THEORY OF COMPUTING, Volume 14 (19), 2018, pp. 1–46
www.theoryofcomputing.org

A Chasm Between Identity and Equivalence Testing with Conditional Queries

Jayadev Acharya* Clément L. Canonne† Gautam Kamath‡

Received December 11, 2015; Revised February 14, 2018; Published December 19, 2018

Abstract: A recent model for property testing of probability distributions (Chakraborty et al., ITCS 2013, Canonne et al., SICOMP 2015) 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, Ron, and Servedio (SICOMP 2015) 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 queries (i. e., samples), independent of the support size n—in contrast to the required $\Omega(\sqrt{n})$ in the standard sampling model. However, it was unclear whether the same stark contrast exists for the case of testing equivalence, where *both* distributions are unknown. Indeed, while Canonne et al. established a $\mathrm{poly}(\log n)$-query upper bound for equivalence testing, very recently brought down to $\tilde{O}(\log\log n)$ by Falahatgar et al. (COLT 2015), whether a dependence on the domain size n is necessary was

An extended abstract of this paper, with only some of the proofs and a subset of the results on non-adaptive algorithms, appeared in the Proceedings of the 2015 Conference on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM) [3].

*Supported by a Cornell University Start Up, and NSF CRII-CIF-1657471. Part of this work was done when the author was a postdoctoral researcher at MIT.

†Supported by a Motwani Postdoctoral Fellowship. Part of this work was performed while the author was a graduate student at Columbia University, and supported by NSF grants CCF-1115703 and NSF CCF-1319788.

‡This work was supported by ONR N00014-12-1-0999, and NSF grants CCF-0953960 (CAREER) and CCF-1101491.

ACM Classification: F.2, G.2, G.3

AMS Classification: 68Q32, 68W20, 68Q25, 68Q17

Key words and phrases: property testing, distribution testing, conditional sampling, lower bounds, equivalence testing, uniformity testing, support size estimation

 DOI: 10.4086/toc.2018.v014a019

still open, and explicitly posed by Fischer at the Bertinoro Workshop on Sublinear Algorithms (2014). In this article, we answer the question in the affirmative, 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 a gap between identity and equivalence testing, absent in the standard sampling model (where both problems have sampling complexity $n^{\Theta(1)}$).

We also obtain results on the query complexity of uniformity testing and support-size estimation with conditional samples. In particular, we answer a question of Chakraborty et al. (ITCS 2013) showing that *non-adaptive* uniformity testing indeed requires $\Omega(\log n)$ queries in the conditional model. This is an exponential improvement on their previous lower bound of $\Omega(\log\log n)$, and matches up to polynomial factors their $\mathrm{poly}(\log n)$ upper bound. For the problem of support-size estimation, we provide both adaptive and non-adaptive algorithms, with query complexities $\mathrm{poly}(\log\log n)$ and $\mathrm{poly}(\log n)$, respectively, and complement them with a lower bound of $\Omega(\log n)$ conditional queries for non-adaptive algorithms.

1 Introduction

"*No, Virginia, there is no constant-query tester.*"[1]

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 work of Goldreich and Ron [27] and Batu et al. [9] that the problem has been considered through the lens of theoretical computer science, more particularly in the setting of *property testing*. In this framework, an unknown "huge object"—here a probability distribution over a humongous domain—can be accessed only by making a few local inspections, and the goal is to decide whether the object satisfies some prespecified property. While most of the literature focuses on the large sample regime and studies error exponents and rates of convergence, more recent testing algorithms look at these problems in the small-sample regime focusing on the probability of errors and sample complexity. (We refer the reader to [25, 35, 36, 26] for an introduction and surveys on the general field of property testing. Moreover, due to the specificities of our model we will interchangeably use the terms sample and query complexity, referring to conditional samples as "queries.")

Over the subsequent decade, a flurry of work explored this new area, resulting in a better and often complete understanding of a number of questions in property testing of distributions, or distribution testing (see, e. g., [27, 8, 11, 33, 39, 4, 12, 38, 29, 5, 18, 45] or [13] for a survey). In many cases, these culminated in provably sample-optimal algorithms, all of which required at least an $n^{\Omega(1)}$ dependence on the domain size n in the sample complexity—a dependence which, while sublinear, can still be prohibitively large. However, the standard setting of distribution testing, where one only obtains independent samples from an unknown distribution D, does not encompass *all* scenarios one may encounter. In recent years, alternative models have thus been proposed to capture more specific situations [28, 16, 14, 30, 15]. Among these is the *conditional oracle model* [16, 14] which will be the focus of our work. In this setting, the testing algorithm is given the ability to sample from conditional distributions: that is, to specify a subset S of the domain and obtain samples from D_S, the distribution induced by D on S (the formal definition of the

[1] The curious reader is referred to [47].

model can be found in Definition 2.1). The hope is that allowing a richer set of queries to the unknown underlying distributions might significantly reduce the number of samples the algorithms need, thereby sidestepping the strong lower bounds that hold in the standard sampling model.

