Electronic Colloquium on Computational Complexity, Report No. 156 (2014)

A Chasm Between Identity and Equivalence Testing with Conditional Queries

Jayadev Acharya* Clément L. Canonne† Gautam Kamath‡

November 26, 2014

Abstract

A recent model for property testing of probability distributions [CFGM13, CRS12] enables tremendous savings in the sample complexity of testing algorithms, by allowing them to condition the sampling on subsets of the domain.

In particular, Canonne et al. [CRS12] showed that, in this setting, testing identity of an unknown distribution D (i.e., whether $D = D^*$ for an explicitly known D^*) can be done with a *constant* number of samples, independent of the support size n – in contrast to the required $\sqrt{n}$ in the standard sampling model. However, it was unclear whether the same held for the case of testing equivalence, where *both* distributions are unknown. Indeed, while the best known upper bound for equivalence testing is polylog(n) [CRS12], whether a dependence on the domain size n is necessary was still open, and explicitly posed at the Bertinoro Workshop on Sublinear Algorithms [Sublinear.info, Problem 66]. In this work, we answer the question in the positive, showing that any testing algorithm for equivalence must make $\Omega\left(\sqrt{\log \log n}\right)$ queries in the conditional sampling model. Interestingly, this demonstrates an intrinsic qualitative gap between identity and equivalence testing, absent in the standard sampling model (where both problems have sampling complexity $n^{\Theta(1)}$).

Turning to another question, we strengthen a result of Ron and Tsur [RT14] on support size estimation in the conditional sampling model, with an algorithm to approximate the support size of an *arbitrary* distribution. This result matches the previously known upper bound in the restricted case where the distribution is guaranteed to be uniform over a subset. Furthermore, we settle a related open problem of theirs, proving tight lower bounds on support size estimation with non-adaptive queries.

*EECS, MIT. Email: jayadev@csail.mit.edu. Research supported by grant from MITEI-Shell program.

†Columbia University. Email: ccanonne@cs.columbia.edu. Research supported by NSF CCF-1115703 and NSF CCF-1319788.

‡EECS, MIT. Email: g@csail.mit.edu.

ISSN 1433-8092

1 Introduction

"No, Virginia, there is no constant-query tester."

Understanding properties and characteristics of an unknown probability distribution is a fundamental problem in statistics, and one that has been thoroughly studied. However, it is only since the seminal work of Goldreich and Ron [GR00] and Batu et al. [BFR+00] that the problem has been studied through the lens of theoretical computer science, more particularly in the setting of *property testing*.

Over the next decade, a flurry of subsequent work explored and delved into this new area, resulting in a better and often complete understanding of a number of questions in distributional property testing (see e.g. [GR00, BFF+01, BKR04, Pan08, RS09, ADJ+11, BFRV11, Rub12, ILR12, ADJ+12, CDVV14, VV14] and references therein). In many cases, these culminated in provably sample-optimal algorithms. However, the standard setting of distribution testing, where one only obtains independent samples from an unknown distribution, does not encompass *all* scenarios one may encounter. In recent years, stronger models have thus been proposed to capture more specific situations [GMV06, CFGM13, CRS12, LRR13, CR14]: for instance, in the *conditional oracle model* of [CFGM13, CRS12] (which will be the focus of our work, and whose formal definition can be found in Definition 2.1), the testers are allowed sampling access to conditional distributions induced on subsets of the domain. In particular, the hope is that allowing algorithms to have stronger interactions with the unknown underlying distributions might reduce the number of samples needed to accomplish the same task, thereby sidestepping the strong lower bounds that hold in the standard sampling model.

1.1 Background and previous work

We focus in this paper on proving lower bounds for testing two extremely natural properties of distributions, namely *equivalence testing* and *support size estimation*. Along the way, we use some of the techniques we develop to obtain an upper bound on the query complexity of the latter. We state below the informal definition of these two problems, along with closely related ones (uniformity and identity testing). Oracle access to a distribution D means access to samples generated independently from D.

Uniformity testing: given oracle access to D, decide whether $D = \mathcal{U}$ (the uniform distribution on $[n]$) or is far from it;

Identity testing: given oracle access to D and the full description of a fixed D^*, decide whether $D = D^*$ or is far from it;

Equivalence (closeness) testing: given independent oracle accesses to D_1, D_2 (both unknown), decide whether $D_1 = D_2$ or D_1, D_2 are far from each other.

Support size estimation: given oracle access to D, output an estimate of the size of the support[1] $\mathrm{supp}(D) = \{\, x : D(x) > 0 \,\}$, accurate within a multiplicative factor.

It is not hard to see that each of the first three problems generalize the previous, and is therefore at least as hard. Each of these tasks is known to require sample complexity $n^{\Omega(1)}$ in the standard

[1] For this problem, it is typically assumed that all points in the support have probability mass at least $\Omega(1)/n$, as without such guarantee it becomes impossible to give any non-trivial estimate (consider for instance a distribution D such that $D(i) \propto 1/2^{in}$).

sampling model, where the algorithm can only ask for independent, identically distributed samples from the unknown distribution(s); yet, as [CFGM13, CRS12] show, their complexity decreases tremendously when one allows the stronger type of access to the distribution(s) provided by a conditional sampling oracle (COND). Indeed, for the problems of uniformity testing and identity testing, the sample complexity even becomes a constant, independent of the domain size n – provided the testing algorithm is allowed to be *adaptive*, i.e. when the next queries are dependent on the previous samples. As is common in the testing literature, the distance between distributions will be measured in terms of the total variation distance between them.

Testing uniformity and identity. Given a complete description of a distribution D^* over $[n]$, a parameter $\varepsilon > 0$, and oracle access to a distribution D, identity testing asks to distinguish the case $D_1 = D^*$ from where their total variation distance $\mathrm{d}_{\mathrm{TV}}(D, D^*)$ is at least ε. This is a generalization of uniformity testing, where D^* is taken to be the uniform distribution over $[n]$. The complexity of these tasks is well-understood in the sampling model; in particular, it is known that for both uniformity and identity testing $\Theta(\sqrt{n}/\varepsilon^2)$ samples are necessary and sufficient (see [GR00, BFR+10, Pan08, VV14] for the tight bounds on these problems).

The uniformity testing problem emphasizes the additional flexibility granted by conditional sampling: as [CRS12] showed, in this setting only $\tilde{O}(1/\varepsilon^2)$ adaptive queries now suffice (and this is optimal, up to logarithmic factors). They further prove that identity testing has constant sample complexity as well, namely $\tilde{O}(1/\varepsilon^4)$. The power of the COND model is evident from the fact that a task requiring polynomially many samples in the standard model can now be achieved with a number of samples that is *independent of the domain size* n.

Testing equivalence. A natural generalization of these two testing problems is the question of *equivalence testing*, defined as follows. Given oracle access to two unknown distributions D_1 and D_2 over $[n]$ and a parameter $\varepsilon > 0$, equivalence testing asks to distinguish between the cases $D_1 = D_2$ and $\mathrm{d}_{\mathrm{TV}}(D_1, D_2) > \varepsilon$. This problem has been extensively studied over the past decade, and its sample complexity is now known to be $\Theta\Big(\max(n^{2/3}/\varepsilon^{4/3}, \sqrt{n}/\varepsilon^2)\Big)$ ([BFR+10, Val11, CDVV14]).

While Canonne et al. showed that equivalence testing is possible with only $\mathrm{poly}(\log n, 1/\varepsilon)$ queries in the COND model, no superconstant lower bound is known[2]. In spite of the exponential improvement over the $n^{\Omega(1)}$ samples needed in the standard sampling model, the state of affairs is therefore unsatisfying: given that both uniformity and identity testing admit constant-query testers, it is natural to wonder where equivalence testing lies. Specifically, does it have constant query complexity as well, or must the number of queries grow with the domain size? Resolving this question was explicitly posed in the Bertinoro Workshop on Sublinear Algorithms 2014 [Sublinear.info, Problem 66].

Support size estimation. Finally, the question of approximating the support size of a distribution has been considered in [VV11, VV10a], where it was shown that obtaining *additive* estimates requires sample complexity almost linear in n (Valiant and Valiant show that a $n/\log n$ dependence is both necessary and sufficient). Based on a standard birthday paradox argument, Goldreich and Ron's uniformity lower bound [GR00] also directly implies that getting a multiplicative estimate, up to a factor two, requires $\Omega(\sqrt{n})$ samples.

[2]It is worth noting that an $\Omega(\log^c n)$ lower bound was known for equivalence testing in a weaker version of the conditional oracle, PAIRCOND (where the tester's queries are restricted to being either $[n]$ or subsets of size 2 [CRS12]).

To the best of our knowledge, the question of getting a multiplicative-factor estimate of the support size of a distribution given COND access to it has not been previously considered. However, we observe that Ron and Tsur [RT14] do study a related problem for which they provide strong upper and lower bounds, in a slightly different model: estimating the size of a "hidden subset" given (a variant of) conditional sampling access. We note that this problem differs from ours in two ways: first, phrased in terms of distribution testing, their algorithms are promised that the unknown distribution be *uniform* on its support, while in our setting it could be arbitrary. Secondly, whenever querying a subset which does not intersect the support of the distribution, their algorithm is notified it "missed" the support, while ours still receives a uniform sample from the query set.

1.2 Our results

In this work, we make significant progress in both problems introduced in the previous section, yielding a better understanding of the intrinsic query complexity of each of them. We prove three results pertaining to the sample complexity of support size estimation and equivalence testing in the COND framework.

Our main result considers the sample complexity of testing equivalence with adaptive queries under the COND model, resolving by the negative the question of whether constant-query complexity was achievable [Sublinear.info, Problem 66]. More precisely, we prove the following theorem:

Theorem 1.1 (Testing Equivalence)**.** *Any* adaptive *algorithm which, given* COND *access to unknown distributions D_1, D_2 on $[n]$, distinguishes with probability at least $2/3$ between (a) $D_1 = D_2$ and (b) $\mathrm{d}_{\mathrm{TV}}(D_1, D_2) \geq \frac{1}{4}$, must have query complexity $\Omega(\sqrt{\log \log n})$.*

Recalling that the related task of identity testing *can* be performed with a constant number of queries in the conditional sampling model, this demonstrates an interesting chasm and intrinsic difference between the two problems. Moreover, the result above can be interpreted as showing a fundamental distinction from the usual sampling model, where both identity and equivalence testing have polynomial sample complexity.

Next, we obtain two new results on the problem of support size estimation, providing both upper and lower bounds of a slightly different flavor. The first of these gives a $\tilde{O}(\log \log n)$ upper bound on constant-factor support size estimation, showing that this problem becomes *double exponentially easier* when conditional queries are allowed. Note that this is in some sense similar to the (incomparable) result of [RT14], which give an instance-optimal $\tilde{O}(\log \log \omega)$-query algorithm for their more restricted setting.

Theorem 1.2 (Support Size Estimation)**.** *There exists an (adaptive) algorithm which, given* COND *access to an unknown distribution D on $[n]$ which has minimum non-zero probability $1/n$, makes $\tilde{O}(\log \log n)$ queries to the oracle and outputs a value $\tilde{\omega}$ such that the following holds. With probability at least $2/3$, $\tilde{\omega} \in [\frac{1}{2} \cdot \omega, 2 \cdot \omega]$, where $\omega = |\mathrm{supp}(D)|$.*

The second gives an (almost) logarithmic lower bound on *non-adaptive* support size estimation, even for the weaker question of obtaining a $(\log n)$-factor approximation. (We also note that a straightforward modification of our previous upper bound shows this result to be optimal, up to $\log \log n$ factors in the query complexity.) In the course of proving this result, we also provide a nearly-tight answer to one of the questions left open in [RT14].

Theorem 1.3 (Non-Adaptive Support Size Estimation)**.** *Any non-adaptive algorithm which, given* COND *access to an unknown distribution D on $[n]$, estimates the size of its support up to a factor $\log n$ must have query complexity $\Omega\left(\frac{\log n}{\log\log n}\right)$.*

1.3 Techniques and proof ideas

We now provide an overview of the techniques and arguments used to prove our results. We first describe the results on support size estimation, and then describe the more involved arguments for our main theorem.

Upper bound on support size estimation. Our algorithm for estimating the support size to a constant factor (Theorem 1.2) is simple in spirit, and follows a guess-and-check strategy. In more detail, it first obtains a "reference point" *outside* the support, to check whether subsequent samples it may consider belong to the support. Then, it attempts to find a *rough upper bound* on the size of the support, of the form 2^{2^j} (so that only $\log\log n$ many options have to be considered); by using its reference point to check if a uniform random subset of this size contains, as it should, at least one point from the support. Once such an upper bound has been obtained using this double-exponential strategy, a refined bound is then obtained via a binary search on the new range of values for the exponent, $\{2^{j-1},\dots,2^j\}$. Not surprisingly, our algorithm draws on similar ideas as in [RT14, Sto85], with some additional machinery to supplement the differences in the models. Interestingly, as a side-effect, this upper bound shows our analysis of Theorem 1.1 to be tight up to a quadratic dependence. Indeed, the lower bound construction we consider (see Section 5.1) can be easily "defeated" if an estimate of the support size is known, and therefore cannot yield better than a $\Omega(\log\log n)$ lower bound. Similarly, this also shows that the adaptive lower bound for support size estimation of Chakraborty et al. [CFGM13] is also tight up to a quadratic dependence.

