} \sim \Lambda^{[s] \setminus I}$. Then notice that $h(x, y) = \operatorname{H}(g_y(\boldsymbol{y}, \boldsymbol{z}) \mid \boldsymbol{y} = y)$, so the above shows that h has $O(\mu \log(m|\Sigma|))$-bounded differences in S_i for the i-th coordinate, as required. □

Proof of Lemma 25. First we reduce the number of the input cells in I: we cluster it and increase the alphabet, so each cluster becomes a single symbol. The challenge is to do it so that average-μ-Lipschitzness does not suffer. Let $\boldsymbol{\theta}_i$ for $i \in [s]$ be the number of queries to the input cell i the trees in f collectively make on the input $\boldsymbol{u} \sim [n]^s$. Observe that $\mathbb{E}[\sum_{i \in [s]} \boldsymbol{\theta}_i] \le md$, since each tree makes at most d queries on the given input. Let $I_1, \dots, I_\ell$ be the partition of I such that $\sum_{i \in I_j} \mathbb{E}[\boldsymbol{\theta}_i] \le \mu$

for every $j \in [\ell]$ and ℓ is minimized. Then no two sets in $I_1, \dots, I_\ell$ can be united so the property is satisfied, hence $\sum_{i \in I_j} \mathbb{E}[\boldsymbol{\theta}_i] \geq \mu/2$ for all $j \in [\ell]$ except perhaps one, thus $\ell \leq 2md/\mu + 1 \leq 3md/\mu$. Then we set $\Lambda \coloneqq [n]^{\max_{j \in [\ell]} |I_j|}$ and for every tree in f replace each query to the input cell $i \in I$ with the query to the cluster $I_j \ni i$. This does not affect the depth of the trees and each new input cell is queried at most μ times in expectation. Thus, the obtained forest $f' \colon \Lambda^\ell \times [n]^{[s] \setminus I} \to [n]^m$ is average-μ-Lipschitz depth-d decision forest such that $f(\boldsymbol{u}) \equiv f'(\boldsymbol{y}, \boldsymbol{z})$ for $\boldsymbol{y} \sim \Lambda^\ell$, $\boldsymbol{z} \sim [n]^{[s] \setminus I}$.

Let $h \colon \Lambda^\ell \to \mathbb{R}$ map $y \in \Lambda^\ell$ to $\mathrm{H}(f'(y, \boldsymbol{z})) = \mathrm{H}(f(y', \boldsymbol{u}_{[s] \setminus I}))$, where $y' \in [n]^I$ is the unclustered version of y. Then $\mathbb{E}[h(\boldsymbol{r})] = \mathrm{H}(f(\boldsymbol{u}) \mid \boldsymbol{u}_I)$.

By Corollary 27 we have that h has $O(\mu \log(m|\Sigma|)$-bounded differences on a set $S \subseteq \Lambda^\ell$ with probability mass $1 - \ell\sqrt{\delta}$. So by Lemma 22 we have

$$\begin{aligned}
\Pr_{\boldsymbol{r} \sim [n]^I}[|h(\boldsymbol{r}) - \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \boldsymbol{r}}[h(\boldsymbol{y})]| \geq \lambda + o(1)] &= \Pr_{\boldsymbol{r} \sim [n]^I}[|h(\boldsymbol{r}) - \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \boldsymbol{r}}[h(\boldsymbol{y})]| \geq \lambda + O(\delta \ell \mu \log(m|\Sigma|)) \cdot \ell\sqrt{\delta}] \\
\text{(by Lemma 22)} \quad &\leq 2 \exp\left(-\Omega\left(\frac{\lambda^2}{c^2 \ell}\right)\right) + 2\ell\sqrt{\delta} \\
&\leq \exp\left(-\Omega\left(\frac{\lambda^2}{\mu \log^2(m|\Sigma|) md}\right)\right) + \exp(-m^{\Omega(1)}).
\end{aligned}$$

Replacing λ with $\lambda' \coloneqq \lambda + o(1)$ we get the claimed inequality. □

4.3 Enforcing average Lipschitzness: proof of Lemma 13

In this section we show that exhaustively fixing input cells that violate average Lipschitzness eventually to random values eventually makes any bounded depth decision forest average-Lipschitz. In [BIL12, Claim 3.10] it is shown for average-$\sqrt{n}$-Lipschitzness and above, we show it for the values below $\sqrt{n}$. We need the following classical fact:

Lemma 28 (Expected stopping time, see e.g. [KK18])**.** *Suppose $\{\boldsymbol{x}_i\}_{i \in \mathbb{Z}_{>0}}$ is a sequence of random variables. Define the stopping time to be $\boldsymbol{t} \coloneqq \min\{t \in \mathbb{Z}_{>0} \mid \boldsymbol{x}_i \geq N\}$, and assume it is finite. Then whenever $\mathbb{E}[\boldsymbol{x}_i - \boldsymbol{x}_{i-1} \mid \boldsymbol{x}_{<i}, i \leq \boldsymbol{t}] \geq \varepsilon$, we have $\mathbb{E}[\boldsymbol{t}] \leq N/\varepsilon$.*

We first show a weaker version of the lemma, we will then see how to easily boost it to get exponential success probability.

Lemma 29. *Suppose $f \colon [n]^s \to [n]^m$ is an arbitrary depth-d decision forest. There exists a $nd/(\varepsilon\mu)$-depth decision tree T querying symbols of $[n]^s$ such that for a random leaf $\boldsymbol{\ell}$ of T, $f|_{\boldsymbol{\ell}}$ is average-μ-Lipschitz with probability $1 - \varepsilon$.*

Proof. We define the tree by describing a random walk from its root to a leaf. Let $\boldsymbol{a}_1, \boldsymbol{a}_2, \dots \in [s]$ be the random variables describing what input cells are fixed at each step of the walk and $\boldsymbol{b}_1, \boldsymbol{b}_2, \dots \in [n]$ describe the values they are fixed to. All $\boldsymbol{b}_i$ are independent and uniform over $[n]$. Let $f_{\boldsymbol{a}_{<i}, \boldsymbol{b}_{<i}}$ denote the forest with the input cell $\boldsymbol{a}_j$ is fixed to $\boldsymbol{b}_j$ for all $j < i$. Then $\boldsymbol{a}_i$ is defined as an element of $[n] \setminus \{\boldsymbol{a}_1, \dots, \boldsymbol{a}_{i-1}\}$ such that the expectation of the number of queries to it by $f_{\boldsymbol{a}_{<i}, \boldsymbol{b}_{<i}}$ is the largest (breaking ties arbitrarily). If that value is less than μ the process stops, so in that case $\boldsymbol{t} = i - 1$.