1.1 Motivation for the conditional model

A recent trend in testing and learning circumvents these impossibility results by providing a more flexible type of queries than independent samples. One example can be found in the recent paradigm of *active learning* [20, 40] (and its testing counterpart, *active testing* [7]), which modifies and generalizes the usual unsupervised learning paradigm. In this setting, the algorithm is provided with unlabeled examples only, but can then adaptively *request* the label of any of these examples. While not directly comparable to these two frameworks (which are not applicable to the study of probability distributions), the conditional sampling model we shall work on shares some similarity in spirit. Specifically, it provides the algorithm with some additional power, and the ability to perform (some type of) more powerful queries to the object to be learned or tested.

The setting of conditional sampling is related to that of *group testing*, where the objective is to identify a set of defective individuals among a large population, by querying whether suitably chosen subsets contain at least one defective individual. Group testing has been a field of interest since the 40's and has remained an active area of research since (see, e. g., [21, 22, 32, 17]). The type of queries allowed in this framework is reminiscent of conditional sampling, where one obtains samples conditioned on a subset. This connection between group testing and conditional sampling is explored in greater detail in [2].

Of course, a crucial aspect of designing and studying these new models of learning and testing is to understand how justified and natural they are, and argue that they do indeed capture natural situations. In the case of distribution testing, the conditional access model does meet these criteria, as discussed in [16] and [14]. Namely, besides the purely theoretical aspect of helping understand the key aspects and limitations of the underlying statistical problems, this framework is characteristic of situations which arise in natural and social sciences. At a very high-level, *any* scenario where an experimenter or practitioner is able to restrict the set of possible outcomes of an experiment or poll—e. g., in chemistry, where one might control some factor such as the acidity of a solution; or in sociology by performing stratified polling—provides this experimenter with the sort of access granted by the conditional model.

1.2 Background and previous work

We focus in this paper on proving lower bounds for testing two extremely natural properties of distributions, namely *equivalence testing* ("are these two distributions identical?") and *support-size estimation* ("how many different outcomes can actually be observed?"). 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 an informal definition of these two problems, along with closely related ones (uniformity and identity testing). Hereafter, "oracle access" to a distribution D over $[n] = \{1,\ldots,n\}$ means access to samples generated independently from D, and "far" is with respect to the total variation distance $\mathrm{d_{TV}}(D_1, D_2) = \sup_{S\subseteq[n]}(D_1(S) - D_2(S))$ between probability distributions. Moreover, as in usual in property testing, in all the problems below we allow the algorithms to be randomized. In the description of the results below, we will often consider the distance parameter ε as a constant (and focus on the domain size n as the key parameter).

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

Identity testing: granted oracle access to D and the full description of a fixed D^*, decide whether D equals D^* or is far from it.

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

Support-size estimation: granted oracle access to D, return a multiplicative approximation of the size of the support[2] $\operatorname{supp}(D) = \{ x : D(x) > 0 \}$.

It is not difficult to see that each of the second and third problems generalizes the previous, and therefore has query complexity at least as big. All of these tasks are known to require sample complexity $n^{\Omega(1)}$ in the standard sampling model (SAMP); yet, as prior work [16, 14] shows, their complexity decreases tremendously when one allows the more flexible type of access to the distribution(s) provided by a conditional sampling oracle (COND). For the problems of uniformity testing and identity testing, the sample complexity even becomes a constant provided the testing algorithm is allowed to be *adaptive* (i. e., when the next queries it makes can depend on the samples it previously obtained).

Testing uniformity and identity. The identity testing question is a generalization of uniformity testing, where D^* is taken to be the uniform distribution over $[n]$. The complexity of both tasks is well-understood in the sampling model; in particular, it is known that for both uniformity and identity testing $\Theta\left(\sqrt{n}/\varepsilon^2\right)$ samples are necessary and sufficient (see [27, 10, 33, 45] for the tight bounds on these problems).