Lower bound on non-adaptive support size estimation. Turning to the corresponding (non-adaptive) lower bound of Theorem 1.3, we define two families of distributions $\mathcal{D}_1$ and $\mathcal{D}_2$. All distributions in $\mathcal{D}_1$ have the same support size, and all distributions in $\mathcal{D}_2$ have the same support size. Moreover, these sizes are a (multiplicative) factor at least $\log n$ from each other. Yet, we argue that no non-adaptive *deterministic* tester making too few queries can distinguish between a random distribution from $\mathcal{D}_1$ and one from $\mathcal{D}_2$, as the tuple of samples it will obtain in both cases is almost identically distributed (where the randomness is over the choice of the distribution itself). To show this last point, we analyze separately the case of "small" queries (conditioning on sets which turn out to be much smaller than the actual support size, and thus with high probability will not even intersect it) and the "big" ones (where the query set A is so big in front of the support size S that a uniform sample from $A \cap S$ is essentially indistinguishable from a uniform sample from A). Bounding the probability of the "intermediate" case (where the support size ends up being slightly too close to the size a query set) finishes the case distinction. We conclude the proof by invoking Yao's Principle, carrying the lower bound back to the setting of non-adaptive *randomized* testers.

Lower bound on adaptive equivalence testing. Finally, in order to prove our main $\omega(1)$ lower bound on the query complexity of testing equivalence in the conditional sampling model, we have to deal with one main conceptual issue: *adaptivity*. While the standard sampling model does

not, by definition, allow any choice on what the next query to the oracle should be, this is no longer the case for COND algorithms. Quantifying the power that this grants an algorithm makes things much more difficult. To handle this point, we follow the approach of Chakraborty et al. [CFGM13] and focus on a restricted class of algorithms they introduce, called "core adaptive testers" (see Section 2.2 for a formal definition). [CFGM13] show that this class of testers is equivalent to general algorithms for the purpose of testing a broad class of properties, namely those which are invariant to any permutation of the domain. Using this characterization, it suffices for us to show that none of these structurally much simpler core testers can distinguish whether they are given conditional access to (a) a pair of random identical distributions (D_1, D_1), or (b) two distributions (D_1, D_2) drawn according to a similar process, but which are far apart.

At a high level, our lower bound works by designing instances where the property can be tested if and only if the support size is known to the algorithm. Our construction randomizes the support size by embedding the instance into a polynomially larger domain. Since the algorithm is only allowed a small number of queries, Yao's Principle allows us to argue that, with high probability, a deterministic algorithm is unable to "guess" the support size. This separates queries into several cases. First, in a sense we make precise, it is somehow "predictable" whether or not a query will return an element we have previously observed. If we do, it is similarly predictable *which* element the query will return. On the other hand, if we observe a fresh element, the query set is either "too small" or "too large." In the former case, the query will entirely miss the support, and the sampling process is identical for both types of instance. In the latter case, the query will hit a large portion of the support, and the information gleamed from a single sample is minimal.

At a lower level, this process itself is reminiscent of the lower bound construction of [CRS12] on testing identity (with a PAIRCOND oracle), with one pivotal twist. As in [CRS12], both D_1 and D_2 are uniform within each of $\omega(1)$ "buckets" whose size grows exponentially and are grouped into "bucket-pairs". Then, D_2 is obtained from D_1 by internally redistributing the probability mass of each pair of buckets, so that the total mass of each pair is preserved but each particular bucket has mass going up or down by a constant factor (see Section 5.1 for details of the construction). However, we now add a final step, where in both D_1 and D_2 the resulting distribution's support is *scaled by a random factor*, effectively reducing it to a (randomly) negligible fraction of the domain. Intuitively, this last modification has the role of "blinding" the testing algorithm: we argue that unless its queries are on sets whose size somehow match (in a sense formalized in Section 5.2) this random size of the support, the sequences of samples it will obtain under D_1 and D_2 are almost identically distributed. (We observe that the above discussion crucially hides many significant aspects and technical difficulties which we address in Section 5.)

Organization. The rest of the paper describes details and proofs of the results mentioned in the above discussion. In Section 2, we introduce the necessary definitions and some of the tools we shall use. Section 3 and Section 4 cover respectively our upper and lower bounds for support size estimation, while Section 5 covers our main result, Theorem 1.1. The reader may independently read the corresponding sections at their discretion.

2 Preliminaries

2.1 Notation and sampling models

All throughout this paper, we denote by $[n]$ the set $\{1, \ldots, n\}$, and by log the logarithm in base 2. A probability distribution over a (countable) domain $[n]$ is a non-negative function $D\colon [n] \to [0,1]$ such that $\sum_{x\in[n]} D(x) = 1$. We denote by $\mathcal{U}(S)$ the uniform distribution on a set S. Given a distribution D over $[n]$ and a set $S \subseteq [n]$, we write $D(S)$ for the total probability mass $\sum_{x\in S} D(x)$ assigned to S by D. Finally, for $S \subseteq [n]$ such that $D(S) > 0$, we denote by D_S the conditional distribution of D restricted to S, that is $D_S(x) = \frac{D(x)}{D(S)}$ for $x \in S$ and $D_S(x) = 0$ otherwise.

As is usual in distribution testing, in this work the distance between two distributions D_1, D_2 on $[n]$ will be the *total variation distance*:

$$\mathrm{d_{TV}}(D_1, D_2) \stackrel{\text{def}}{=} \frac{1}{2}\|D_1 - D_2\|_1 = \frac{1}{2}\sum_{x\in[n]} |D_1(i) - D_2(i)| = \max_{S\subseteq[n]} (D_1(S) - D_2(S)) \tag{1}$$

which takes value in $[0,1]$.

In this work, we focus on the setting of *conditional access* to the distribution, as introduced and studied by [CFGM13, CRS12]. We reproduce below the corresponding definition of a conditional oracle, henceforth referred to as COND:

Definition 2.1 (Conditional access model)**.** Fix a distribution D over $[n]$. A COND *oracle for* D, denoted COND_D, is defined as follows: the oracle takes as input a *query set* $S \subseteq [n]$, chosen by the algorithm, that has $D(S) > 0$. The oracle returns an element $i \in S$, where the probability that element i is returned is $D_S(i) = D(i)/D(S)$, independently of all previous calls to the oracle.

Note that as described above the behavior of $\mathsf{COND}_D(S)$ is undefined if $D(S) = 0$, i.e., the set S has zero probability under D. Various definitional choices could be made to deal with this. These choice do not do not make significant difference in most situations, as most (adaptive) algorithms can always include in their next queries a sample previously obtained; this is how our upper bounds will be obtained, while our lower bounds can be thought of as putting exponentially small probability mass of elements outside the support. For this reason, and for convenience, we shall hereafter assume, following Chakraborty et al., that the oracle returns in this case a sample uniformly distributed in S.

Finally, recall that a *property* $\mathcal{P}$ of distributions over $[n]$ is a set consisting of all distributions that have the property. The distance from D to a property $\mathcal{P}$, denoted $\mathrm{d_{TV}}(D, \mathcal{P})$, is then defined as $\inf_{D'\in\mathcal{P}} \mathrm{d_{TV}}(D, \mathcal{P})$. We use the standard definition of testing algorithms for properties of distributions over $[n]$, tailored for the setting of conditional access to an unknown distribution:

Definition 2.2 (Property tester)**.** Let $\mathcal{P}$ be a property of distributions over $[n]$. A *t-query* COND *testing algorithm for* $\mathcal{P}$ is a randomized algorithm $\mathcal{T}$ which takes as input n, $\varepsilon \in (0,1]$, as well as access to COND_D. After making at most $t(\varepsilon, n)$ calls to the oracle, $\mathcal{T}$ either outputs ACCEPT or REJECT, such that the following holds:

[2]Recall that a non-adaptive tester is an algorithm whose queries do not depend on the answers obtained from previous ones, but only on its internal randomness. Equivalently, it is a tester that can commit "upfront" to all the queries it will make to the oracle.

- if $D \in \mathcal{P}$, $\mathcal{T}$ outputs ACCEPT with probability at least 2/3;
- if $\mathrm{d_{TV}}(D, \mathcal{P}) \geq \varepsilon$, $\mathcal{T}$ outputs REJECT with probability at least 2/3.

We observe that the above definitions can be straightforwardly extended to the more general setting of *pairs* of distributions, where given independent access to two oracles COND_{D_1}, COND_{D_2} the goal is to test whether (D_1, D_2) satisfies a property (now a set of pairs of distributions). This will be the case in Section 5, where we will consider equivalence testing, that is the property $\mathcal{P}_{\mathrm{eq}} = \{\ (D_1, D_2) :\ D_1 = D_2\ \}$.

2.2 Adaptive Core Testers

In order to deal with adaptivity in our lower bounds, we will use ideas introduced by Chakraborty et al. in [CFGM13]. These ideas, for the case of *label-invariant* properties[3] allow one to narrow down the range of possible testers and focus on a restricted class of such algorithms called *adaptive core testers*. These core testers do not have access to the full information of the samples they draw, but instead only get to see the relations (inclusions, equalities) between the queries they make and the samples they get. Yet, Chakraborty et al. show that any tester for a label-invariant property can be converted into a core tester with same query complexity; thus, it is enough to prove lower bounds against this – seemingly – weaker class of algorithms.

We here rephrase the definitions of a core tester and the view they have of the interaction with the oracle (the *configuration* of the samples), tailored to our setting.

Definition 2.3 (Atoms and partitions)**.** Given a family $\mathcal{A} = (A_1, \ldots, A_t) \subseteq [n]^t$, the *atoms* generated by $\mathcal{A}$ are the (at most) 2^t distinct sets of the form $\bigcap_{r=1}^{t} C_r$, where $C_r \in \{A_r, [n] \setminus A_r\}$. The family of all such atoms, denoted $\mathrm{At}(\mathcal{A})$, is the *partition* generated by $\mathcal{A}$.

This definition essentially captures "all sets (besides the A_i's) about which something can be learnt from querying the oracle on the sets of $\mathcal{A}$". Now, given such a sequence of queries $\mathcal{A} = (A_1, \ldots, A_t)$ and pairs of samples $\mathbf{s} = ((s_1^{(1)}, s_1^{(2)}), \ldots, (s_t^{(1)}, s_t^{(2)})) \in A_1^2 \times \cdots \times A_t^2$, we would like to summarize "all the label-invariant information available to an algorithm that obtains $((s_1^{(1)}, s_1^{(2)}), \ldots, (s_t^{(1)}, s_t^{(2)}))$ upon querying $A_1, \ldots, A_t$ for D_1 and D_2". This calls for the following definition:

Definition 2.4 (t-configuration)**.** Given $\mathcal{A} = (A_1, \ldots, A_t)$ and $\mathbf{s} = ((s_j^{(1)}, s_j^{(2)}))_{1 \leq j \leq t}$ as above, the *t-configuration of* $\mathbf{s}$ consists of the $6t^2$ bits indicating, for all $1 \leq i, j \leq t$, whether

- $s_i^{(k)} = s_j^{(\ell)}$, for $k, \ell \in \{1, 2\}$; and (relations between samples)
- $s_i^{(k)} \in A_j$, for $k \in \{1, 2\}$. (relations between samples and query sets)

In other terms, it summarizes which is the unique atom $S_i \in \mathrm{At}(\mathcal{A})$ that contains $s_i^{(k)}$, and what collisions between samples have been observed.

As aforementioned, the key idea is to argue that, without loss of generality, one can restrict one's attention to algorithms that only have access to t-configurations, and generate their queries in a specific (albeit adaptive) fashion:

[3] Recall that a property is label-invariant (or *symmetric*) if it is closed under relabeling of the elements of the support. More precisely, a property of distributions (resp. pairs of distributions) $\mathcal{P}$ is label-invariant if for any distribution $D \in \mathcal{P}$ (resp. $(D_1, D_2) \in \mathcal{P}$) and permutation σ of $[n]$, one has $D \circ \sigma \in \mathcal{P}$ (resp. $(D_1 \circ \sigma, D_2 \circ \sigma) \in \mathcal{P}$).

Definition 2.5 (Core adaptive tester)**.** A *core adaptive distribution tester* for pairs of distributions is an algorithm $\mathcal{T}$ that acts as follows.

- In the i-th phase, based only on its own internal randomness and the configuration of the previous queries $A_1, \ldots, A_{i-1}$ and samples obtained $(s_1^{(1)}, s_1^{(2)}), \ldots, (s_{i-1}^{(1)}, s_{i-1}^{(2)})$ – whose labels it does not actually know, $\mathcal{T}$ provides:
  - a number k_i^A for each $A \in \mathrm{At}(A_1, \ldots, A_{i-1})$, between 0 and $\left|A \setminus \{s_j^{(1)}, s_j^{(2)}\}_{1 \le j \le i-1}\right|$ ("how many *fresh, not-already-seen* elements of each particular atom A should be included in the next query")
  - sets $K_i^{(1)}, K_i^{(2)} \subseteq \{1, \ldots, i-1\}$ ("which of the samples $s_1^{(k)}, \ldots, s(k)_{i-1}$ (whose label is unknown to the tester, but referred to by the index of the query it got them) will be included in the next query").
- based on these specifications, the next query A_i is drawn (but not revealed to $\mathcal{T}$) by
  - drawing uniformly at random a set Λ_i in

$$\left\{ \Lambda \subseteq [n] \setminus \{s_j^{(1)}, s_j^{(2)}\}_{1 \le j \le i-1} \,:\, \forall A \in \mathrm{At}(A_1, \ldots, A_{i-1}),\, |\Lambda \cap A| = k_i^A \right\} .$$

    That is, among all sets, containing only "fresh elements", whose intersection with each atom contains as many elements as $\mathcal{T}$ requires.
  - adding the selected previous samples to this set:

$$\Gamma_i \stackrel{\text{def}}{=} \left\{ s_j^{(1)} \,:\, j \in K_i^{(1)} \right\} \cup \left\{ s_j^{(2)} \,:\, j \in K_i^{(2)} \right\} ;$$

$$A_i \stackrel{\text{def}}{=} \Lambda_i \cup \Gamma_i .$$

    This results in a set A_i, not fully known to $\mathcal{T}$ besides the samples it already got and decided to query again; in which the *labels* of the fresh elements are unknown, but the *proportions* of elements belonging to each atom are known.
- samples $s_i^{(1)} \sim (D_1)_{A_i}$ and $s_i^{(2)} \sim (D_2)_{A_i}$ are drawn (but not disclosed to $\mathcal{T}$). This defines the i-configuration of $A_1, \ldots, A_i$ and $(s_1^{(1)}, s_1^{(2)}), \ldots, (s_i^{(1)}, s_i^{(2)})$, which is revealed to $\mathcal{T}$. Put differently, the algorithm only learns (i) to which of the A_ℓ's the new sample belongs, and (ii) if it is one of the previous samples, in which stage(s) and for which of D_1, D_2 it has already seen it.