Let $\boldsymbol{p}_i$ be the expected number of queries made by $f_{\boldsymbol{a}_{\leq i}, \boldsymbol{b}_{\leq i}}$ on a uniform random input $\boldsymbol{u} \sim [n]^{[s] \setminus \{\boldsymbol{a}_1, \dots, \boldsymbol{a}_i\}}$. We claim that $\mathbb{E}[\boldsymbol{p}_{i-1} - \boldsymbol{p}_i \mid \boldsymbol{a}_{<i}, \boldsymbol{b}_{<i}] \geq \mu$. Let $\boldsymbol{q}_{i,S}$ be the expected number of

queries to $S \subseteq [s]$ made by $f_{\boldsymbol{a}_{\le i}, \boldsymbol{b}_{\le i}}$ on a uniform random input, so $\boldsymbol{p}_i = \boldsymbol{q}_{i,[s]}$. Then

$$\begin{aligned}
\mathbb{E}[\boldsymbol{p}_{i-1} - \boldsymbol{p}_i \mid \boldsymbol{a}_{<i}, \boldsymbol{b}_{<i}] &= \boldsymbol{p}_{i-1} - \mathbb{E}[\boldsymbol{q}_{i,[s]\setminus \boldsymbol{a}_i} + \boldsymbol{q}_{i,\boldsymbol{a}_i} \mid \boldsymbol{a}_{<i}, \boldsymbol{b}_{<i}] \\
(\boldsymbol{a}_i \text{ is already assigned}) \quad &= \boldsymbol{p}_{i-1} - \mathbb{E}[\boldsymbol{q}_{i,[s]\setminus \boldsymbol{a}_i} \mid \boldsymbol{a}_{<i}, \boldsymbol{b}_{<i}] \\
&= \mathbb{E}[\boldsymbol{q}_{i-1,[s]\setminus \boldsymbol{a}_i} - \boldsymbol{q}_{i,[s]\setminus \boldsymbol{a}_i} + \boldsymbol{q}_{i-1,\boldsymbol{a}_i} \mid \boldsymbol{a}_{<i}, \boldsymbol{b}_{<i}] \\
(\text{expected number of queries decreases}) \quad &\ge \mathbb{E}[\boldsymbol{q}_{i-1,\boldsymbol{a}_i} \mid \boldsymbol{a}_{<i}, \boldsymbol{b}_{<i}] \\
&\ge \mu.
\end{aligned}$$

Applying Lemma 28 to $\boldsymbol{p}_0 - \boldsymbol{p}_i$ we get that $\mathbb{E}[\boldsymbol{t}] \le \boldsymbol{p}_0/\mu \le nd/\mu$. Hence, the expected depth of a leaf of the tree is at most nd/μ. Let us then prune all branches of the tree at depth more than $nd/(\varepsilon\mu)$ and get observe that we cut just ε-fraction of the leaves by Markov's inequality on $\boldsymbol{t}$. □

We are now ready to prove the stronger version.

Lemma 13. *Suppose $f\colon [n]^s \to [n]^m$ is an arbitrary depth-d decision forest. There exists a $2nd/\mu \cdot \log(1/\varepsilon)$-depth decision tree T querying symbols of $[n]^s$ such that for a random leaf $\boldsymbol{\ell}$ of T, $f|_{\boldsymbol{\ell}}$ is average-μ-Lipschitz with probability $1-\varepsilon$.*

Proof. Applying Lemma 29 with $\varepsilon = 1/2$ we get that there exists a tree of depth $2nd/\mu$-depth decision tree T_0 such that for its random leaf $\boldsymbol{\ell}$ we have that $f|_{\boldsymbol{\ell}}$ is average-μ-Lipschitz with probability $1/2$. We say that a leaf ℓ is *successful* if $f|_\ell$ is average-μ-Lipschitz, otherwise it is *failed*. Let us construct the tree T_1 by taking T_0 and for its every leaf ℓ such that $f|_\ell$ is *not* average-μ-Lipschitz hang a tree T' obtained by applying Lemma 29 to $f|_\ell$ with $\varepsilon = 1/2$. Define $T_2, \dots, T_{\log(1/\varepsilon)}$ the same way: T_i is obtained from T_{i-1} by hanging trees given by Lemma 29 to all its failed leaves.

Then consider a random walk down $T_{\log(1/\varepsilon)}$: with probability $1/2$ it ends in a successful leaf of T_0, conditioned on it passing through a failed leaf of T_0 with probability $1/2$ it ends in a successful leaf of T_1. Hence with probability $1 - 2^{-\log(1/\varepsilon)} = 1 - \varepsilon$ the walk down $T_{\log(1/\varepsilon)}$ terminates in a successful leaf. □

5 Containment Lemma

In this section we prove a lemma that formalizes the intuition that if the sampled distribution has low entropy it must be very far from the target high-entropy distribution. While this is always true in a weak sense (see Lemma 4), for Lipschitz functions the distance can be boosted to exponentially close to 1.

Lemma 12. *Suppose $f\colon [n]^s \to [n]^m$ is an average-$n^{0.1}$-Lipschitz depth-d decision forest with $m \ge n^{0.99}$, $d = \log^{O(1)} n$, $\boldsymbol{u} \sim [n]^s$, and $\mathrm{H}(f(\boldsymbol{u})) \le m \log n/4$. Then there exists a set $F \subseteq [n]^m$ of size $n^{3m/4}$ such that*

$$\Pr[f(\boldsymbol{u}) \in F] \ge 1 - \exp(-n^{\Omega(1)}).$$

Our starting point is Lemma 4: since $\mathrm{H}(f(\boldsymbol{u})) \ge m \log n/4$, the lemma implies that there exists an event $G \subseteq [n]^m$ of size $n^{m/2}$ such that $\Pr[f(\boldsymbol{u}) \in G] \ge 1/2$. In other words, at least half of the elements in $[n]^s$ are mapped to a set of outputs of size at most $n^{m/2}$. Our goal is to enlarge this preimage so that it covers almost the entire $[n]^s$, while keeping the image size relatively small, as shown in Figure 2. To proceed further, we will need the following lemma, which we will prove later:

Lemma 30. *Let $T\colon \Sigma^M \to \{0,1\}$ be a depth-k decision tree with $\mu \coloneqq \Pr_{\boldsymbol{x}\sim\Sigma^M}[T(\boldsymbol{x})=1]$. Then, there exists 2-dimensional distribution $\mathcal{D}$ with marginals uniform over Σ^M and $T^{-1}(1)$ respectively such that*

$$\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[\mathrm{dist}(\boldsymbol{x},\boldsymbol{y})] \le O\left(\sqrt{k\log(1/\mu)}\right).$$

The set $f^{-1}(G)$ can be recognized by a decision tree $T'\colon [n]^s \to \{0,1\}$ of depth nd: $T(x)$ just computes $f_i(x)$ one by one and then accepts iff the resulting vector $(f_1(x),\dots,f_n(x))$ is in G. Then we are in a position to apply Lemma 30 with $\mu \coloneqq 1/2$: let $(\boldsymbol{u},\boldsymbol{y})$ be coupled with $\boldsymbol{u}\sim[n]^s$, $\boldsymbol{y}\sim f^{-1}(G)$ and

$$r \coloneqq O(\sqrt{nd}) \ge \mathbb{E}\left[\mathrm{dist}(\boldsymbol{u},\boldsymbol{y})\right] \ge \mathbb{E}\left[\min_{y\in f^{-1}(G)} \mathrm{dist}(\boldsymbol{u},y)\right] = \mathbb{E}[\mathrm{dist}(\boldsymbol{u},f^{-1}(G))], \tag{3}$$

At this point we use Lipschitzness. First, we apply Lemma 6 to get that f is in fact $(n^{0.2},\delta)$-Lipschitz for $\delta = \exp(-n^{\Omega(1)})$. We now claim that we can use Lipschitzness to say that $f(\boldsymbol{u})$ is close to G in expectation.