The uniformity testing problem exemplifies the savings granted by conditional sampling—as Canonne, Ron, and Servedio [14] showed, in this setting only $\tilde{O}\left(1/\varepsilon^2\right)$ adaptive queries[3] are sufficient (and this is optimal, up to logarithmic factors). They further prove that identity testing has constant sample complexity as well, namely $\tilde{O}\left(1/\varepsilon^4\right)$—very recently improved to a near-optimal $\tilde{O}\left(1/\varepsilon^2\right)$ by Falahatgar et al. [24]. 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 *independent of the domain size n*.

The aforementioned algorithms crucially leverage the ability to make adaptive conditional queries to the probability distributions. Restricting the study to non-adaptive algorithms, Chakraborty et al. [16] describe a $\operatorname{poly}(\log n, 1/\varepsilon)$-query non-adaptive tester for uniformity, showing that even without the full power of conditional queries one can still get an exponential improvement over the standard sampling setting. They also obtain an $\Omega(\log\log n)$ lower bound for this problem, and leave open the possibility of improving this lower bound to a logarithmic dependence. We answer this question, establishing in Theorem 1.3 that any non-adaptive uniformity tester must perform $\Omega(\log n)$ conditional queries.

[2]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 $D(i) \propto 1/2^{in}$).

[3]Here and throughout, we use the notation $\tilde{O}(f)$ to hide polylogarithmic dependencies on the argument, i. e., for expressions of the form $O(f \log^c f)$ (for some absolute constant c).

Testing equivalence. The *equivalence testing* problem has been extensively studied over the past decade, and its sample complexity is now known to be $\Theta(\max(n^{2/3}/\varepsilon^{4/3}, \sqrt{n}/\varepsilon^2))$ in the sampling model [10, 46, 18].

In the COND setting, Canonne, Ron, and Servedio showed that equivalence testing is possible with only $\operatorname{poly}(\log n, 1/\varepsilon)$ queries. Concurrent to our work, Falahatgar et al. [24] brought this upper bound down to $\tilde{O}\big((\log\log n)/\varepsilon^5\big)$, a *doubly exponential* improvement over the $n^{\Omega(1)}$ samples needed in the standard sampling model. However, these results still left open the possibility of a constant query complexity—given that both uniformity and identity testing admit constant-query testers, it is natural to wonder where equivalence testing lies.[4] This question was posed by Fischer at the Bertinoro Workshop on Sublinear Algorithms 2014 [Sublinear.info, Problem 66]. We make decisive progress in answering it, ruling out the possibility of any constant-query tester for equivalence. Along with the upper bound of Falahatgar et al. [24], our results nearly settle the dependence on the domain size, showing that $(\log\log n)^{\Theta(1)}$ samples are both necessary and sufficient.

Support-size estimation. Raskhodnikova et al. [34] showed that obtaining *additive* estimates of the support size requires sample complexity almost linear in n. Subsequent work by Valiant and Valiant [44, 42] settles the question, establishing that $\Theta(n/\log n)$ samples are both necessary and sufficient. Note that the proof of the Valiants' lower bound translates to multiplicative approximations as well, as it hinges on the hardness of distinguishing a distribution with support $s \le n$ from a distribution with support $s + \varepsilon n \ge (1+\varepsilon)s$. To the best of our knowledge, the question of getting a multiplicative-factor estimate of the support size of a distribution given conditional sampling access has not been previously considered. We provide upper and lower bounds for both the adaptive and non-adaptive versions of this problem.

1.3 Our results

We make significant progress in each of the problems introduced in the previous section, yielding a better understanding of their query complexities. We prove four results pertaining to the sample complexity of equivalence testing, support-size estimation, and uniformity testing in the COND framework.

Our main result gives a lower bound on the sample complexity of testing equivalence with adaptive queries under the COND model, resolving in the negative the question of whether constant-query complexity was achievable [Sublinear.info, Problem 66].

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) $d_{\mathrm{TV}}(D_1, D_2) \ge 1/4$, must have query complexity $\Omega\big(\sqrt{\log\log n}\big)$.*

Combined with the recent $\tilde{O}(\log\log n)$ upper bound of Falahatgar et al. [24], this almost settles the sample complexity of this question. Furthermore, as the related task of identity testing *can* be performed with a constant number of queries in the conditional sampling model, this demonstrates an intriguing and intrinsic difference between the two problems. Our result also shows an interesting contrast with the usual sampling model, where both identity and equivalence testing have polynomial sample complexity.

[4] 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 [14]).