After $t = t(\varepsilon, n)$ such stages, $\mathcal{T}$ outputs either ACCEPT or REJECT, based only on the configuration of $A_1, \ldots, A_t$ and $(s_1^{(1)}, s_1^{(2)}), \ldots, (s_t^{(1)}, s_t^{(2)})$ (which is all the information it ever had access to).

Note that in particular, $\mathcal{T}$ does not know the labels of samples it got, nor the actual queries it makes: it knows all about their sizes and sizes of their intersections, but not the actual "identity" of the elements they contain.

2.3 On the use of Yao's Principle in our lower bounds

We recall Yao's Principle (e.g., see Chapter 2.2 of [MR95]), a technique which is ubiquitous in the analysis of randomized algorithms. Consider a set S of instances of some problem: what this principle states is that the worst-case expected cost of a randomized algorithm on instances in S

is lowerbounded by the expected cost of the best deterministic algorithm on an instance drawn randomly from S.

As an example, we apply it in a standard way in Section 4: instead of considering a randomized algorithm working on a fixed instance, we instead analyze a *deterministic* algorithm working on a *random* instance. (We note that, importantly, the randomness in the samples returned by the COND oracle is "external" to this argument, and these samples behave identically in an application of Yao's Principle.)

On the other hand, our application in Section 5 is slightly different, due to our use of adaptive core testers. Once again, we focus on deterministic algorithms working on random instances, and the randomness in the samples is external and therefore unaffected by Yao's Principle. However, we stress that the randomness in the choice of the set Λ_i is also external to the argument, and therefore unaffected – similar to the randomness in the samples, the algorithm has no control here. Another way of thinking about this randomness is via another step in the distribution over instances: after an instance (which is a pair of distributions) is randomly chosen, we permute the labels on the elements of the distribution's domain uniformly at random. We note that since the property in question is label-invariant, this does not affect its value. We can then use the model as stated in Section 2.2 for ease of analysis, observing that this can be considered an application of the principle of deferred decisions (as in Chapter 3.5 of [MR95]).

3 An Upper Bound for Support Size Estimation

In this section, we prove our upper bound for constant-factor support size estimation, reproduced below:

Theorem 1.2 (Support Size Estimation)**.** *There exists an (adaptive) algorithm which, given* COND *access to an unknown distribution D on $[n]$ which has minimum non-zero probability $1/n$, makes $\tilde{O}(\log \log n)$ queries to the oracle and outputs a value $\tilde{\omega}$ such that the following holds. With probability at least $2/3$, $\tilde{\omega} \in [\frac{1}{2} \cdot \omega, 2 \cdot \omega]$, where $\omega = |\mathrm{supp}(D)|$.*

Proof. At a high-level, what our algorithm does is repeatedly try to guess a good candidate $\tilde{\omega}$, by performing first a doubly exponential search, then a binary search among the $\log n$ possible values $\{n, \frac{n}{2}, \frac{n}{4}, \ldots, 1\}$. For any fixed value $\tilde{\omega}$, it then queries a random set $S = S_{\tilde{\omega}}$ of size roughly $n/\tilde{\omega}$; if $\tilde{\omega}$ is a good approximation of the support size, this set should with constant probability intersect the support on constantly many points. However, if $\tilde{\omega} \ll \omega$, then the set will not hit the support at all; thus, the key is to efficiently determine in which of the two cases we are. To do so, and sweeping for now a lot of details under the rug, we rely on a simple observation: if $S \cap \mathrm{supp}(D) = \Theta(1)$, then by querying D_S we get a point that will not be "comparable" (having non-zero probability) to a reference point r from outside the support. However, if $S \cap \mathrm{supp}(D) = \emptyset$, then the oracle will return a uniformly random point x from S; and the distribution on $\{x, r\}$ will be exactly uniform (both have zero mass under D). This intuition, once formalized, will be at the heart of our algorithm.

We will use as subroutines the following results of Canonne et al.; the first one provides a way to "compare" the probability mass of disjoint subsets of elements:

Lemma 3.1 ([CRS12, Lemma 2])**.** *Given as input two disjoint subsets of points $X, Y \subseteq [n]$ together with parameters $\eta \in (0, 1]$, $K \geq 1$, and $\delta \in (0, 1/2]$, as well as* COND *query access to a distribution D on $[n]$, there exists a procedure* Compare *that either outputs a value $\rho > 0$ or outputs* High *or* Low*, and satisfies the following:*

(i) *If* $D(X)/K \le D(Y) \le K \cdot D(X)$ *then with probability at least* $1-\delta$ *the procedure outputs a value* $\rho \in [1-\eta, 1+\eta]D(Y)/D(X)$*;*

(ii) *If* $D(Y) > K \cdot D(X)$ *then with probability at least* $1-\delta$ *the procedure outputs either* High *or a value* $\rho \in [1-\eta, 1+\eta]D(Y)/D(X)$*;*

(iii) *If* $D(Y) < D(X)/K$ *then with probability at least* $1-\delta$ *the procedure outputs either* Low *or a value* $\rho \in [1-\eta, 1+\eta]D(Y)/D(X)$*.*

The procedure performs $O\left(\frac{K \log(1/\delta)}{\eta^2}\right)$ COND *queries on the set* $X \cup Y$*.*

The second subroutine we shall rely on is an algorithm that computes an additive estimate of the distance between a distribution D and the uniform distribution $\mathcal{U}_{[n]}$, performing only a constant number of queries:

Theorem 3.2 ([CRS12, Theorem 21])**.** *Given as input parameters* $\varepsilon, \delta \in (0, 1/2]$*, as well as* COND *query access to a distribution* D *on* $[n]$*, there exists an algorithm that performs* $\tilde{O}\left(\frac{1}{\varepsilon^{20}} \log \frac{1}{\delta}\right)$ *queries and returns an estimate* $\hat{d}$ *such that, with probability at least* $1-\delta$*,* $|\hat{d} - \mathrm{d_{TV}}(D,\mathcal{U})| \le \varepsilon$*.*

With this in hand, we are ready to analyze the behavior of our algorithm, described in Algorithm 1.

Algorithm 1 EstimateSupport$_D$

1: Get an estimate $\hat{d}$ of $\mathrm{d_{TV}}(D,\mathcal{U})$, calling the algorithm from Theorem 3.2 with ε set to $1/4$ and δ to $1/10$. ▷ If $\omega < \frac{n}{2}$, then $\mathrm{d_{TV}}(D,\mathcal{U}) \ge 1/2$.
2: **if** $\hat{d} \le 3/4$ **then return** $\tilde{\omega} \leftarrow n$
3: **end if**
4: Call GetNonSupport$_D(\frac{n}{2}, \frac{1}{10})$ to obtain a non-support reference point r.
5: **for** j **from** 0 **to** $\log\log n - 1$ **do**
6: Set $\tilde{\omega} \leftarrow 2^{2^j}$.
7: Call IsAtMostSupportSize$_D(\tilde{\omega}, r, \frac{1}{100\cdot(j+1)^2})$ to check if $\tilde{\omega}$ is an upper bound on ω.
8: **if** the call returned no **then**
9: Perform a binary search on $\{2^{j-1}, \ldots, 2^j\}$ to find i^*, the smallest i such that IsAtMostSupportSize$_D(2^i, r, \frac{1}{10(j+1)})$ returns no.
10: **return** $\tilde{\omega} \leftarrow 2^{i^*-1}$.
11: **end if**
12: **end for**

Correctness. We start by arguing correctness of the two subroutines, IsAtMostSupportSize and GetNonSupport. The latter is straightforward, as the probability that any fixed one of the uniformly drawn points hits the support is at most $(1 - \frac{m}{n})$; and that calling COND$_D$ on two points $\{x, y\}$ will return each of the two points with same probability if none belongs to $\mathrm{supp}(D)$.
As for IsAtMostSupportSize, we consider two cases.

- If $\sigma \le \omega$, then the expected intersection $\mathbb{E}|S \cap \mathrm{supp}(D)| = \frac{3\omega}{4\sigma}$ is at least $3/4$. But then, the probability to have $S \cap \mathrm{supp}(D) = \emptyset$ is $(1 - \frac{3}{4\sigma})^\omega \le (1 - \frac{3}{4\omega})^\omega \le e^{-3/4}$.
- If $\sigma > 2\omega$, then the expected intersection is less than $3/8$; by Markov's inequality, we get that $\Pr[|S \cap \mathrm{supp}(D)| \ge 1] \le 3/8 \le e^{-3/4}$.

Algorithm 2 GetNonSupport$_D(m, \delta)$

Require: COND access to D; upper bound m on $\mathrm{supp}(D)$, probability of failure δ
Ensure: Returns $r \in [n]$ such that, with probability at least $1-\delta$, $r \notin \mathrm{supp}(D)$

Set $k \stackrel{\text{def}}{=} \left\lceil \log \frac{2}{\delta} \log^{-1} \frac{1}{1-m/n} \right\rceil$.
Draw independently k points $s_1, \ldots, s_k \sim \mathcal{U}_{[n]}$
for all $1 \le i < j \le k$ **do**
    Call Compare$(\{s_i\}, \{s_j\}, \eta = \frac{1}{2}, K = 2, \frac{\delta}{2k^2})$ to get either a constant-factor approximation ρ of $D(s_j)/D(s_i)$ or High (resp. Low).
    if Compare returned High or a value ρ **then**
        Record $s_i \preceq s_j$
    else
        Record $s_j \prec s_j$
    end if
end for
return $\arg\min_{\preceq}\{s_1, \ldots, s_k\}$ ▷ Return (any) minimal element for $\preceq$.

Algorithm 3 IsAtMostSupportSize$_D(\sigma, r, \delta)$

Require: COND access to D; size σ, non-support element r, probability of failure δ
Ensure: Returns, with probability at least $1-\delta$, yes if $\sigma \le |\mathrm{supp}(D)|$ and no if $\sigma > 2|\mathrm{supp}(D)|$.

for $m = O\left(\log \frac{1}{\delta}\right)$ times **do**
    Draw a subset $S \subseteq [n]$ by including independently each $x \in [n]$ with probability $3/(4\sigma)$.
    Draw $x \sim D_S$.
    Call Compare$(\{x\}, \{r\}, \eta = \frac{1}{2}, K = 1, \frac{1}{100})$ ▷ Low if $S \cap \mathrm{supp}(D) \neq \emptyset$; $\rho \in [\frac{1}{2}, 2)$ o.w.
    Record yes if Compare returned Low, no otherwise.
end for
return yes at least $\frac{m}{2}$ "yes"'s were recorded, no otherwise.

Therefore, by a union bound with the call to Compare each of the inner loops outputs records the correct answer (either yes or no) except with probability at most $1/100 + e^{-3/4} < 49/100$. Boosting the probability of success to $1-\delta$ can then easily be done by repeating $O(\log(1/\delta))$ times and taking the majority vote.

Conditioning from now on each of the calls to the subroutines being correct (which overall happens except with probability at most $1/10 + 1/10 + \sum_{j=1}^{\infty} 1/(100j^2) + 1/10 < 1/3$ by a union bound), we show that the output $\tilde{\omega}$ of EstimateSupport is indeed within a factor 2 of ω.

- If $\hat{d} \leq 3/4$ in the first test, then we have $\mathrm{d_{TV}}(D, \mathcal{U}) \leq 1/2$; this is only possible if the support has size at least $\omega \geq n/2$, and thus the estimate we output is correct.
- Conversely, if $\hat{d} > 3/4$ then we claim that $\omega < n/2$. To see why, note that in this case $\mathrm{d_{TV}}(D, \mathcal{U}) > 1/2$; but by definition of the promise problem $\mathrm{d_{TV}}(D, \mathcal{U}) = (n-\omega) \cdot \frac{1}{n} = 1 - \frac{\omega}{n}$.