Claim 31. $\mathbb{E}[\mathrm{dist}(f(\boldsymbol{u}),G)] = \mathbb{E}[\min_{o\in G}\mathrm{dist}(f(\boldsymbol{u}),o)] \le O(n^{0.8})$.

Proof. Let $E\subseteq[n]^s$ be the set of inputs where $(n^{0.2},\delta)$-Lipschitzness is violated, i.e. some input cell $i\in[s]$ is queried by more than $n^{0.2}$ trees of f. We then have $\Pr_{\boldsymbol{u}\sim[n]^s}[\boldsymbol{u}\in E] \le s\delta = \exp(-n^{\Omega(1)})$. Consider any $x\notin E$ and $y\in[n]^s$. Then on the input x the set of input cells $\{i\in[s] \mid x_i\neq y_i\}$ is queried by at most $n^{0.2}\cdot\mathrm{dist}(x,y)$ trees in f. Let $D_x\subseteq[m]$ be the set of these trees. The output of the trees outside D_x does not change when we replace the input x with y: $f_{[n]\setminus D_x}(x) = f_{[n]\setminus D_x}(y)$. Consequently $\mathrm{dist}(f(x),f(y)) \le n^{0.2}\cdot\mathrm{dist}(x,y)$. Thus, we write

$$\mathbb{E}\left[\mathrm{dist}(f(\boldsymbol{u}),G)\right] \le \mathbb{E}\left[n^{0.2}\cdot\mathrm{dist}(\boldsymbol{u},f^{-1}(G)) \,\middle|\, \boldsymbol{u}\notin E\right] + s\cdot\Pr[\boldsymbol{u}\in E] = O(n^{0.8}). \qquad \square$$

Let $\mu = n^{0.1}$ be the average Lipschitzness parameter of f. Following the same strategy of clustering coordinates as in Lemma 25, we get a partition $I_1, I_2, \dots, I_\ell$ of $[s]$ with property

$$\sum_{i\in I_j} \mathbb{E}[\boldsymbol{\theta}_j] \le 2\mu$$

for every $j\in[\ell]$ ($\boldsymbol{\theta}_j$ here, as before, denotes number of trees querying j on input $\boldsymbol{u}$), with $\ell \le 3md/\mu$. Let $f'\colon [n]^{I_1}\times[n]^{I_2}\times\ldots\times[n]^{I_\ell}\to[n]^m$ be the clustered version of f. By construction, it is average-2μ-Lipschitz, so we apply Lemma 6 (setting the input alphabet $\Lambda \coloneqq [n]^{\max_{i\in[\ell]}|I_i|}$) and find that f' is $(6\mu d^2\log(1/\delta),\delta)$-Lipschitz, i.e. $(n^{0.2},\delta)$-Lipschitz for $\delta=\exp(-n^{\Omega(1)})$. We will abuse the notation and write $f'(x)$ for $x\in[n]^s$ meaning $f'(x_{I_1},x_{I_2},\dots,x_{I_\ell})$, and identify $[n]^s$ with $[n]^{I_1}\times[n]^{I_2}\times\ldots\times[n]^{I_\ell}$.

Let us define the function $h\colon [n]^{I_1}\times[n]^{I_2}\times\ldots\times[n]^{I_\ell}\to\mathbb{R}$ as follows:

$$h(x) \coloneqq \mathrm{dist}(f'(x),G) = \min_{o\in G}\mathrm{dist}(f'(x),o).$$

Let $S\subseteq[n]^s$ be the set of inputs, such that every position of f' is queried at most $n^{0.2}$ times. By the definition of Lipschitzness, $\Pr_{\boldsymbol{u}\sim[n]^s}[\boldsymbol{u}\in S] \ge 1-s\delta$. Analogously to Claim 31, h has $n^{0.2}$-bounded differences over S, so we apply Lemma 22 and get

$$\Pr_{\boldsymbol{u}\sim[n]^s}\left[|h(\boldsymbol{u}) - \mathop{\mathbb{E}}_{\boldsymbol{u}\sim[n]^s}[h(\boldsymbol{u}) \mid \boldsymbol{u}\in S]| \ge \lambda + s\delta\cdot n^{0.2}\cdot\tfrac{2md}{\mu}\right] \le$$

$$\le 2s\delta + 2\exp\left(-\Omega\left(\frac{\lambda^2}{n^{0.4}\cdot md/\mu}\right)\right).$$

So, substituting $\lambda = n^{0.8}$ and noting that $|\mathbb{E}[h(\boldsymbol{u}) \mid \boldsymbol{u} \in S] - \mathbb{E}[\mathrm{dist}(f(\boldsymbol{u}), G)]| \le s\delta m$, we get $\Pr\left[f(\boldsymbol{u}), G) \ge Cn^{0.8}\right] \le \exp(-n^{\Omega(1)})$ for a large enough constant $C > 0$.

Finally, we define $F \coloneqq \mathcal{N}_{Cn^{0.8}}(G)$, where $\mathcal{N}_r(P) \coloneqq \{x \mid \exists y \in P \ \mathrm{dist}(x, y) \le r\}$. We can upper bound the size of F as $|F| \le |G| \cdot \binom{n}{Cn^{0.8}} \cdot n^{Cn^{0.8}} \le n^{3m/4}$ and for such F, we have $\Pr[f(\boldsymbol{u}) \in F] \ge 1 - \exp(-n^{\Omega(1)})$.

5.1 Proof of Lemma 30

Recall that the statistical (total variation) distance between two *distributions* ν_1 and ν_2 can be defined through optimal couplings:

$$\Delta(\nu_1, \nu_2) = \min\left\{ \Pr_{\boldsymbol{a},\boldsymbol{b}\sim\mathcal{C}}[\boldsymbol{a} \ne \boldsymbol{b}] \;\middle|\; \mathcal{C}\text{: distribution with marginals } \nu_1,\ \nu_2 \right\}.$$

We say that $\mathcal{C}$ that achieves this minimum[5] is *the optimal coupling* of ν_1 and ν_2.

The coupling $\mathcal{D}$ claimed by Lemma 30 is basically a composition of couplings $\mathcal{C}$ for each pair of symbols in $\boldsymbol{x}$ and $\boldsymbol{y}$: for each node of T querying i we enforce that the distributions of $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ conditioned on both $\boldsymbol{x}$ and $\boldsymbol{y}$ passing through the node are optimally coupled, see Algorithm 1.