Next, we establish a logarithmic lower bound on *non-adaptive* support-size estimation, for any (large enough) constant factor. This improves on the result of Chakraborty et al. [16], which gave a doubly logarithmic lower bound for constant factor support-size estimation.

Theorem 1.2 (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 $\gamma \geq \sqrt{2}$ must have query complexity $\Omega\big((\log n)/\log^2 \gamma\big)$.*

Moreover, the approach used to prove this theorem also implies an analogous lower bound on *non-adaptive* uniformity testing in the conditional model, answering a conjecture of Chakraborty et al. [16].

Theorem 1.3 (Non-Adaptive Uniformity Testing)**.** *Any non-adaptive algorithm which, given* COND *access to an unknown distribution D on $[n]$ and parameter $\varepsilon \in (0, 1/4)$, distinguishes with probability at least $2/3$ between (a) $D = \mathcal{U}_{[n]}$ and (b) $\mathrm{d_{TV}}(D, \mathcal{U}_{[n]}) \geq \varepsilon$, must have query complexity $\Omega((\log n)/\varepsilon)$.*

We note that these results complement polylog(n)-query upper bounds on non-adaptive support-size estimation and uniformity testing, the former of which we sketch in this paper, and the latter obtained by Chakraborty et al. [16]. This shows that both of these problems have query complexity $\log^{\Theta(1)} n$ in the non-adaptive case.

Finally, we conclude with an upper bound for *adaptive* support-size estimation. Specifically, we provide a $\tilde{O}(\log\log n)$-query algorithm for support-size estimation. This shows that the question becomes *double exponentially* easier when conditional samples are allowed.

Theorem 1.4 (Adaptive Support-Size Estimation)**.** *Let $\tau > 0$ be any constant. There exists an adaptive algorithm which, given* COND *access to an unknown distribution D on $[n]$ (guaranteed to have probability mass at least τ/n on every element of its support) and accuracy parameter $\varepsilon \in (0, 1)$, makes $\tilde{O}\big((\log\log n)/\varepsilon^3\big)$ queries to the oracle*[5] *and returns a value $\tilde{\omega}$ such that the following holds. With probability at least $2/3$, $\tilde{\omega} \in [(1/(1+\varepsilon)) \cdot \omega, (1+\varepsilon) \cdot \omega]$, where $\omega = |\mathrm{supp}(D)|$.*

1.3.1 Relation to the Ron-Tsur model

Recent work of Ron and Tsur [37] studies a model which is slightly different—and more favorable to the—than ours. In their setting, the algorithm still performs queries consisting of a subset of the domain, as in our case. However, the algorithm is also given the promise that the distribution is uniform on a subset of the domain, and whenever a query set contains 0 probability mass the oracle explicitly indicates this is the case. Their paper provides a number of results for support-size estimation in this model.

We point out two connections between our work and theirs. First, our $\Omega(\log n)$ lower bound for non-adaptive support-size estimation (Theorem 1.2) holds in the model of Ron and Tsur. Although lower bounds in the conditional sampling setting do not apply directly to their model, our construction and analysis do carry over, and provide a nearly tight answer to a question left unanswered in their paper. Also, our $\tilde{O}(\log\log n)$-query algorithm for adaptive support-size estimation (Theorem 1.4) can be seen as generalizing their result to the weaker conditional sampling model (most significantly, when we are not given the promise that the distribution be uniform).

[5] We remark that the constant in the $\tilde{O}$ depends polynomially on $1/\tau$.

Problem	COND model	Standard model
Testing equivalence	$\tilde{O}\left(\frac{\log\log n}{\varepsilon^5}\right)$ [24] $\Omega\left(\sqrt{\log\log n}\right)$ (Theorem 1.1)	$\Theta\left(\max\left(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{n^{1/2}}{\varepsilon^2}\right)\right)$ [18]
Estimating support size (adaptive)	$\tilde{O}\left(\frac{\log\log n}{\varepsilon^3}\right)$ (Theorem 1.4) $\Omega\left(\sqrt{\log\log n}\right)$ [16] (†)	$\Theta\left(\frac{n}{\log n}\right)$ [42]
Estimating support size (non-adaptive)	$O(\mathrm{poly}(\log n, 1/\varepsilon))$ (Section 5.4) $\Omega(\log n)$ (Theorem 1.2)	
Testing uniformity (non-adaptive)	$\tilde{O}\left(\frac{\log^5 n}{\varepsilon^6}\right)$ [16] $\Omega\left(\frac{\log n}{\varepsilon}\right)$ (Theorem 1.3)	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [33]

Table 1: Summary of results. Note that the lower bound (†) can also be easily derived from our lower bound on testing equivalence.