Therefore, if we reach Step 4 then $\frac{n}{2}$ is indeed an upper bound on ω, and GetNonSupport will return a point $r \notin \mathrm{supp}(D)$ as expected. The analysis of the rest of the algorithm is straightforward: from the guarantee of IsAtMostSupportSize, the binary search will be performed for the first index j such that $\omega \in [2^{2^{j-1}}, 2^{2^j}]$; and will be on a set of 2^{j-1} values. Similarly, for the value i^* eventually obtained, it must be the case that $2^{i^*} > \omega$ (by contrapositive, as no was returned by the subroutine) but $2^{i^*-1} \leq 2\omega$ (again, as the subroutine returned yes). But then, $\tilde{\omega} = 2^{i^*-1} \in (\omega/2, 2\omega]$ as claimed.

Query complexity. The query complexity of our algorithm originates from the following different steps:

- the call to the distance estimation routine from Theorem 3.2, which given our choice of ε, δ, makes $O(1)$ queries;
- the call to GetNonSupport, on Step 4, that from the choice of parameters (and the definition of Compare) also costs $O(1)$ queries;
- the (at most) $\log\log n$ calls to IsAtMostSupportSize on Step 7. Observing that the query complexity of IsAtMostSupportSize is (again, from the definition of Compare) only $O(\log(1/\delta))$, and from the choice of $\delta = \frac{1}{(j+1)^2}$ at the j-th iteration this step costs at most

$$\sum_{j=1}^{\log\log n} O\Big(\log(j^2)\Big) = \tilde{O}(\log\log n)$$

  queries.
- Similarly, Step 9 results in at most $j \leq \log\log n$ calls to IsAtMostSupportSize with δ set to $1/(10(j+1))$, again costing $O(j\log j) = \tilde{O}(\log\log n)$ queries.

Gathering all terms, the overall query complexity is $\tilde{O}(\log\log n)$, as claimed. □

Remark 3.3. We observe that a straightforward modification of the above analysis of the query complexity actually yields a stronger statement: that is, whenever the algorithm runs correctly (i.e., with probability at least 2/3) the number of queries performed is $\tilde{O}(\log\log\omega)$ (where as before $\omega \leq n$ is the unknown support size). Furthermore, our algorithm can easily be adapted (as in the case of [RT14]) to yield a $\tilde{O}(\log n)$ *non-adaptive* estimation algorithm, by replacing the two-stage approach (doubly exponential search followed by a binary one) by a one-stage one (single exponential

search, trying iteratively $\tilde{\omega} \in \{2, 4, 8, .., n\}$). Combined with Theorem 4.1, this essentially settles the complexity of this question for non-adaptive algorithms.

4 A Lower Bound for Non-Adaptive Support Size Estimation

In this section, we prove our lower bound for non-adaptive support size estimation, rephrased below:

Theorem 1.3 (Non-Adaptive Support Size Estimation)**.** *Any non-adaptive algorithm which, given* COND *access to an unknown distribution D on $[n]$, estimates the size of its support up to a factor $\log n$ must have query complexity $\Omega\left(\frac{\log n}{\log\log n}\right)$.*

This theorem directly follows from the slightly stronger result below, that shows that estimating the support size of a distribution to a *logarithmic* factor is hard for non-adaptive algorithms, even in the slightly stronger model of Ron and Tsur [RT14]. (Recall that in this model the algorithm is notified when it queries a set that does not intersect the support.)

Theorem 4.1. *There exists an absolute constant $c > 0$ such that the following holds. Any non-adaptive algorithm which given* COND *access to a uniform distribution over an unknown subset S of $[n]$ estimates $|S|$ up to a factor $\log n$ must make at least $c\frac{\log n}{\log\log n}$ queries in the worst case.*

We note that Theorem 4.1 provides a nearly tight answer to one of the questions left unanswered in [RT14], namely a lower bound on the number of samples for non-adaptive queries.

By Yao's Principle, we consider deterministic tests and study their performance over random distributions. Let $A_1, \ldots, A_t$ be any fixed query sets, with $t < c \cdot \frac{\log n}{\log\log n}$ for an absolute constant c to be determined later, and $a_i \stackrel{\text{def}}{=} |A_i|$ be the size of the i-th set. Finally, let $\mathcal{S} \stackrel{\text{def}}{=} \{n^{1/4}, 2n^{1/4}, 4n^{1/4}, \ldots, n^{3/4}\}$.

We note that given a constant factor approximation of the support size, a non-adaptive algorithm can verify its correctness – our lower bound comes from the fact that there are many well-separated choices for the support size, and we don't have enough queries to try them all. The following lemma formalizes the fact that an algorithm can't "guess" the support size, and all queries are either "too large" or "too small."

Lemma 4.2. *If $c < 1/500$, a number s drawn uniformly at random from $\mathcal{S}$ satisfies $\frac{a_i s}{n} \notin \left(\frac{1}{\log^6 n}, \log^6 n\right)$ for all $1 \le i \le t$ with probability at least $9/10$.*

Proof. Say a number a_i is *close* to a number s if $\max\left\{\frac{a_i s}{n}, \frac{n}{a_i s}\right\} < \log^6 n$. Any a is close to at most $12\log\log n$ elements of $\mathcal{S}$. Therefore, the union of all elements in $\mathcal{S}$ that are close to at least one of the a_i's comprises at most $12t\log\log n$ values. As the total size of $\mathcal{S}$ is $\frac{1}{2}\log n$, if

$$12t\log\log n < \frac{1}{10}\cdot\frac{1}{2}\log n$$

with probability at least $9/10$ a uniformly chosen element of $\mathcal{S}$ is not close to any a_i. □

We now condition on the event of the lemma having occurred (which happens with probability at least 9/10), i.e. that we picked an s satisfying the statement. Define a *transcript* to be the (random) sequence of samples $(s_1, \ldots, s_t)$ observed from the sequence of queries $(A_1, \ldots, A_t)$. We show that transcripts obtained from the distributions resulting from the following two processes have a *small* total variation distance:

- $\mathcal{D}_1$: Pick a random set of size s from $[n]$, and set U_1 to be the uniform distribution on this set.
- $\mathcal{D}_2$: Pick a random set of size $s \log n$ from $[n]$ and set U_1 is the uniform distribution on this set.

We show that it is not possible to distinguish $\mathcal{D}_1$ from $\mathcal{D}_2$ with probability 2/3, thereby proving Theorem 4.1.

Without loss of generality assume that $a_1 \geq a_2 \ldots \geq a_t$.

First, we analyze the queries which are "too small." Let t' be the largest number such that $\frac{a_{t'} s}{n} > \log^6 n$ (or $t+1$ if there is none). Then, by our assumption,

$$\frac{a_j s}{n} < \frac{1}{\log^6 n}$$

for all $j > t'$. We first show that with high probability, there is "no information" from the sets A_j's for $j > t'$ when a distribution is picked randomly from $\mathcal{D}_1$ or $\mathcal{D}_2$, formalized in the following lemma:

Lemma 4.3. *With probability at least $1 - O\left(\frac{1}{\log^3 n}\right)$, a distribution drawn at random from $\mathcal{D}_1$ or $\mathcal{D}_2$ does not intersect any A_j, for $j > t'$.*

Proof. The expected size of intersection of a random set of size s with a set of size a_j is $a_j s/n$. By our assumption, this expected value is at most $1/\log^6 n$ for a distribution from $\mathcal{D}_1$ (of size s) and at most $1/\log^5 n$ for a distribution from $\mathcal{D}_1$ (of size $s \log n$). By Markov's inequality, the probability that S intersects A_j is at most $1/\log^5 n$.
By a union bound, it follows that

$$\Pr\left[\, \exists j > t',\ S \cap A_j \neq \emptyset \,\right] \leq \frac{t - t'}{\log^5 n}.$$

Recalling there are at most $t < c \log n / \log \log n$ such sets, with probability at least $1 - O(\log^{-3} n)$ none of them intersects any A_j. □

Therefore, with high probability (over the choice of s), we observe that the output of conditional sampling over these sets is uniform, as they contain no element in the support; and this happens for both $\mathcal{D}_1$ and $\mathcal{D}_2$.

Now, we analyze the queries which are "too large." We show that the distribution of samples from $A_1, \ldots, A_{t'}$ according to $\mathcal{D}_1$ and $\mathcal{D}_2$ have a total variation distance $o(1)$; combining this with Lemma 4.3 will yield the theorem. This is done by showing that the distribution of samples $s_1, \ldots, s_{t'}$ from sets $A_1, \ldots, A_{t'}$ respectively, are close in distribution to the process that generates samples uniformly at random from each A_i. The rest follows by the triangle inequality.

Lemma 4.4. *Suppose $A_1, \ldots, A_{t'}$, and s are such that $\frac{|A_i| s}{n} > \log^3 n$. Then, the distributions over tuples $(s_1, \ldots, s_{t'})$ generated according to the following two processes have total variation $o(1)$.*

P1: *Pick S uniformly from all subsets of $[n]$ of size s; then, for each $i \in [t']$, draw s_i uniformly from $A_i \cap S$ (or set $s_i = 1$ if $A_i \cap S = \emptyset$).*

P2: *For $i \in [t']$, draw s_i uniformly from A_i.*

Remark 4.5. The distributions from $\mathcal{D}_1$ satisfy the condition in the statement of Lemma 4.4 by assumption (for s, and t' as before), and it holds a fortiori for a distribution from $\mathcal{D}_2$ as well, as it has support $s \log n$ larger than one from $\mathcal{D}_1$.

Proof. First, observe that given the assumption on the $|A_i|$'s, standard concentration bounds show that, for a random subset S of size s we have $A_i \cap S \neq \emptyset$ for all $i \in [t']$, except with probability $o(1)$. It thus suffices to bound the total variation between the two distributions, conditioned on this event. For any tuple $\mathbf{s} = (s_1, \ldots, s_{t'}) \in A_1 \times \ldots \times A_{t'}$, the probability of observing $\mathbf{s}$ under P2 is

$$\Pr_{\mathrm{P2}}[\,\mathbf{s}\,] = \prod_{i=1}^{t'} \frac{1}{|A_i|}. \tag{2}$$

By the condition of the lemma, $|A_i| > n/s \cdot \log^3 n \geq \log^3 n$. Therefore, the probability that the samples are all distinct is

$$\Pr_{\mathrm{P2}}[\,\text{distinct } s_1, \ldots, s_{t'}\,] \geq 1 \cdot \left(1 - \frac{1}{\log^3 n}\right) \cdots \left(1 - \frac{t'}{\log^3 n}\right) \geq \left(1 - \frac{t'}{\log^3 n}\right)^{t'}.$$

Using the fact that $e^x > 1 + x$ and $1 - x \geq e^{-2x}$ for $0 < x < \frac{1}{2}$, we get

$$\Pr_{\mathrm{P2}}[\,s_1, \ldots, s_{t'} \text{ all distinct }\,] \geq \exp\left(\frac{-2t'^2}{\log^3 n}\right) \geq 1 - \frac{-2t'^2}{\log^3 n} > 1 - \frac{c^2}{\log n}.$$

Focusing hereafter on P1, we show that

$$\sum_{s_1,\ldots,s_{t'} \text{ distinct}} \left|\Pr_{\mathrm{P1}}[\,s_1, \ldots, s_{t'}\,] - \Pr_{\mathrm{P2}}[\,s_1, \ldots, s_{t'}\,]\right| < \frac{1}{\log\log n}. \tag{3}$$

This is enough to bound $\mathrm{d_{TV}}(\mathrm{P1}, \mathrm{P2})$ as follows. By the triangle inequality in Equation 3,

$$\begin{aligned}
\Pr_{\mathrm{P1}}[\,s_1, \ldots, s_{t'} \text{ all distinct }\,] &\geq \Pr_{\mathrm{P2}}[\,s_1, \ldots, s_{t'} \text{ all distinct}\,] - \frac{1}{\log\log n} \\
&> 1 - \frac{c^2}{\log n} - \frac{1}{\log\log n} \\
&> 1 - \frac{2}{\log\log n}\,.
\end{aligned}$$

Therefore,

$$\begin{aligned}
2\mathrm{d_{TV}}(\mathrm{P1}, \mathrm{P2}) &= \sum_{\mathbf{s}} \left|\Pr_{\mathrm{P1}}[\,\mathbf{s}\,] - \Pr_{\mathrm{P2}}[\,\mathbf{s}\,]\right| \\
&= \sum_{s_1,\ldots,s_{t'} \text{ all distinct}} \left|\Pr_{\mathrm{P1}}[\,\mathbf{s}\,] - \Pr_{\mathrm{P2}}[\,\mathbf{s}\,]\right| + \sum_{s_1,\ldots,s_{t'} \text{ not all distinct}} \left|\Pr_{\mathrm{P1}}[\,\mathbf{s}\,] - \Pr_{\mathrm{P2}}[\,\mathbf{s}\,]\right| \\
&\leq \sum_{s_1,\ldots,s_{t'} \text{ all distinct}} \left|\Pr_{\mathrm{P1}}[\,\mathbf{s}\,] - \Pr_{\mathrm{P2}}[\,\mathbf{s}\,]\right| + \frac{4}{\log\log n} \\
&\leq \frac{5}{\log\log n},
\end{aligned}$$

proving that the two processes cannot be distinguished with small error probability.