Wlog we assume that T is a full depth-k decision tree, and at any path, it does not query the same input cell twice. Let $\boldsymbol{x} \sim \Sigma^M$. We denote by $\mathsf{path}(r)$ the computation path of r in T. We construct $\boldsymbol{y} \in T^{-1}(1)$ using Algorithm 1.

Algorithm 1 The algorithm defining the coupling $\mathcal{D}$.

$\boldsymbol{y} \leftarrow \boldsymbol{x}$.
$v \leftarrow$ root of T.
while v is not a leaf **do**
    Let $i \in [M]$ be the coordinate queried by v, and $\{w_i\}_{i\in\Sigma}$ be its children.
    Let $\mathcal{D}$ be the optimal coupling of $\boldsymbol{x}_i$ and $(\boldsymbol{z}_i \mid \mathsf{path}(\boldsymbol{z}) \ni v)$ where $\boldsymbol{z} \sim T^{-1}(1)$.
    Define $(\boldsymbol{y}_i \mid \mathsf{path}(\boldsymbol{y}) \ni v)$ such that $(\boldsymbol{x}_i, (\boldsymbol{y}_i \mid \mathsf{path}(\boldsymbol{y}) \ni v))$ is distributed according to $\mathcal{C}$.
    $v \leftarrow w_{\boldsymbol{y}_i}$.
end while

Let $\boldsymbol{z} \sim T^{-1}(1)$. By definition for every internal node v of T that queries i, the distributions $(\boldsymbol{y}_i \mid \mathsf{path}(\boldsymbol{y}) \ni v)$ and $(\boldsymbol{z}_i \mid \mathsf{path}(\boldsymbol{z}) \ni v)$ coincide. Thus, the next branch taken from v on $\mathsf{path}(\boldsymbol{y})$ has the same conditional distribution as for $\boldsymbol{z}$; by backward induction over the depth, $\mathsf{path}(\boldsymbol{y})$ is uniform over accepting paths. The coordinates not queried by T remain uniform ($\boldsymbol{y}_j = \boldsymbol{x}_j$), therefore $\boldsymbol{y}$ is uniform over $T^{-1}(1)$.

Now it remains to bound $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y})$. Let $\boldsymbol{p} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k) \in \Sigma^k$ encode a uniformly random *accepting* root-to-leaf path in T, and for $\tau \in \{0,1\}^j$ let $v(\tau)$ denote the node in T that is reached by going from the root according to τ. At a node v querying i let

$$\delta_v \coloneqq \Pr[\boldsymbol{y}_i \ne \boldsymbol{x}_i \mid \mathsf{path}(\boldsymbol{x}) \ni v \wedge \mathsf{path}(\boldsymbol{y}) \ni v] = \Delta(\boldsymbol{x}_i, (\boldsymbol{y}_i \mid \mathsf{path}(\boldsymbol{y}) \ni v)).$$

Then

$$\mathbb{E}[\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y})] \le \sum_{j\in[k]} \mathbb{E}[\delta_{v(\boldsymbol{p}_{\le j})}].$$

[5] We work with variables over a finite support, so the minimum always exists.

Pinsker's inequality (see [CT06, Lemma 11.6.1]) applied to $(\boldsymbol{y}_i \mid \mathsf{path}(\boldsymbol{y}) \ni v)$ and the uniform distribution $\boldsymbol{u} \sim \Sigma$ (i.e. the distribution of $\boldsymbol{x}_i$) states

$$\delta_v^2 = \Delta((\boldsymbol{y}_i \mid \mathsf{path}(\boldsymbol{y}) \ni v), \boldsymbol{u})^2 \le 2 \ln 2 (\mathrm{H}(\boldsymbol{u}) - \mathrm{H}(\boldsymbol{y}_i \mid \mathsf{path}(\boldsymbol{y}) \ni v)),$$

so $\log|\Sigma| - \delta_v^2/2\ln 2 \ge \mathrm{H}(\boldsymbol{y}_i \mid \mathsf{path}(\boldsymbol{y}) \ni v)$. The event "$\mathsf{path}(\boldsymbol{y}) \ni v$" is equivalent to "$\boldsymbol{p}_{<j} = \alpha$", and $\boldsymbol{p}_j$ is determined by $\boldsymbol{y}_i$. Hence, we have

$$\mathrm{H}(\boldsymbol{p}_j \mid \boldsymbol{p}_{<j} = \alpha) \le \log|\Sigma| - \delta_{v(\alpha)}^2/2\ln 2.$$

Taking the expectation over α, and summing over j, we obtain

$$k\log|\Sigma| - \Big(\sum_{j\in[k]} \mathbb{E}[\delta^2_{v(\boldsymbol{p}_{\le j})}]\Big)/2\ln 2 \ge \sum_{j\in[k]} \mathrm{H}(\boldsymbol{p}_j \mid \boldsymbol{p}_{<j}) = \mathrm{H}(\boldsymbol{p}) = k\log|\Sigma| - \log(1/\mu).$$

By Jensen's and Cauchy–Schwarz's inequalities we get,

$$2\ln 2 \cdot \log(1/\mu) \ge \sum_{j\in[k]} \mathbb{E}[\delta^2_{v(\boldsymbol{p}_{\le j})}] \ge \sum_{j\in[k]} \Big(\mathbb{E}[\delta_{v(\boldsymbol{p}_{\le j})}]\Big)^2 \ge \frac{1}{k}\Big(\sum_{j\in[k]} \mathbb{E}[\delta_{v(\boldsymbol{p}_{\le j})}]\Big)^2$$

Therefore,

$$\mathbb{E}[\mathrm{dist}(\boldsymbol{x},\boldsymbol{y})] \le \sum_{j\in[k]} \mathbb{E}[\delta_{v(\boldsymbol{p}_{\le j})}] \le O\left(\sqrt{k\log(1/\mu)}\right).$$

6 Collision Lemma

In this section, we prove the general version of our collision lemma. We first prove the version for the independent random variables (that directly generalizes Lemma 5) and then will derive Lemma 16, which is a simple corollary of the independent variables case.

Lemma 32. *Let $\boldsymbol{z}_1, \dots, \boldsymbol{z}_m$ be independent random variables supported over $[n] \cup \{\perp\}$. Assume that for every i we have $\mathrm{H}(\boldsymbol{z}_i) \ge \delta \cdot m \log n$ and $m \ge n^{1-\varepsilon}$ and $\delta = \delta(n) \ge \max(2\log\log n/\log n, 4\varepsilon)$. Then*

$$\Pr[\exists i \ne j \in [m] \colon \boldsymbol{z}_i = \boldsymbol{z}_j \ne \perp] \ge 1 - \exp(-\Omega(\delta^4 m^3/n^2)).$$

6.1 Proof of Lemma 32

Our goal is to show that with high probability there exists a collision among the $\boldsymbol{z}_i$'s. We visualize the setting with a complete bipartite graph $H = ([m], [n]\cup\{\perp\}, [m] \times ([n]\cup\{\perp\}))$ in Figure 4, the upper part nodes correspond to the variables $\boldsymbol{z}_1, \dots, \boldsymbol{z}_m$, the lower part nodes correspond to their values in $[n]\cup\{\perp\}$, every edge (i,k) is labeled with the probability $p_k^i := \Pr[\boldsymbol{z}_i = k]$.

Then every value $z_1, \dots, z_m \in ([n]\cup\{\perp\})^m$ corresponds to a set of edges $\{(i, z_i) \mid i \in [m]\}$ in H. We then have a collision iff there is a node $k \in [n]$ with a degree at least 2 in the edges $\{(i, z_i) \mid i \in [m]\}$. We define the function f as a smoothed version of counting the number of nodes in $[m]$ of degree at least 2.

$$f(z_1, \dots, z_m) := \sum_{k\in[n]} \max\big(0, \big|\{i \in [m] \mid z_i = k\}\big| - 1\big), \tag{4}$$

Then by definition there is a collision if and only if the value of f is non-zero:

$$\Pr[\exists i \ne j \colon \boldsymbol{z}_i = \boldsymbol{z}_j \ne \perp] = \Pr[f(\boldsymbol{z}_1, \dots, \boldsymbol{z}_m) \ne 0],$$

thus it suffices to show that $f(\boldsymbol{z}_1, \dots, \boldsymbol{z}_m)$ is nonzero with high probability. The key feature of thus defined f is that it satisfies the 2-bounded differences property over its entire domain: indeed if z and z' differ in a coordinate $i \in [m]$ then the only summands in (4) that may change are $k \in \{z_i, z'_i\}$, each of those can change by at most by one. Therefore McDiarmid's inequality applies:

$$\Pr[f(\boldsymbol{z}) = 0] \le \Pr[f(\boldsymbol{z}) \le \mathbb{E}[f(\boldsymbol{z})]/2] = \exp(-\Omega(\mathbb{E}[f(\boldsymbol{z})]^2/m)).$$

Thus, the main technical part of the proof is to show that $\mathbb{E}[f(\boldsymbol{z})] = \Omega(\delta^2 m^2/n)$.

Bounding the expectation. By the definition of f we have $\mathbb{E}[f(\boldsymbol{z})] \ge \sum_{k\in[n]} \Pr[\exists i \ne j \in [m] \colon \boldsymbol{z}_i = \boldsymbol{z}_j = k]$. For each $k \in [n]$ the probability that there exist $\boldsymbol{z}_i = \boldsymbol{z}_j = k$ can be computed explicitly: indeed we have independent events "$\boldsymbol{z}_i = k$" and the probability we bound is that at least two of these events occur. If all $\Pr[\boldsymbol{z}_i = k]$ are small enough, we can use the following simple bound:

Claim 33. *Suppose events $E_1, \dots, E_\ell$ are independent, each E_i occurs with probability $q_i \le \alpha$ such that $\bar{q} \coloneqq \sum_{i\in[\ell]} q_\ell \le 1/8$. Then*

$$\Pr\Big[\sum_{i\in[\ell]} [\![E_i]\!] \ge 2\Big] \ge \bar{q}^2/4 - 2\alpha\bar{q}.$$

Let $\bar{p}_k \coloneqq \sum_{i\in[n]} p_k^i$. If Claim 33 applied for every $k \in [n]$ with a negligible α, we would get that $\mathbb{E}[f(\boldsymbol{z})]$ is at least $(1 - o(1)) \cdot \sum_{k\in[n]} \bar{p}_k^2$. Since $\sum_{k\in[n]} \bar{p}_k \approx m$, we could apply Cauchy-Shwartz's inequality to get the desired bound. We face two technical problems: $\bar{p}_k$ might be too large, and p_k^i (and thus α) may not be negligible. We fix both of those directly: remove too large p_k^i and split the node k in the graph H into several nodes so that the $\bar{p}$-value for each of the nodes does not exceed $1/8$.

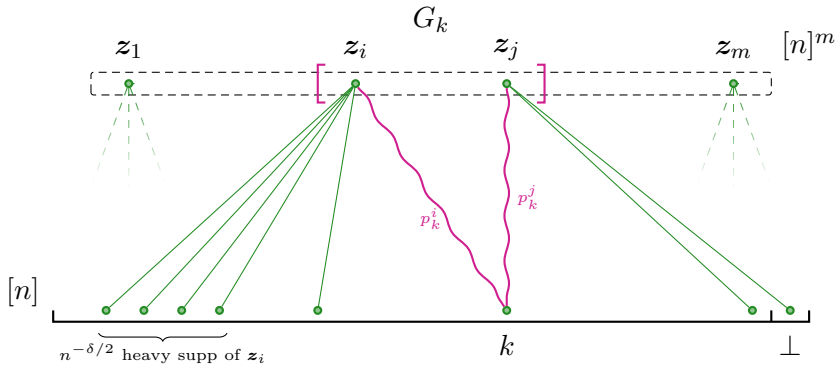

Figure 4: This picture illustrates the approach to prove that the probability of having *collision value* k is significant, i.e. $p \coloneqq \Pr[\exists i \ne j \in [m] \colon \boldsymbol{z}_i = \boldsymbol{z}_j = k]$ is bounded away from zero. The key technical step in the proof is to remove "heavy" edges from the graph—the ones with $p_k^i > n^{-\delta/2}$. G_k denotes the neighborhood of k in H with all heavy edges removed.

In the first step we remove from H all edges with too large p_k^i, the following claim implies that we retain significant probability mass after this removal:

Claim 34. *Let $\boldsymbol{a}$ be a random variable over $[n]$ with $\mathrm{H}(\boldsymbol{a}) \ge c \log n$, where $c = c(n) > 4/n$. Then with $p \colon [n] \to [0, 1]$ being the probability function of $\boldsymbol{a}$, we have*

$$\Pr[p(\boldsymbol{a}) \le n^{-c/2}] = \sum_{p(i)\le n^{-c/2}} p(i) \ge c/8.$$

Applying Claim 34 with $\boldsymbol{a} = \boldsymbol{z}_i$, $c = \delta$ we get that $\sum_{(i,k)\in[m]\times([n]\cup\{\perp\})} p_k^i \geq \delta m/8$. Observe that we can now remove the node $\perp$ and still retain most of the probability mass:

$$\sum_{i\in[m]\colon p_\perp^i \leq n^{-\delta/2}} p_\perp^i \leq mn^{-\delta/2} \leq m/\log n \leq \delta m/16.$$

The last two inequalities utilize $\delta \geq 2\log\log n/\log n$. Thus, we get

$$\sum_{(i,k)\in[m]\times[n]\colon p_k^i \leq n^{-\delta/2}} p_k^i \geq \Omega(\delta m).$$