We now prove Equation 3. Suppose, $X_i = 1$ if $i \in S$, and 0 otherwise. Then, $\sum X_i = s$, and therefore they are negatively correlated. Now, from standard Chernoff-like concentration bounds for negatively correlated random variables (see, i.e., Theorem 4.3 in [DR96]),

Lemma 4.6. *Suppose A is a set of size a, and S is a uniformly chosen random set of size s. Then, for all $\eta \in (0, 1]$*

$$\Pr\left[|A \cap S| > (1+\eta)\frac{as}{n}\right] < e^{-\eta^2 \cdot \frac{as}{3n}}, \text{ and } \Pr\left[|A \cap S| < (1-\eta)\frac{as}{n}\right] < e^{-\eta^2 \cdot \frac{as}{3n}}.$$

We now estimate the probability of observing distinct elements $(s_1, \ldots, s_{t'}) \in A_1 \times \ldots \times A_{t'}$ under P1. The probability that all of the s_i's lie in our random set S is

$$\Pr_S[\,\forall i, s_i \in S\,] = \binom{n-t'}{s-t'}\binom{n}{s}^{-1} = \frac{s(s-1)\ldots(s-t'+1)}{n(n-1)\ldots(n-t'+1)}.$$

Conditioning on this event, Lemma 4.6 and a union bound allow us to bound the size of the intersection of each A_i with S (that now contains $s_1, \ldots, s_{t'}$ by conditioning) as follows, for all $\eta \in (0, 1]$:

$$\Pr\left[\,\forall i \in [t'],\, |A_i \cap S| \in \left[(1-\eta)\frac{a_i s}{n} - t', (1+\eta)\frac{a_i s}{n} + t'\right]\right] \geq 1 - 2t'e^{-\eta^2 \cdot \frac{as}{3n}}.$$

Recall that by assumption $a_i s/n > \log^3 n$ and $t' = O(\log n/\log\log n)$. Taking $\eta = \frac{\Theta(1)}{\log n}$ in the equation above thus implies that, for n sufficiently large, a random set S containing $s_1, \ldots, s_{t'}$ satisfies

$$\Pr\left[\,\forall i \in [t'],\, |A_i \cap S| \in \left[\left(1 - \frac{1}{\log n}\right)\frac{a_i s}{n}, \left(1 + \frac{1}{\log n}\right)\frac{a_i s}{n}\right]\right] > 1 - \frac{2\log n}{n^{1/3}}.$$

Therefore, by Bayes rule we get that

$$\begin{aligned}\Pr_{\text{P1}}[\,(s_1, \ldots, s_{t'})\,] &= \Pr_S[\,\forall i,\ s_i \in S\,] \cdot \Pr_{\text{P1}}[\,(s_1, \ldots, s_{t'}) \mid \forall i, s_i \in S\,] \\ &\in \left[\left(1 - \frac{1}{\log n}\right)^{t'}\left(1 - \frac{t'}{s}\right)^{t'}\left(1 + \frac{t'}{n}\right)^{t'}\prod_{i=1}^{t'}\frac{1}{a_i},\ \left(1 + \frac{1}{\log n}\right)^{t'}\prod_{i=1}^{t'}\frac{1}{a_i}\right].\end{aligned}$$

Since $t' = O(\log n/\log\log n)$, this last interval is a subset of

$$\left[\left(1 - \frac{2}{\log\log n}\right)\prod_{i=1}^{t'}\frac{1}{a_i},\ \left(1 + \frac{2}{\log\log n}\right)\prod_{i=1}^{t'}\frac{1}{a_i}\right].$$

Combined with Equation 2, this implies that for any distinct $s_1, \ldots, s_{t'}$, the ratio of the probabilities of observing these samples under P1 and P2 is within a multiplicative factor of $1 \pm 2/\log\log n$. Therefore, the total variation distance between P1 and P2 is at most $\frac{O(1)}{\log\log n}$, concluding the proof. □

5 A Lower Bound for Equivalence Testing

We prove our main lower bound on the sample complexity of testing equivalence between unknown distributions. We construct two priors $\mathcal{Y}$ and $\mathcal{N}$ over *pairs* of distributions (D_1, D_2) over $[n]$. $\mathcal{Y}$ is a distribution over pairs of distributions of the form (D, D), namely the case when the distributions are identical. Similarly, $\mathcal{N}$ is a distribution over (D_1, D_2) with $\mathrm{d_{TV}}(D_1, D_2) \geq \frac{1}{4}$. We then show that no algorithm $\mathcal{T}$ making $O(\sqrt{\log \log n})$ queries to $\mathsf{COND}^{D_1}, \mathsf{COND}^{D_2}$ can distinguish between a draw from $\mathcal{Y}$ and $\mathcal{N}$ with constant probability (over the choice of (D_1, D_2), the randomness in the samples it obtains, and its internal randomness).

We describe the construction of $\mathcal{Y}$ and $\mathcal{N}$ in Section 5.1, and provide a detailed analysis in Section 5.2.

5.1 Construction

We now summarize how a pair of distribution is constructed under $\mathcal{Y}$ and $\mathcal{N}$. In the subsequent paragraphs, we will give more detail about each specific step.

1. **Effective Support**
   (a) Pick k_b from the set $\{0, 1, \ldots, \frac{1}{2}\log n\}$ at random.
   (b) Let $b = 2^{k_b}$ and $m \stackrel{\text{def}}{=} b \cdot n^{1/4}$.
2. **Buckets**
   (a) ρ and r are chosen with $\sum_{i=1}^{2r} \rho^i = n^{1/4}$.
   (b) Divide $\{1, \ldots, m\}$ into intervals $B_1, \ldots, B_{2r}$ with $|B_i| = b \cdot \rho^i$.
3. **Distributions**
   (a) Assign probability mass $\frac{1}{2r}$ uniformly over B_i to generate distribution D_1.
   (b) (i) Let $\pi_1, \ldots, \pi_r$ be independent 0/1 with $\Pr(\pi_i = 0) = \frac{1}{2}$.
   (ii) If $\pi_i = 0$, assign probability mass $\frac{1}{4r}$ and $\frac{3}{4r}$ over B_{2i-1} and B_{2i} respectively, else $\frac{3}{4r}$ and $\frac{1}{4r}$ respectively. This generates a distribution D_2.
4. **Support relabeling**
   (a) Pick a permutation $\sigma \in S_n$ of the *total* support n.
   (b) Relabel the symbols of D_1 and D_2 according to σ.
5. **Output:** Generate (D_1, D_1) for $\mathcal{Y}$, and (D_1, D_2) otherwise.

We now describe the various steps of the construction in greater detail.

Effective support. Both D_1 and D_2, albeit distributions on $[n]$, will have (common) *sparse* support. The support size is taken to be $m \stackrel{\text{def}}{=} b \cdot n^{1/4}$. Note that, from the above definition, m is chosen uniformly at random from products of $n^{1/4}$ with powers of 2, resulting in values in $[n^{1/4}, n^{3/4}]$.

In this step b will act as a random scaling factor. The objective of this random scaling is to induce uncertainty in the algorithm's knowledge of the true support size of the distributions, and to prevent it from leveraging this information to test equivalence. In fact one can verify that the class

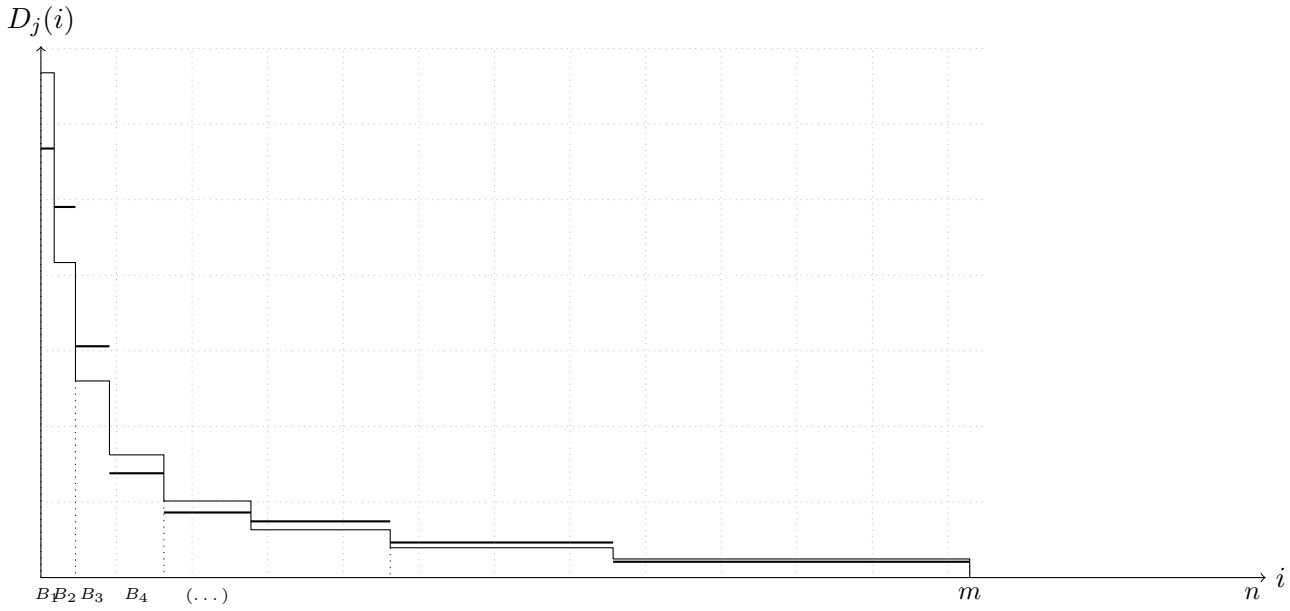

Figure 1: A no-instance (D_1, D_2) (before permutation).

of distributions induced for a single value of b, namely all distributions have the same value of m, then one can distinguish the $\mathcal{Y}$ and $\mathcal{N}$ cases with only $O(1)$ conditional queries.

Buckets. Our construction is inspired by the lower bound of [CRS12, Theorem 8] for the more restrictive PAIRCOND access model. We partition the support in $2r$ consecutive intervals (henceforth referred to as *buckets*) $B_1, \ldots, B_{2r}$, where the size of the i-th bucket is $b\rho^i$. We note that r and ρ will be chosen such that $\sum_{i=1}^{2r} b\rho^i = bn^{1/4}$, i.e., the buckets fill the effective support.

Distributions. We output a pair of distributions (D_1, D_2). Each distribution that we construct is uniform within any particular bucket B_i. In particular, the first distribution assigns the same mass $\frac{1}{2r}$ to each bucket. Therefore, points within B_i have the same probability mass $\frac{1}{(2rb\rho^i)}$. For the $\mathcal{Y}$ case, the second distribution is identical to the first. For the $\mathcal{N}$ case, we pair buckets in r consecutive *bucket-pairs* $\Pi_1, \ldots, \Pi_r$, with $\Pi_i = B_{2i-1} \cup B_{2i}$. For the second distribution D_2, we consider the same buckets as D_1, but repartition the mass $1/r$ *within* each Π_i. More precisely, in each pair, one of the buckets gets now total probability mass $\frac{1}{4r}$ while the other gets $\frac{3}{4r}$ (so that the probability of every point is either decreased by a factor $\frac{1}{2}$ or increased by $\frac{3}{2}$). The choice of which goes up and which goes down is done uniformly and independently at random for each bucket-pair determined by the random choices of π_i's.

Random relabeling. The final step of the construction randomly relabels the symbols, namely is a random injective map from $[m]$ to $[n]$. This is done to ensure that no information about the individual symbol labels can be used by the algorithm for testing. For example, without this the algorithm can consider a few symbols from the first bucket and distinguish the $\mathcal{Y}$ and $\mathcal{N}$ cases. As mentioned in Section 2.3, for ease of analysis, the randomness in the choice of the permutation is, in some sense, deferred to the randomness in the choice of Λ_i during the algorithm's execution.

Summary. A no-instance (D_1, D_2) is thus defined by the following parameters: the support size m, the vector $(\pi_1, \ldots, \pi_m) \in \{0,1\}^r$ (which only impacts D_2), and the final permutation σ of the domain. A yes-instance (D_1, D_1) follows an identical process, however, π has no influence on the final outcome. See Figure 1 for an illustration of such a (D_1, D_2) when σ is the identity permutation and thus the distribution is supported over the first m natural numbers.

Values for ρ and r. By setting $r = \frac{\log n}{8 \log \rho} + O(1)$, we have as desired $\sum_{i=1}^{2r} |B_i| = m$ and there is a factor $(1+o(1))n^{1/4}$ between the height of the first bucket B_1 and the one of the last, B_{2r}. It remains to choose the parameter ρ itself; we shall take it to be $2^{\sqrt{\log n}}$, resulting in $r = \frac{1}{8}\sqrt{\log n} + O(1)$. (Note that for the sake of the exposition, we ignore technical details such as the rounding of parameters, e.g. bucket sizes; these can be easily taken care of at the price of cumbersome case analyses, and do not bring much to the argument.)

5.2 Analysis

We now prove our main lower bound, by analyzing the behavior of core adaptive testers (as per Definition 2.5) on the families $\mathcal{Y}$ and $\mathcal{N}$ from the previous section. In Section 5.2.1, we argue that, with high probability, the sizes of the queries performed by the algorithm satisfy some specific properties. Conditioned upon this event, in Section 5.2.2, we show that the algorithm will get similar information from each query, whether it is running on a yes-instance or a no-instance.

Before moving to the heart of the argument, we state the following fact, straightforward from the construction of our no-instances:

Fact 5.1. *For any (D_1, D_2) drawn from $\mathcal{N}$, one has $\mathrm{d_{TV}}(D_1, D_2) = 1/4$.*

Moreover, as allowing more queries can only increase the probability of success, we hereafter focus on a core adaptive tester that performs exactly $q = \frac{1}{10}\sqrt{\log \log n}$ (adaptive) queries; and will show that it can only distinguish between yes- and no-instances with probability $o(1)$.