Let $G \subseteq [m]\times[n]$ be the set of the light edges of H defined as $G \coloneqq \{(i,k) \mid p_k^i \leq n^{-\delta/2}\}$ and for every $k \in [n]$ let $G_k \coloneqq \{i \in [m] \mid (i,k) \in G\}$ and $\bar{p}_k \coloneqq \sum_{i\in G_k} p_k^i$. In order to force $\bar{p}_k \leq 1/8$, we split G_k into the smallest number of parts $G_k^1, \ldots G_k^{t_k}$ with the property

$$\forall h \in [t_k] \sum_{i\in G_k^h} p_k^i \leq 1/8.$$

It is possible, since $n^{-\delta/2} \leq 1/8$, and we will get $t_k \leq 16\bar{p}_k + 1$, so the total number of parts will be $O(n)$. Clearly,

$$\mathbb{E}[f(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m)] \geq \sum_{k\in[n]} \sum_{h\in[t_k]} \Pr[\exists i \neq j \in G_k^h \colon \boldsymbol{z}_i = \boldsymbol{z}_j = k].$$

Now we denote $\bar{p}_{kh} = \sum_{i\in G_k^h} p_k^i$ and get by Claim 33 applied to the events "$\boldsymbol{z}_i = k$" for $i \in G_k^h$ that:

$$\Pr[\exists i \neq j \in G_k^h \colon \boldsymbol{z}_i = \boldsymbol{z}_j = k] \geq \bar{p}_{kh}^2/4 - 2n^{-\delta/2}\bar{p}_{kh}.$$

Finally, we obtain

$$\begin{aligned}
\mathbb{E}[f(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m)] &\geq \frac{1}{4} \sum_{k\in[n]} \sum_{h\in[t_k]} \bar{p}_{kh}^2 - 2n^{-\delta/2} \sum_{k\in[n]} \sum_{h\in[t_k]} \bar{p}_{kh} \\
\text{(by Cauchy–Schwarz's inequality)} \quad &\geq \left(\sum_{k\in[n]} \sum_{h\in[t_k]} \bar{p}_{kh} \right)^2 / O(n) - 2n^{-\delta/2} \sum_{k\in[n]} \sum_{h\in[t_k]} \bar{p}_{kh} \\
\text{(using } \delta \geq 4\varepsilon) \quad &\geq \frac{1}{2} \left(\sum_{k\in[n]} \sum_{h\in[t_k]} \bar{p}_{kh} \right)^2 / O(n) \geq \Omega(\delta^2 m^2/n).
\end{aligned}$$

6.1.1 Proof of Claim 33

Let $p \coloneqq \Pr[\sum_{i\in[\ell]} [\![E_i]\!] \geq 2]$. The event "$\sum_{i\in[\ell]} [\![E_i]\!] \geq 2$" does not occur iff none of E_i occur, or exactly one of them occurs. Thus

$$p = 1 - \prod_{i\in[\ell]} (1-q_i) - \sum_{i\in[\ell]} q_i/(1-q_i) \cdot \prod_{i\in[\ell]} (1-q_i).$$

We then rewrite using $q_i \leq \alpha$: $1 - p \leq \prod_{i\in[\ell]}(1-q_i) \cdot (1 + \bar{q}/(1-\alpha))$. Since the geometric mean does not exceed the arithmetic mean, we have $\prod_{i\in[\ell]}(1-q_i) \leq (\sum_{i\in[\ell]}(1-q_i)/\ell)^\ell = (1-\bar{q}/\ell)^\ell$. Moreover, since $\alpha \leq \bar{q} < 1/2$ we have $1/(1-\alpha) \leq 1 + 2\alpha$. Then

$$1 - p \leq (1-\bar{q}/\ell)^\ell (1 + \bar{q}(1+2\alpha)).$$

Now we use a simple analytical fact to bound the first multiplier:

Fact 4. *For $x \in (0,1)$ and $n \geq 2$ the inequality $(1-x)^n \leq 1 - nx + (nx)^2/2$ holds.*

Proof. By Taylor's theorem [Rud76, Theorem 5.15] for $(1-x)^n$ at $x_0 = 0$ we get that there exists $\xi \in (0,x)$ such that $(1-x)^n = 1 - nx + n(n-1)/2 \cdot (1-\xi)^{n-2}x^2 \leq 1 - nx + (nx)^2/2$. □

Then by Fact 4 we conclude:

$$\begin{aligned} 1 - p &\leq (1 - \bar{q} + \bar{q}^2/2)(1 + \bar{q}(1+2\alpha)) \\ &= 1 - \bar{q}^2/2 + 2\alpha\bar{q} + \bar{q}^3\alpha + \bar{q}^3/2 - 2\alpha\bar{q}^2 \\ (\text{since } \bar{q} \leq 1/8) \quad &\leq 1 - \bar{q}^2/4 + 2\alpha\bar{q}. \end{aligned}$$

6.1.2 Proof of Claim 34

$$\sum_{p(i) > n^{-c/2}} -p(i)\log p(i) \leq \sum_{p(i) > n^{-c/2}} \frac{c}{2} \cdot \log n \cdot p(i) \leq c \log n/2,$$

Function $x \mapsto -x \log x$ is ascending at $[0, 1/e]$, so

$$\sum_{p(i) < n^{-2}} -p(i)\log p(i) \leq n \cdot n^{-2} \cdot \log n = \log n/n \leq c \log n/4.$$

Combining the above inequalities with $\mathrm{H}(\boldsymbol{a}) \geq c \log n$, we get

$$\sum_{n^{-2} \leq p(i) \leq n^{-c/2}} p(i) \geq \sum_{n^{-2} \leq p(i) \leq n^{-c/2}} -\frac{p(i)\log p(i)}{2\log n} \geq \frac{c\log n/4}{2\log n} = c/8$$

6.2 Depth-1 decision forests

In this section we prove a natural corollary of Lemma 32: virtually the same bound holds for functions that are computable with depth-1 decision forests.

Lemma 16. *Let $\boldsymbol{u} \sim [n]^s$ and $f\colon [n]^s \to ([n] \cup \{\perp\})^m$ be a depth-1 decision forest. Suppose that $\mathrm{H}(f(\boldsymbol{u})) \geq \delta \cdot m \log n$, and $(\delta^2/4)m \geq n^{1-\varepsilon}$ for $\delta = \delta(n) \geq \max(4\log\log n/\log n, 8\varepsilon)$. Then*

$$\Pr[\exists i \neq j \in [m]\colon f_i(\boldsymbol{u}) = f_j(\boldsymbol{u}) \neq \perp] \geq 1 - \exp(-\Omega(\delta^4 m^3/n^2)).$$

Proof. In order to apply Lemma 32 we need to choose a subset $I \subseteq [m]$ such that the output cells in I are independent and the entropy rate is not reduced too severely. All trees in f can be partitioned into subsets $J_1 \sqcup \cdots \sqcup J_\ell = [m]$ where in each J_i the trees query the same input cell, which is different for different sets. Observe that $\mathrm{H}(f_{J_i}(\boldsymbol{u})) \leq \log n$. Thus, we have that $\ell \geq \mathrm{H}(f(\boldsymbol{u}))/\log n \geq \delta m$.