5.2.1 Banning "bad queries"

As mentioned in Section 5.1, the draw of a yes- or no-instance involves a random scaling of the size of the support of the distributions, meant to "blind" the testing algorithm. Recall that a testing algorithm is specified by a decision tree, which at step i, specifies how many unseen elements from each atom to include in the query ($\{k_i^A\}$) and which previously seen elements to include in the query (sets $K_i^{(1)}, K_i^{(2)}$, as defined in Section 2.2), where the algorithm's choice depends on the observed configuration at that time. Note that, using Yao's Principle (as discussed in Section 2.3), these choices are deterministic for a given configuration – in particular, we can think of all $\{k_i^A\}$ and $K_i^{(1)}, K_i^{(2)}$ in the decision tree as being fixed. In this section, we show that all k_i^A values satisfy with high probability some particular conditions with respect to the choice of distribution, where the randomness is over the choice of the support size.

First, we recall an observation from [CFGM13], though we modify it slightly to apply to configurations on pairs of distributions and we apply a slightly tighter analysis. This essentially limits the number of states an algorithm could be in by a function of how many queries it makes.

Proposition 5.2. *The number of nodes in a decision tree corresponding to a q-sample algorithm is at most 2^{6q^2+1}.*

Proof. As mentioned in Definition 2.4, an i-configuration can be described using $6i^2$ bits, resulting in at most 2^{6i^2} i-configurations. Since each i-configuration leads us to some node on the i-th level of the decision tree, the total number of nodes can be upper bounded by summing over the number of i-configurations for i ranging from 0 to q, giving us the desired bound. □

For the sake of the argument, we will introduce a few notions applying to the *sizes* of query sets: namely, the notions of a number being *small*, *large*, or *stable*, and of a vector being *incomparable*. Roughly speaking, a number is small if a uniformly random set of this size does not, in expectation, hit the largest bucket B_{2r}. On the other hand, it is large if we expect such a set to intersect many bucket-pairs (i.e., a significant fraction of the support). The definition of stable numbers is slightly more quantitative: a number β is stable if a random set of size β, for each bucket B_i, either completely misses B_i or intersects it in a number of points very close to the expected number (in this case, we say the set *concentrates* over B_i). Finally, a vector of values (β_j) is incomparable if the union of random sets $S_1, \ldots, S_m$ of sizes $\beta_1, \ldots, \beta_m$ contains (with high probability) an amount of mass $D\left(\bigcup_j S_j\right)$ which is either much smaller or much larger than the probability $D(s)$ of any single element s.
We formalize these concepts in the definitions below. To motivate them, it will be useful to bear in mind that, from the construction described in Section 5.1, the expected intersection of a uniform random set of size β with a bucket B_i is of size $\beta b\rho^i/n$; while the expected probability mass from B_i it contains (under either D_1 or D_2) is $\beta/(2rn)$.

Definition 5.3. Let q be an integer, and let $\varphi = \Theta(q^{5/2})$. A number β is said to be *small* if $\beta < \frac{n}{b\rho^{2r}}$; it is *large* (with relation to some integer q) if $\beta \geq \frac{n}{b\rho^{2r-2\varphi}}$.

Note that the latter condition equivalently means that, in expectation, a set of large size will intersect at least $\varphi + 1$ bucket-pairs (as it hits an expected $2\varphi + 1$ buckets, since $\beta|B_{2r-2\varphi}|/n \geq 1$). From the above definitions we get that, with high probability, a random set of any fixed size will in expectation either hit many or no buckets:

Proposition 5.4. *A number is either small or large with probability* $1 - O\left(\frac{\varphi \log \rho}{\log n}\right)$.

Proof. A number β is neither large nor small if $\frac{\rho^{2\varphi} n}{\beta\rho^{2r}} \leq b \leq \frac{n}{\beta\rho^{2r}}$. The ratio of the endpoints of the interval is $\rho^{2\varphi}$. Since $b = 2^{k_b}$, this implies that at most $\log \rho^{2\varphi} = 2\varphi \log \rho$ values of k_b could result in a fixed number falling in this range. As there are $\Theta(\log n)$ values for k_b, the proposition follows. □

The next definition characterizes the sizes of query sets for which the expected intersection with any bucket is either close to 0 (less than $1/\alpha$, for some threshold α), or very big (more than α). (It will be helpful to keep in mind that we will eventually use this definition with $\alpha = \text{poly}(q)$.)

Definition 5.5. A number β is said to be α-*stable* (for $\alpha \geq 1$) if, for each $j \in [2r]$, $\beta \notin \left[\frac{n}{\alpha b\rho^j}, \frac{\alpha n}{b\rho^j}\right]$. A vector of numbers is said to be α-stable if all numbers it contains are α-stable.

Proposition 5.6. *A number is* α*-stable with probability* $1 - O\left(\frac{r \log \alpha}{\log n}\right)$.

Proof. Fix some $j \in [2r]$. A number β does not satisfy the definition of α-stability for this j if $\frac{n}{\alpha\beta\rho^j} \leq b \leq \frac{n\alpha}{\beta\rho^j}$. Since $b = 2^{k_b}$, this implies that at most $\log 2\alpha$ values of k_b could result in a fixed number falling in this range. Noting that there are $\Theta(\log n)$ values for k_b and taking a union bound over all $2r$ values for j, the proposition follows. □

The following definition characterizes the sizes of query sets which have a probability mass far from the probability mass of any individual element. (For the sake of building intuition, the reader may replace ν in the following by the parameter b of the distribution.)

Definition 5.7. A vector of numbers $(\beta_1, \ldots, \beta_\ell)$ is said to be (α, τ)-*incomparable with respect to* ν (for $\tau \geq 1$) if the two following conditions hold.

- $(\beta_1, \ldots, \beta_\ell)$ is α-stable.
- Let Δ_j be the minimum $\Delta \in \{0, \ldots, 2r\}$ such that $\frac{\beta_j \nu \rho^{2r-\Delta}}{n} \leq \frac{1}{\alpha}$, or $2r$ if no such Δ exists. For all $i \in [2r]$, $\frac{1}{2rn} \sum_{j=1}^{\ell} \beta_j \Delta_j \notin \left[\frac{1}{\tau 2r\nu\rho^i}, \frac{\tau}{2r\nu\rho^i}\right]$.

Recall from the definition of α-stability of a number that a random set of this size either has essentially no intersection with a bucket or "concentrates over it" (i.e., with high probability, the probability mass contained in the intersection with this bucket is very close to the expected value). The above definition roughly captures the following. For any j, Δ_j is the number of buckets that will concentrate over a random set of size β_j. The last condition asks that the total probability mass from D_1 (or D_2) enclosed in the union of m random sets of size $\beta_1, \ldots, \beta_\ell$ be a multiplicative factor of τ from the individual probability weight $\frac{1}{2rb\rho^i}$ of a single element from any of the $2r$ buckets.

Proposition 5.8. *Given that a vector of numbers of length ℓ is α-stable, it is (α, q^2)-incomparable with respect to b with probability at least $1 - O\left(\frac{r \log q}{\log n}\right)$.*

Proof. Fix any vector $(\beta_1, \ldots, \beta_\ell)$. By the definition above, for each value b such that $(\beta_1, \ldots, \beta_\ell)$ is α-stable, we have

$$\beta_j \cdot \frac{\alpha\rho^{2r}}{n} \leq \frac{\rho^{\Delta_j}}{b} < \beta_j \cdot \frac{\alpha\rho^{2r+1}}{n}, \quad j \in [\ell]$$

or, equivalently,

$$\frac{\log \frac{\alpha\beta_j}{n}}{\log \rho} + 2r + \frac{\log b}{\log \rho} \leq \Delta_j < \frac{\log \frac{\alpha\beta_j}{n}}{\log \rho} + 2r + \frac{\log b}{\log \rho} + 1, \quad j \in [\ell].$$

Writing $\lambda_j \stackrel{\text{def}}{=} \frac{\log \frac{\alpha\beta_j}{n}}{\log \rho} + 2r$ for $j \in [\ell]$, we obtain that

$$\sum_{j=1}^{\ell} \beta_j \Delta_j b = b \sum_{j=1}^{\ell} \beta_j (\lambda_j + O(1)) + \frac{b \log b}{\log \rho} \sum_{j=1}^{\ell} \beta_j.$$

- If it is the case that $\log \rho \cdot \sum_{j=1}^{\ell} \beta_j (\lambda_j + O(1)) \ll \log b \cdot \sum_{j=1}^{\ell} \beta_j$. Then, for any fixed $i \in [2r]$, to meet the second item of the definition of incomparability we need $\sum_{j=1}^{\ell} \beta_j \Delta_j b \notin [n/(200q\rho^i), 200qn/\rho^i]$. This is essentially, with the assumption above, requiring that

$$b \log b \notin \left[\frac{n \log \rho}{2q^2 \rho^i \sum_{j=1}^{\ell} \beta_j}, \frac{2q^2 n \log \rho}{\rho^i \sum_{j=1}^{\ell} \beta_j}\right].$$

  Recalling that $b \log b = k_b 2^{k_b}$, this means that $O(\log q / \log \log q)$ values of k_b are to be ruled out. (Observe that this is the number of possible "bad values" for b without the condition from the case distinction above; since we add an extra constraint on b, there are at most this many values to avoid.)

- Conversely, if $\log \rho \cdot \sum_{j=1}^{\ell} \beta_j(\lambda_j + O(1)) \gg \log b \cdot \sum_{j=1}^{\ell} \beta_j$ the requirement becomes

$$b \notin \left[\frac{n \log \rho}{2q^2 \rho^i \sum_{j=1}^{\ell} \beta_j(\lambda_j + O(1))}, \frac{2q^2 n \log \rho}{\rho^i \sum_{j=1}^{\ell} \beta_j(\lambda_j + O(1))} \right].$$

  ruling out this time $O(\log q)$ values for k_b.

- Finally, the two terms are comparable only if $\log b = \Theta\Big(\log \rho \cdot \sum_{j=1}^{\ell} \beta_j(\lambda_j + O(1)) \cdot \Big(\sum_{j=1}^{\ell} \beta_j \Big)^{-1} \Big)$; given that $\log b = k_b$, this rules out this time $O(1)$ values for k_b.

A union bound over the $2r$ possible values of i, and the fact that k_b can take $\Theta(\log n)$ values, complete the proof. □

We put these together to obtain the following lemma:

Lemma 5.9. *With probability at least* $1 - O\Big(\frac{2^{6q^2+q}(r \log \alpha + \varphi \log \rho) + 2^{6q^2}(r \log q)}{\log n} \Big)$, *the following holds for the decision tree corresponding to a q-query algorithm:*

- *the size of each atom is α-stable and either large or small;*
- *the size of each atom, after excluding elements we have previously observed,*[4] *is α-stable and either large or small;*
- *for each i, the vector $(k_i^A)_{A \in \mathrm{At}(A_1, \ldots, A_i)}$ is (α, q^2)-incomparable (with respect to b).*

Proof. From Proposition 5.2, there are at most 2^{6q^2+1} tree nodes, each of which contains one vector $(k_i^A)_A$, and at most 2^q atom sizes. The first point follows from Propositions 5.4 and 5.6 and applying the union bound over all $2^{6q^2+1} \cdot 2 \cdot 2^q$ sizes, where we note the additional factor of 2 comes from either including or excluding the old elements. The latter point follows from Proposition 5.8 and applying the union bound over all 2^{6q^2+1} (k_i^A) vectors. □

5.2.2 Key lemma: bounding the variation distance between decision trees

In this section, we prove a key lemma on the variation distance between the distribution on leaves of any decision tree, when given access to either an instance from $\mathcal{Y}$ or $\mathcal{N}$. This lemma will in turn directly yield Theorem 1.1. Hereafter, we set the parameters α (the threshold for stability), φ (the parameter for smallness and largeness) and γ (an accuracy parameter for how well things concentrate over their expected value) as follows:[5] $\alpha \stackrel{\text{def}}{=} q^7$, $\varphi \stackrel{\text{def}}{=} q^{5/2}$ and $\gamma \stackrel{\text{def}}{=} 1/\varphi = q^{-5/2}$. (Recall further that $q = \frac{1}{10}\sqrt{\log \log n}$.)

Lemma 5.10. *Conditioned on the events of Lemma 5.9, consider the distribution over leaves of any decision tree corresponding to a q-query adaptive algorithm when the algorithm is given a yes-instance, and when it is given a no-instance. These two distributions have total variation distance $o(1)$.*

[4] More precisely, we mean to say that for each $i \le q$, for every atom A defined by the partition of $(A_1, \ldots, A_i)$, the values k_i^A and $|A \setminus \{s_1^{(1)}, s_1^{(2)}, \ldots, s_{i-1}^{(1)}, s_{i-1}^{(2)}\}| - k_i^A$ are α-stable and either large or small;

[5] This choice of parameters is not completely arbitrary: combined with the setting of q, r and ρ, they ensure a total bound $o(1)$ on variation distance and probability of "bad events" as well as a (relative) simplicity and symmetry in the relevant quantities.

Proof. This proof is by induction. We will have three inductive hypotheses, $\boldsymbol{E_1}(t), \boldsymbol{E_2}(t)$, and $\boldsymbol{E_3}(t)$. Assuming all three hold for all $t < i$, we prove $\boldsymbol{E_1}(i)$. Additionally assuming $\boldsymbol{E_1}(i)$, we prove $\boldsymbol{E_2}(i)$ and $\boldsymbol{E_3}(i)$.

Roughly, the first inductive hypothesis states that the query sets behave similarly to as if we had picked a random set of that size. It also implies that whether or not we get an element we have seen before is "obvious" based on past observances and the size of the query we perform. The second states that we never observe two distinct elements from the same bucket-pair. The third states that the next sample is distributed similarly in either a yes-instance or a no-instance. Note that this distribution includes both features which our algorithm can observe (i.e., the atom which the sample belongs to and if it collides with a previously seen sample), as well as those which it can not (i.e., which bucket-pair the observed sample belongs to). It is necessary to show the latter, since the bucket-pair a sample belongs to may determine the outcome of future queries.

More precisely, the three inductive hypotheses are as follows:

- $\boldsymbol{E_1}(i)$: In either a yes-instance or a no-instance, the following occurs: For an atom S in the partition generated by $A_1, \dots, A_i$, let $S' = S \setminus \{s_1^{(1)}, s_1^{(2)}, \dots, s_{i-1}^{(1)}, s_{i-1}^{(2)}\}$. For every such S', let $\ell^{S'}$ be the largest index $\ell \in \{0, \dots, 2r\}$ such that $\frac{|S'|b\rho^\ell}{n} \le \frac{1}{\alpha}$, or 0 if no such ℓ exists. We claim that $\ell^{S'} \in \{0, \dots, 2r - \varphi - 2\} \cup \{2r\}$, and say S' is small if $\ell^{S'} = 2r$ and large otherwise. Additionally:
  - for $j \le \ell^{S'}, |S' \cap B_j| = 0$;
  - for $j > \ell^{S'}, |S' \cap B_j|$ lies in $[1 - i\gamma, 1 + i\gamma]\frac{|S'|b\rho^j}{n}$.

  Furthermore, let p_1 and p_2 be the probability mass contained in Λ_i and Γ_i, respectively. Then $\frac{p_1}{p_1+p_2} \le O\left(\frac{1}{q^2}\right)$ or $\frac{p_2}{p_1+p_2} \le O\left(\frac{1}{q^2}\right)$ (that is, either almost all the probability mass comes from elements which we have not yet observed, or almost all of it comes from previously seen ones).
- $\boldsymbol{E_2}(i)$: No two elements from the set $\{s_1^{(1)}, s_1^{(2)}, \dots, s_i^{(1)}, s_i^{(2)}\}$ belong to the same bucket-pair.
- $\boldsymbol{E_3}(i)$: Let T_i^{yes} be the random variable representing the atoms and bucket-pairs[6] containing $(s_i^{(1)}, s_i^{(2)})$, as well as which of the previous samples they intersect with, when the i-th query is performed on a yes-instance, and define T_i^{no} similarly for no-instances. Then $\mathrm{d_{TV}}(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}}) \le O\left(\frac{1}{q^2} + \frac{1}{\rho} + \gamma + \frac{1}{\varphi}\right) = o(1)$.

We will show that $\boldsymbol{E_1}(i)$ holds with probability $1 - O\left(2^i \exp\left(-\frac{2\gamma^2\alpha}{3}\right)\right)$ and $\boldsymbol{E_2}(i)$ holds with probability $1 - O(i/\varphi)$. Let T^{yes} be the random variable representing the q-configuration and the bucket-pairs containing each of the observed samples in a yes-instance, and define T^{no} similarly for a no-instance. We note that this random variable determines which leaf of the decision tree we reach. By a union bound, coupling argument, and triangle inequality, the total variation distance between T^{yes} and T^{no} will be $O\left(2^q \exp\left(-\frac{2\gamma^2\alpha}{3}\right) + \frac{q^2}{\varphi} + \frac{1}{q} + \frac{q}{\rho} + q\gamma + \frac{q}{\varphi}\right) = o(1)$ (from our choice of α, γ, φ), giving the desired result.

We proceed with the inductive proofs of $\boldsymbol{E_1}(i)$, $\boldsymbol{E_2}(i)$, and $\boldsymbol{E_3}(i)$, noting that the base cases hold trivially for all three of these statements. Throughout this proof, recall that Λ_i is the set of unseen support elements which we query, and Γ_i is the set of previously seen support elements which we query.

[6] If a sample $s_i^{(k)}$ does not belong to any bucket (if the corresponding i-th query did not intersect the support), it is marked in T_i^{yes} with a "dummy label" to indicate so.

Lemma 5.11. *Assuming that $\boldsymbol{E_1}(t), \boldsymbol{E_2}(t), \boldsymbol{E_3}(t)$ hold for all $1 \le t \le i-1$, then $\boldsymbol{E_1}(i)$ holds with probability at least $1 - O\left(2^i \exp\left(-\frac{2\gamma^2\alpha}{3}\right)\right) = 1 - 2^{i-\Omega(q^2)}$.*

Proof. We start with the first part of the statement of $\boldsymbol{E_1}(i)$, prior to "Furthermore"; and let S (and the corresponding S') be any atom as in $\boldsymbol{E_1}(i)$. First, we note that $\ell^{S'} \in \{0, \dots, 2r-\varphi-2\} \cup \{2r\}$ since we are conditioning on Lemma 5.9: $|S'|$ is α-stable and either large or small, which enforces this condition.

Next, suppose S' is contained in some other atom T generated by $A_1, \dots, A_{i-1}$, and let $T' = T \setminus \{s_1^{(1)}, s_1^{(2)}, \dots, s_{i-1}^{(1)}, s_{i-1}^{(2)}\}$. Since $|S'| \le |T'|$, this implies that $\ell^{T'} \le \ell^{S'}$. We argue about $|T' \cap B_j|$ for three regimes of j:

- The first case is $j \le \ell^{T'}$. By the inductive hypothesis, $|T' \cap B_j| = 0$, so $|S' \cap B_j| = 0$ with probability 1.
- The next case is $\ell^{T'} < j \le \ell^{S'}$. Recall from the definition of an adaptive core tester that S' will be chosen uniformly at random from all subsets of T' of the appropriate size. By the inductive hypothesis,
$$\frac{|T' \cap B_j|}{|T'|} \in [1-(i-1)\gamma, 1+(i-1)\gamma]\,\frac{b\rho^j}{n},$$
and therefore
$$\mathbb{E}\llbracket S' \cap B_j \rrbracket \in [1-(i-1)\gamma, 1+(i-1)\gamma]\,\frac{|S'|\,b\rho^j}{n}, \quad \text{implying } \mathbb{E}\llbracket S' \cap B_j \rrbracket \le \frac{2}{\alpha\rho^{\ell^{S'}-j}};$$
where the inequality is by the definition of $\ell^{S'}$ and using the fact that $(i-1)\gamma \le 1$. Using a Chernoff bound for negatively correlated random variables (as in e.g. [DR96]),
$$\begin{aligned}\Pr[\,|S' \cap B_j| \ge 1\,] &= \Pr\left[|S' \cap B_j| \ge \left(1 + \frac{1-\mu}{\mu}\right)\mu\right] \\ &\le \exp\left(-\frac{(1-\mu)^2}{3\mu}\right) \\ &\le \exp\left(-\frac{1}{12}\alpha\rho^{\ell^{S'}-j}\right),\end{aligned}$$
where the second inequality holds because $\mu \le \frac{2}{\alpha\rho^{\ell^{S'}-j}}$ and $(1-\mu)^2 \ge \frac{1}{2}$ for n sufficiently large.
- The final case is $j > \ell^{S'}$. As in the previous one,
$$\mathbb{E}\llbracket S' \cap B_j \rrbracket \in [1-(i-1)\gamma, 1+(i-1)\gamma]\,\frac{|S'|\,b\rho^j}{n}, \quad \text{implying } \mathbb{E}\llbracket S' \cap B_j \rrbracket \ge \frac{\alpha\rho^{j-\ell^{S'}-1}}{2};$$
where the inequality is by the definition of $\ell^{S'}$, α-stability, and using the fact that $(i-1)\gamma \le \frac{1}{2}$. Using again a Chernoff bound for negatively correlated random variables,
$$\begin{aligned}\Pr\left[|S' \cap B_j| - \mathbb{E}\llbracket S' \cap B_j \rrbracket \ge \gamma\frac{|S'|\,b\rho^j}{n}\right] &\le \Pr[\,|S' \cap B_j| - \mathbb{E}\llbracket S' \cap B_j \rrbracket \ge \gamma 2\mathbb{E}\llbracket S' \cap B_j \rrbracket\,] \\ &\le 2\exp\left(-\frac{(2\gamma)^2\mathbb{E}\llbracket S' \cap B_j \rrbracket}{3}\right) \\ &\le 2\exp\left(-\frac{2}{3}\gamma^2\alpha\rho^{j-\ell^{S'}-1}\right)\end{aligned}$$

where the first inequality comes from $2(1-(i-1)\gamma) \geq 1$, the second one is the Chernoff bound, and the third derives from $\mathbb{E}[|S' \cap B_j|] \geq \frac{\alpha\rho^{j-\ell^{S'}-1}}{2}$.

Combining these, the probability that S' does not satisfy the required conditions is at most

$$\sum_{j \leq \ell^{T'}} 0 + \sum_{\ell^{T'} < j \leq \ell^{S'}} \exp\left(-\frac{1}{12}\alpha\rho^{\ell^{S'}-j}\right) + \sum_{j > \ell^{S'}} 2\exp\left(-\frac{2}{3}\gamma^2\alpha\rho^{j-\ell^{S'}-1}\right).$$

This probability is maximized when $\ell^{S'} = \ell^{T'} = 0$, in which case it is

$$\sum_{j=1}^{2r} 2\exp\left(-\frac{2}{3}\gamma^2\alpha\rho^{j-1}\right) \leq \sum_{j=1}^{\infty} 2\exp\left(-\frac{2}{3}\gamma^2\alpha\rho^{j-1}\right) \leq 3\exp\left(-\frac{2}{3}\gamma^2\alpha\right).$$

Taking a union bound over at most 2^i sets gives us the desired probability bound.

Finally, we prove the statement following "Furthermore"; this will follow from the definition of incomparability.

- First, we focus on Γ_i. Suppose that Γ_i contains at least one element with positive probability mass (if not, the statement trivially holds). Let p_2' be the probability mass of the heaviest element in Γ_i. Since our inductive hypothesis implies that Γ_i has no elements in the same bucket pair, the maximum possible value for p_2 is

$$\begin{aligned} p_2 &\leq p_2' + \frac{3p_2'}{\rho} + \frac{3p_2'}{\rho^3} + \cdots \leq p_2' + \frac{3p_2'}{\rho}\sum_{k=0}^{\infty}\frac{1}{\rho^{2k}} = \left(1 + \frac{3}{\rho}\frac{\rho^2}{\rho^2-1}\right)p_2' \\ &\leq (1+o(1))p_2' \end{aligned}$$

  Therefore, $p_2 \in [p_2', (1+o(1))p_2']$. Supposing this heaviest element belongs to bucket j, we can say that $p_2 \in [\frac{1}{2}, (1+o(1))\frac{3}{2}]\frac{1}{2rb\rho^j}$.
- Next, we focus on Λ_i. Consider some atom A, from which we selected k_A elements which have not been previously observed: call the set of these elements A'. In the first part of this proof, we showed that for each bucket B_k, either $|A' \cap B_k| = 0$ or $|A' \cap B_k| \in [1-i\gamma, 1+i\gamma]\frac{|A'|b\rho^k}{n}$. In the latter case, noting that $i\gamma \leq \frac{1}{2}$ and that the probability of an individual element in B_k is within $[1,3]\frac{1}{4rb\rho^k}$, the probability mass contained by $|A' \cap B_k|$ belongs to $[1,9]\frac{|A'|}{8rn}$. Recalling the definition of Δ_A as stated in Definition 5.7, as shown earlier in this proof, this non-empty intersection happens for exactly Δ_A bins. Therefore, the total probability mass in Λ_i is in the interval $\left[\frac{1}{4}, \frac{9}{4}\right]\frac{1}{2rn}\sum_{A \in \mathrm{At}(A_1,\ldots,A_i)} k_i^A \Delta_A$.

Recall that we are conditioning on Lemma 5.9 which states that the vector $(k_i^A)_{A \in \mathrm{At}(A_1,\ldots,A_i)}$ is (α, q^2)-incomparable with respect to b. Applying this definition to the bounds just obtained on the probability masses in Λ_i and Γ_i gives the desired result. □

Lemma 5.12. *Assuming that $\boldsymbol{E_1}(t), \boldsymbol{E_2}(t), \boldsymbol{E_3}(t)$ hold for all $1 \leq t \leq i-1$ and additionally $\boldsymbol{E_1}(i)$, then $\boldsymbol{E_2}(i)$ holds with probability at least $1 - O\left(\frac{i}{\varphi}\right)$.*

Proof. We focus on $s_i^{(1)}$. If $s_i^{(1)} \in \Gamma_i$, the conclusion is trivial, so suppose $s_i^{(1)} \in \Lambda_i$. From $\boldsymbol{E_1}(i)$, no small atom intersects any of the buckets, so condition that $s_i^{(1)}$ belongs to some large atom S. Since

we want $s_i^{(1)}$ to fall in a distinct bucket-pair from $2(i-1)+1$ other samples, there are at most $2i-1$ bucket-pairs which $s_i^{(1)}$ should not land in. Using $\boldsymbol{E_1}(i)$, the maximum probability mass contained in the intersection of these bucket-pairs and S is $(1+i\gamma)(2i-1)\frac{|S|}{rn}$. Similarly, additionally using the definition of a large atom, the minimum probability mass contained in S is $(1-i\gamma)\varphi\frac{|S|}{rn}$. Taking the ratio of these two terms gives an upper bound on the probability of breaking this invariant, conditioned on landing in S, as $O(i/\varphi)$, where we note that $\frac{1+i\gamma}{1-i\gamma} = O(1)$. Since the choice of large atom was arbitrary, we can remove the conditioning. Taking the union bound for $s_i^{(1)}$ and $s_i^{(2)}$ gives the result. □

Lemma 5.13. *Assuming that $\boldsymbol{E_1}(t), \boldsymbol{E_2}(t), \boldsymbol{E_3}(t)$ hold for all $1 \le t \le i-1$ and additionally $\boldsymbol{E_1}(i)$, then $\boldsymbol{E_3}(i)$ holds.*

Proof. We focus on some fixed setting of the history of the interaction, i.e. the configuration and the bucket-pairs the past elements belong to, and show that the results of the next query will behave similarly, whether the instance is a yes-instance or a no-instance. We note that, since we are assuming the inductive hypotheses hold, certain settings which violate these hypotheses are not allowed. We also note that $s_i^{(1)}$ is distributed identically in both instances, so we focus on $s_i \stackrel{\text{def}}{=} s_i^{(2)}$ for the remainder of this proof.
First, we condition that, based on the setting of the past history, s_i will either come from Λ_i or Γ_i – this event happening with probability $1 - O(1/q^2)$.