We first form I' by taking a representative $j_i \in J_i$ that maximizes $\mathrm{H}(f_{j_i}(\boldsymbol{u}))$ for each $i \in [\ell]$. We use the following simple fact

Fact 5. *For $a_1, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{R}_{\geq 0}$ we have $\sum_{i\in[n]}(a_i/b_i) \geq (\sum_{i\in[n]} a_i)^2/(\sum_{i\in[n]} a_i b_i)$.*

Proof. We first rewrite

$$\sum_{i\in[n]} \frac{a_i}{b_i} = \Big(\sum_{i\in[n]} a_i\Big) \cdot \sum_{i\in[n]} \frac{a_i}{\sum_{i\in[n]} a_i} \cdot \frac{1}{b_i}.$$

Then applying Jensen inequality for the function $1/x$ we get that the right multiplier is at least $(\sum_{i\in[n]} a_i)/(\sum_{i\in[n]} a_i b_i)$, which concludes the proof. □

Then we get

$$\mathrm{H}(f_{I'}(\boldsymbol{u})) = \sum_{i\in[\ell]} \mathrm{H}(f_{j_i}(\boldsymbol{u})) \geq \sum_{i\in[\ell]} \frac{\mathrm{H}(f_{J_i}(\boldsymbol{u}))}{|J_i|} \overset{\text{Fact 5}}{\geq}$$

$$\frac{(\mathrm{H}(f(\boldsymbol{u})))^2}{\sum_{i\in[\ell]} |J_i|\,\mathrm{H}(f_{J_i}(\boldsymbol{u}))} \geq \frac{(\delta m \log n)^2}{\log n \cdot m} = \delta^2 m \log n \geq \delta\ell \log n.$$

Now we pick $I \subseteq I'$ of $i \in I'$ such that $f_i(\boldsymbol{u}) \geq \delta \log n/2$. Then $\delta\ell \log n \leq \mathrm{H}(f_{I'}(\boldsymbol{u})) \leq |I| \log n + (\ell - |I|)\delta \log n/2$, so $|I| \geq \delta\ell(1-\delta/2)/2 \geq \delta\ell/4$. Then we apply Lemma 32 to f_I, so $m' = (\delta^2/4)\cdot m$, and $\mathrm{H}(f_i(\boldsymbol{u})) \geq (\delta/2)\log n$, so we need $\delta/2 \geq \max(2\log\log n/\log n, 4\varepsilon)$, which is what we have by the assumption. □

6.3 Simplified collision lemmas

In this section, we derive Lemmas 5 and 10.

Lemma 5. *Let $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$ be independent random variables over $[n]$ such that $\mathrm{H}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m) \geq (m \log n)/8$ for $m \geq n^{0.99}$. Then $\Pr[\exists i \neq j \in [m] \colon \boldsymbol{z}_i = \boldsymbol{z}_j] \geq 1 - o(1)$.*

Proof. We apply Lemma 16 with $\delta = 1/8$, and $\varepsilon = 1/64$. It is easy to check that $(\delta^2/4)m \geq n^{1-\varepsilon}$ for large enough n since $(\delta^2/4)m = n^{0.99}/256$ and $n^{1-\varepsilon} = n^{1-1/64} = o(n^{0.99})$. □

Lemma 10. *Let $\boldsymbol{u} \sim [n]^s$ and $f \colon [n]^s \to [n]^n$ be a depth-1 decision forest. Suppose that $\mathrm{H}(f(\boldsymbol{u})) \geq 4(n \log\log n)$. Then $\Pr[\exists i \neq j \in [n] \colon f_i(\boldsymbol{u}) = f_j(\boldsymbol{u})] \geq 1 - \exp(-\Omega(n/\mathrm{poly}(\log n)))$.*

Proof. We apply Lemma 16 with $\delta = 4\log\log n/\log n$, $\varepsilon = \log\log n/(16\log n)$, and $m = n$ Then $(\delta^2/4)m = 4n(\log\log n)^2/\log^2 n$, and $n^{1-\varepsilon} = n \cdot \exp(-\log n \cdot \log\log n/(16\log n)) = n \cdot \exp(-\log\log n/16) = n \cdot \log^{-1/16} n$. Thus, the conditions of Lemma 16 are satisfied. □

7 Open questions

Quantitative improvements. Our lower bounds can potentially be quantitatively improved in many ways:

- Can the adaptive cell-probe bound be improved to, say $\Omega(\log n/\log\log n)$?
- Can the distance bound be improved to $1 - \exp(-n^{1-o(1)})$?
- What is the right bound for the number of nonadaptive probes in Theorem 2?

Our lower bound actually works for sampling $m = n^{1-\varepsilon}$ distinct elements that can be sampled with $O(\log n)$ adaptive *bit-probes* [Czu15]. Can one show a better upper bound for *cell-probes* in that case, say $O(\log n/\log\log n)$?

Can any of the lower bounds be improved if s is bounded? That would be sufficient for improving the data structure lower bounds in Corollary 3.

Other distributions. What *symmetric* distributions in $\{0,1\}^n$ are samplable in $O(1)$ *nonadaptive* cell-probes to $[n]^{\mathbb{N}}$? [KOW25] show that essentially the only nontrivial distribution that can be sampled with *nonadaptive bit-probes* is uniform over odd Hamming weight vectors. For the large input alphabet, this does not hold. For example, one can sample the uniform distribution over $\binom{[n]}{k}$ with k nonadaptive cell-probes (as well as *any* distribution uniform over a set of size n^k). On the other hand, it is hard even with $o(\log(n/k)/\log\log(n/k))$ adaptive *bit* probes [FLRS23].

We conjecture that the uniform distribution over $\binom{[n]}{n/2}$ requires $\omega(1)$ adaptive cell-probes to sample. Showing this even in the nonadaptive case for a *fixed constant* number of nonadaptive probes is open.

An intermediate challenge is to show that *permutation matrices* are hard to sample with adaptive cell-probes. The target distribution is $\boldsymbol{M} \in \{0,1\}^{n \times n}$ and $\boldsymbol{M}$ is uniform over matrices with exactly one 1-entry in every row and column. An efficient cell-probe sampler for $\boldsymbol{M}$ is not directly ruled out by our theorems even for the nonadaptive case, but we think that the technique should translate for that case as well.

Acknowledgments

We thank Ziyi Guan, Gilbert Maystre, and Weiqiang Yuan for discussions related to Lemma 30.

References

[Bab87] László Babai. Random oracles separate pspace from the polynomial-time hierarchy. *Information Processing Letters*, 26(1):51–53, 1987. doi:10.1016/0020-0190(87)90036-6.

[BIL12] Chris Beck, Russell Impagliazzo, and Shachar Lovett. Large deviation bounds for decision trees and sampling lower bounds for ac0-circuits. In *FOCS 2012*, pages 101–110, 2012. doi:10.1109/FOCS.2012.82.

[CGFS86] Fan Chung, Ronald Graham, Péter Frankl, and James Shearer. Some intersection theorems for ordered sets and graphs. *Journal of Combinatorial Theory, Series A*, 43(1):23–37, 1986. doi:10.1016/0097-3165(86)90019-1.

[CGZ22] Eshan Chattopadhyay, Jesse Goodman, and David Zuckerman. The Space Complexity of Sampling. In *ITCS 2022*, volume 215, pages 40:1–40:23, Dagstuhl, Germany, 2022. doi:10.4230/LIPIcs.ITCS.2022.40.