Proposition 5.14. *In either a yes-instance or a no-instance, s_i will either come from Λ_i or Γ_i with probability $1 - O\left(\frac{1}{q}\right)$, where the choice of which one is deterministic based on the fixed configuration and choice for the bucket-pairs of previously seen elements.*

Proof. This is simply a rephrasing of the portion of $\boldsymbol{E_1}(i)$ following "Furthermore." □

By a coupling argument, after conditioning on this event, we must show that the total variation distance in either case is at most $O\left(\frac{1}{\rho} + \gamma + \frac{1}{\varphi}\right) = O\left(\frac{1}{q^{5/2}}\right)$. We break this into two cases, the first being when s comes from Γ_i. In this case, we incur a cost in total variation distance which is $O(1/\rho)$:

Proposition 5.15. *In either a yes-instance or a no-instance, condition that s_i comes from Γ_i. Then. one of the following holds:*

- *$|\Gamma_i \cap B_j| = 0$ for all $j \in [2r]$, in which case s_i is distributed uniformly at random from the elements of Γ_i;*
- *or $|\Gamma_i \cap B_j| \neq 0$ for some $j \in [2r]$, in which case s_i will be equal to some $s \in \Gamma_i$ with probability $1 - O\left(\frac{1}{\rho}\right)$, where the choice of s is deterministic based on the fixed configuration and choice for the bucket-pairs of previously seen elements.*

Proof. The former case follows from the definition of the sampling model. For the latter case, let p be the probability mass of the heaviest element in Γ_i. Since our inductive hypothesis implies that Γ_i has no elements in the same bucket-pair, the maximum possible value for the rest of the elements is

$$\frac{3p}{\rho} + \frac{3p}{\rho^3} + \frac{3p}{\rho^5} + \cdots \le \frac{3p}{\rho}\sum_{k=0}^{\infty}\frac{1}{\rho^{2k}} = \frac{3p}{\rho}\frac{\rho^2}{\rho^2-1} = O\left(\frac{p}{\rho}\right).$$

Since the ratio of this value and p is $O\left(\frac{1}{\rho}\right)$, with probability $1 - O\left(\frac{1}{\rho}\right)$ the sample returned is the heaviest element in Γ_i. □

Finally, we examine the case when s comes from Λ_i:

Proposition 5.16. *Condition that s_i comes from Λ_i. Then either:*

- $|\Lambda_i \cap B_j| = 0$ *for all* $j \in [2r]$*, in which case* $\mathrm{d_{TV}}(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}}) = 0$*;*
- *or* $|\Lambda_i \cap B_j| \neq 0$ *for some* $j \in [2r]$*, in which case* $\mathrm{d_{TV}}(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}}) \leq O\left(\gamma + \frac{1}{\varphi}\right) = O\left(\frac{1}{q^{5/2}}\right)$

Proof. The former case follows from the definition of the sampling model – since Λ_i does not intersect any of the buckets, the sample will be labeled as such. Furthermore, the sample returned will be drawn uniformly at random from Λ_i, and the probability of each atom will be proportional to the cardinality of its intersection with Λ_i, in both the yes- and the no-instances.

We next turn to the latter case. Let $\mathcal{X}$ be the event that, if the intersection of Λ_i and some atom A has a non-empty intersection with an odd number of buckets, then s_i does not come from the unpaired bucket. Note that $\boldsymbol{E_1}(i)$ and the definition of a large atom imply that an unpaired bucket can only occur if the atom intersects at least φ bucket-pairs: conditioned on the sample coming from a particular atom, the probability that it comes from the unpaired bucket is $O(1/\varphi)$. Since the choice of A was arbitrary, we may remove the conditioning, and note that $\Pr(\mathcal{X}) = 1 - O(1/\varphi)$.

Since

$$\begin{aligned}\mathrm{d_{TV}}(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}}) &\leq \mathrm{d_{TV}}(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}} \mid \mathcal{X}) \Pr(\mathcal{X}) + \mathrm{d_{TV}}(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}} \mid \bar{\mathcal{X}}) \Pr(\bar{\mathcal{X}}) \\ &\leq \mathrm{d_{TV}}(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}} \mid \mathcal{X}) + O(1/\varphi),\end{aligned}$$

it remains to show that $\mathrm{d_{TV}}(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}} \mid \mathcal{X}) \leq O(\gamma)$.

First, we focus on the distribution over atoms, conditioned on $\mathcal{X}$. Let N^A be the number of bucket-pairs with which A intersects both buckets, i.e., conditioned on $\mathcal{X}$, the sample could come from $2N^A$ buckets, and let $N \stackrel{\text{def}}{=} \sum_{A \in \mathrm{At}(A_1,\ldots,A_i)} N^A$. By $\boldsymbol{E_1}(i)$, the maximum amount of probability mass that can be assigned to atom A is $\frac{(1+\gamma)|S|N^A/rn}{(1-\gamma)|S|N/rn}$, and the minimum is $\frac{(1-\gamma)|S|N^A/rn}{(1+\gamma)|S|N/rn}$, so the total variation distance in the distribution incurred by this atom is at most $O\left(\gamma N^A/N\right)$. Summing over all atoms, we get the desired $O(\gamma)$.

Finally, we bound the distance on the distribution over bucket-pairs, again conditioned on $\mathcal{X}$. By $\boldsymbol{E_1}(i)$ only large atoms will contain non-zero probability mass, so condition on the sample coming from some large atom A. Let N^A be the number of bucket-pairs with which A intersects both buckets, i.e., conditioned on $\mathcal{X}$, the sample could come from $2N^A$ buckets. Using $\boldsymbol{E_1}(i)$, the maximum amount of probability mass that can be assigned to any intersecting bucket-pair is $\frac{(1+\gamma)\frac{|A|}{rn}}{(1-\gamma)\frac{|A|}{rn}N^A}$, and the minimum is $\frac{(1-\gamma)\frac{|A|}{rn}}{(1+\gamma)\frac{|A|}{rn}N^A}$, so the total variation distance in the distribution incurred by this bucket-pair is at most $O\left(\gamma\frac{1}{N^A}\right)$. Summing this difference over all N^A bucket-pairs, we get $\frac{2\gamma}{1-\gamma^2} = O(\gamma)$. Since the choice of large atom A was arbitrary, we can remove the conditioning on the choice of atom. The statement follows by applying the union bound on the distribution over bucket-pairs and the distribution over atoms. □

We note that in both cases, the cost in total variation distance which is incurred is $O\left(\frac{1}{\rho}+\gamma+\frac{1}{\varphi}\right)$, which implies $\boldsymbol{E_3}(i)$. □

This concludes the proof of Lemma 5.10. □

With this lemma in hand, the proof of the main theorem is straightforward:

Proof of Theorem 1.1. Conditioned on Lemma 5.9, Lemma 5.10 implies that the distribution over the leaves in a yes-instance vs. a no-instance is $o(1)$. Since an algorithm's choice to accept or reject depends deterministically on which leaf is reached, this bounds the difference between the conditional probability of reaching a leaf which accepts. Since Lemma 5.9 occurs with probability $1-o(1)$, the difference between the unconditional probabilities is also $o(1)$. □

Acknowledgments. Clément Canonne would like to thank Dana Ron and Rocco Servedio for the many helpful discussions and remarks that influenced the lower bound construction of Section 5.

References

[ADJ+11] Jayadev Acharya, Hirakendu Das, Ashkan Jafarpour, Alon Orlitsky, and Shengjun Pan. Competitive closeness testing. In Sham M. Kakade and Ulrike von Luxburg, editors, *COLT*, volume 19 of *JMLR Proceedings*, pages 47–68. JMLR.org, 2011. 1

[ADJ+12] Jayadev Acharya, Hirakendu Das, Ashkan Jafarpour, Alon Orlitsky, Shengjun Pan, and Ananda Theertha Suresh. Competitive classification and closeness testing. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *COLT*, volume 23 of *JMLR Proceedings*, pages 22.1–22.18. JMLR.org, 2012. 1

[BFF+01] Tuğkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Proceedings of FOCS*, pages 442–451, 2001. 1

[BFR+00] Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing that distributions are close. In *Proceedings of FOCS*, pages 189–197, 2000. 1, 5.2.2

[BFR+10] Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing closeness of discrete distributions. (abs/1009.5397), 2010. This is a long version of [BFR+00]. 1.1, 1.1

[BFRV11] Arnab Bhattacharyya, Eldar Fischer, Ronitt Rubinfeld, and Paul Valiant. Testing monotonicity of distributions over general partial orders. In *Proceedings of ITCS*, pages 239–252, 2011. 1

[BKR04] Tuğkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *Proceedings of STOC*, pages 381–390, New York, NY, USA, 2004. ACM. 1

[CDVV14] Siu-On Chan, Ilias Diakonikolas, Gregory Valiant, and Paul Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of SODA*, pages 1193–1203. Society for Industrial and Applied Mathematics (SIAM), 2014. 1, 1.1

[CFGM13] Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. In *Proceedings of ITCS*, pages 561–580, New York, NY, USA, 2013. ACM. (document), 1, 1.1, 1.3, 1.3, 2.1, 2.2, 5.2.1

[CR14] Clément L. Canonne and Ronitt Rubinfeld. Testing probability distributions underlying aggregated data. In *Proceedings of ICALP*, pages 283–295, 2014. 1

[CRS12] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing probability distributions using conditional samples. Technical Report abs/1211.2664, ArXiV, November 2012. (document), 1, 1.1, 1.1, 2, 1.3, 2.1, 3.1, 3.2, 5.1

[DR96] Devdatt Dubhashi and Desh Ranjan. Balls and Bins: A Study in Negative Dependence. *Random Structures and Algorithms*, 13:99–124, 1996. 4, 5.2.2

[GMV06] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *Proceedings of SODA*, pages 733–742, Philadelphia, PA, USA, 2006. Society for Industrial and Applied Mathematics (SIAM). 1

[GR00] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. Technical Report TR00-020, Electronic Colloquium on Computational Complexity (ECCC), 2000. 1, 1.1, 1.1

[ILR12] Piotr Indyk, Reut Levi, and Ronitt Rubinfeld. Approximating and Testing k-Histogram Distributions in Sub-linear Time. In *Proceedings of PODS*, pages 15–22, 2012. 1

[LRR13] Reut Levi, Dana Ron, and Ronitt Rubinfeld. Testing properties of collections of distributions. *Theory of Computing*, 9(8):295–347, 2013. 1

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995. 2.3

[Pan08] Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008. 1, 1.1

[RS09] Ronitt Rubinfeld and Rocco A. Servedio. Testing monotone high-dimensional distributions. *Random Structures and Algorithms*, 34(1):24–44, January 2009. 1

[RT14] Dana Ron and Gilad Tsur. The power of an example: Hidden set size approximation using group queries and conditional sampling. *CoRR*, abs/1404.5568, 2014. (document), 1.1, 1.2, 1.2, 1.3, 3.3, 4, 4

[Rub12] Ronitt Rubinfeld. Taming Big Probability Distributions. *XRDS*, 19(1):24–28, September 2012. 1

[Sto85] L. Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing*, 14(4):849–861, 1985. 1.3

[Sub] List of Open Problems in Sublinear Algorithms: Problem 66. http://sublinear.info/66. Bertinoro Workshop on Sublinear Algorithms 2014 (suggested by Eldar Fischer). (document), 1.1, 1.2

[Val11] Paul Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing*, 40(6):1927–1968, 2011. 1.1

[VV10a] Gregory Valiant and Paul Valiant. A CLT and tight lower bounds for estimating entropy. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:179, 2010. 1.1, 5.2.2

[VV10b] Gregory Valiant and Paul Valiant. Estimating the unseen: A sublinear-sample canonical estimator of distributions. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:180, 2010. 5.2.2

[VV11] Gregory Valiant and Paul Valiant. The power of linear estimators. In *Proceedings of FOCS*, pages 403–412, October 2011. See also [VV10a] and [VV10b]. 1.1

[VV14] Gregory Valiant and Paul Valiant. An automatic inequality prover and instance optimal identity testing. In *Proceedings of FOCS*, 2014. 1, 1.1

ECCC ISSN 1433-8092

http://eccc.hpi-web.de