[CK00] Artur Czumaj and Miroslaw Kutylowski. Delayed path coupling and generating random permutations. *Random Struct. Algorithms*, 17(3–4):238–259, October 2000.

[Com15] Richard Combes. An extension of mcdiarmid's inequality. *CoRR*, abs/1511.05240, 2015. arXiv:1511.05240.

[CS16] Gil Cohen and Leonard J. Schulman. Extractors for near logarithmic min-entropy. In *FOCS 2016*, pages 178–187, 2016. doi:10.1109/FOCS.2016.27.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *STOC 2016*, page 670–683, New York, NY, USA, 2016. doi:10.1145/2897518.2897528.

[Czu15] Artur Czumaj. Random permutations using switching networks. In *STOC 2015*, pages 703–712, 2015. doi:10.1145/2746539.2746629.

[Dur64] Richard Durstenfeld. Algorithm 235: Random permutation. *Commun. ACM*, 7(7):420, July 1964. doi:10.1145/364520.364540.

[FLRS23] Yuval Filmus, Itai Leigh, Artur Riazanov, and Dmitry Sokolov. Sampling and certifying symmetric functions. In *APPROX/RANDOM 2023*, volume 275, pages 36:1–36:21, 2023. doi:10.4230/LIPICS.APPROX/RANDOM.2023.36.

[FSS84] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, 1984. doi:10.1007/BF01744431.

[GKM+25] Daniel Grier, Daniel M. Kane, Jackson Morris, Anthony Ostuni, and Kewen Wu. Quantum advantage from sampling shallow circuits: Beyond hardness of marginals, 2025. arXiv:2510.07808.

[Gol09] Alexander Golynski. Cell probe lower bounds for succinct data structures. In *SODA 2009*, pages 625–634, 2009. doi:10.1137/1.9781611973068.69.

[Hag91] Torben Hagerup. Fast parallel generation of random permutations. In *Automata, Languages and Programming*, pages 405–416, Berlin, Heidelberg, 1991.

[Har66] L. H. Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1:385–393, 1966.

[KK18] Timo Kötzing and Martin S. Krejca. First-hitting times under additive drift, 2018. URL: https://arxiv.org/abs/1805.09415, arXiv:1805.09415.

[KOW24] Daniel M. Kane, Anthony Ostuni, and Kewen Wu. Locality bounds for sampling hamming slices. In *STOC 2024*, pages 1279–1286, 2024. doi:10.1145/3618260.3649670.

[KOW25] Daniel M. Kane, Anthony Ostuni, and Kewen Wu. Locally sampleable uniform symmetric distributions. In *STOC 2025*, pages 1807–1816, 2025. doi:10.1145/3717823.3718243.

[LV12] Shachar Lovett and Emanuele Viola. Bounded-depth circuits cannot sample good codes. *Comput. Complexity*, 21(2):245–266, 2012. doi:10.1007/s00037-012-0039-3.

[Mar06] A.A. Markov. Extension of the law of large numbers to quantities, depending on each other (1906). reprint. *Journal Électronique d'Histoire des Probabilités et de la Statistique [electronic only]*, 2(1b):Article 10, 12 p., electronic only–Article 10, 12 p., electronic only, 2006.

[McD89] Colin McDiarmid. *On the method of bounded differences*, page 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989.

[Mor08] Ben Morris. The mixing time of the thorp shuffle. *SIAM Journal on Computing*, 38(2):484–504, 2008. doi:10.1137/050636231.

[Mor09] Ben Morris. Improved mixing time bounds for the thorp shuffle and l-reversal chain. *The Annals of Probability*, 37(2):453–477, 2009.

[Mor13] Ben Morris. Improved mixing time bounds for the thorp shuffle. *Comb. Probab. Comput.*, 22(1):118–132, January 2013. doi:10.1017/S0963548312000478.

[MRRS12] Ian Munro, Rajeev Raman, Venkatesh Raman, and Rao Srinivasa. Succinct representations of permutations and functions. *Theor. Comput. Sci.*, 438:74–88, June 2012. doi:10.1016/j.tcs.2012.03.005.

[MRS09] Ben Morris, Phillip Rogaway, and Till Stegers. How to encipher messages on a small domain. In *CRYPTO 2009*, pages 286–302, Berlin, Heidelberg, 2009.

[MT06] Ravi Montenegro and Prasad Tetali. *Mathematical Aspects of Mixing Times in Markov Chains*. 2006. doi:10.1561/0400000003.

[MV91] Yossi Matias and Uzi Vishkin. Converting high probability into nearly-constant time—with applications to parallel hashing. In *STOC 1991*, page 307–316, New York, NY, USA, 1991. doi:10.1145/103418.103453.

[Rei85] John H Reif. An optimal parallel algorithm for integer sorting. In *FOCS 1985*, pages 496–504. IEEE, 1985.

[Rud76] W. Rudin. *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1976.

[SS24] Ronen Shaltiel and Jad Silbak. Explicit codes for poly-size circuits and functions that are hard to sample on low entropy distributions. In *STOC 2024*, page 2028–2038, New York, NY, USA, 2024. doi:10.1145/3618260.3649735.

[Tho73] Edward Thorp. Nonrandom shuffling with applications to the game of faro. *Journal of the American Statistical Association*, 68(344):842–847, 1973. doi:10.2307/2284510.

[Vio12a] Emanuele Viola. Bit-probe lower bounds for succinct data structures. *SIAM Journal on Computing*, 41(6):1593–1604, 2012.

[Vio12b] Emanuele Viola. The complexity of distributions. *SIAM Journal on Computing*, 41(1):191–218, 2012. doi:10.1137/100814998.

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

[Vio18] Emanuelle Viola. The complexity of distributions, 2018. Talk at Simons Institute. URL: https://www.youtube.com/live/O78b085HE3w?si=i7e44r9QuNzrR2dV&t=324.

[Vio20] Emanuele Viola. Sampling lower bounds: Boolean average-case and permutations. *SIAM Journal on Computing*, 49(1):119–137, 2020. doi:10.1137/18M1198405.

[Vio23] Emanuele Viola. New sampling lower bounds via the separator. In *CCC 2023*, volume 264, pages 26:1–26:23, 2023. doi:10.4230/LIPICS.CCC.2023.26.

[WP23] Adam Bene Watts and Natalie Parham. Unconditional quantum advantage for sampling with shallow circuits, 2023. arXiv:2301.00995.

[Yao81] Andrew Chi-Chih Yao. Should tables be sorted? *J. ACM*, 28(3):615–628, July 1981. doi:10.1145/322261.322274.

[Yu20] Huacheng Yu. Nearly optimal static las vegas succinct dictionary. In *STOC 2020*, page 1389–1401, New York, NY, USA, 2020. doi:10.1145/3357713.3384274.

[YZ24] Huacheng Yu and Wei Zhan. Sampling, Flowers and Communication. In *ITCS 2024*, volume 287, pages 100:1–100:11, 2024. doi:10.4230/LIPIcs.ITCS.2024.100.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il