1.4 Techniques and proof ideas

Lower bound on adaptive equivalence testing. In order to prove our main lower bound, Theorem 1.1, 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. [16] and focus on a restricted class of algorithms they introduce, called "core adaptive testers" (see Section 2.2 for a formal definition). They 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 remains 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, 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 Minimax 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 previously observed. If this occurs, it is similarly predictable *which* element the query will return. On the other hand, if a fresh element is observed, 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 amount of information gleaned from a single sample is minimal.

At a lower level, this process itself is reminiscent of the "hard" instances underlying the lower bound of Canonne, Ron, and Servedio [14] for testing identity (with a PAIRCOND oracle), with one pivotal twist. As in their work, 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 3.1 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 3.2) this random size of the support, the sequences of samples it will obtain under D_1 and D_2 are almost identically distributed. The above discussion crucially hides many significant aspects and technical difficulties which we address in Section 3. Moreover, we observe that the lower bound we obtain seems to be optimal with regard to our proofs techniques (specifically, to the decision tree approach), and not an artifact of our lower bound instances. Namely, there appear to be conceptual barriers to strengthening our result, which would require new ideas.

Lower bound on non-adaptive support-size estimation. Turning to the (non-adaptive) lower bound of Theorem 1.2, we define two families of distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, where an instance is either a draw (D_1, D_2) from $\mathcal{D}_1 \times \mathcal{D}_2$, or simply (D_1, D_1). Any distribution in $\mathcal{D}_2$ has support size γ times that of its corresponding distribution in $\mathcal{D}_1$. Yet, we argue that no non-adaptive *deterministic* tester making too few queries can distinguish between these two cases, as the tuple of samples it will obtain from D_1 or (the corresponding) D_2 is almost identically distributed (where the randomness is over the choice of the instance 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 "large" ones (where the query set S is so big compared to the support T that a uniform sample from $S \cap T$ is essentially indistinguishable from a uniform sample from S). We conclude the proof by invoking Yao's Principle, carrying the lower bound back to the setting of non-adaptive *randomized* testers.

Interestingly, this argument essentially gives us Theorem 1.3 "for free." Indeed, the big-query-set case above is handled by proving that the distribution of samples returned on those queries is indistinguishable, both for $\mathcal{D}_1$ and $\mathcal{D}_2$, from samples obtained from the *actual* uniform distribution. Considering again the small-query-set case separately, this allows us to argue that a random distribution from (say) $\mathcal{D}_1$ is indistinguishable from uniform.

Upper bound on support-size estimation. Our algorithm for estimating the support size within a constant factor (Theorem 1.4) 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, $\{2^{j-1}, \ldots, 2^j\}$. Not surprisingly, our algorithm draws on similar ideas as in [37, 41], 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 improvement. Indeed, the lower bound construction we consider (see Section 3.1) 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 further shows that the adaptive lower bound for support-size estimation of Chakraborty et al. [16] is also tight up to a quadratic improvement.

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 covers our main result on adaptive equivalence testing, Theorem 1.1. In Section 4 we prove our lower bounds for support-size estimation and uniformity testing, and Section 5 details our upper bounds for support-size estimation. The corresponding sections may be read independently.

2 Preliminaries

2.1 Notation and sampling models

All throughout this paper, we denote by $[n]$ the set $\{1,\dots,n\}$, 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 $D_S(x) = D(x)/D(S)$ for $x \in S$ and $D_S(x) = 0$ otherwise.

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

$$\mathrm{d_{TV}}(D_1,D_2) \stackrel{\text{def}}{=} \frac{1}{2}\|D_1 - D_2\|_1 = \frac{1}{2}\sum_{x\in[n]} |D_1(i) - D_2(i)| = \max_{S\subseteq[n]}(D_1(S) - D_2(S)) \tag{2.1}$$

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

In this work, we focus on the setting of *conditional access* to the distribution, as introduced and studied in [16, 14]. 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 choices do not make a significant difference in most situations, as adaptive algorithms can include in their next

[5]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.

queries a sample previously 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. Furthermore, as in [14, 16] we do not take the *complexity* of specifying the set S to the oracle into account, and indeed allow arbitrary sets as queries.[6]

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 returns ACCEPT or REJECT, such that the following holds:

- if $D \in \mathcal{P}$, $\mathcal{T}$ returns ACCEPT with probability at least $2/3$;
- if $\mathrm{d_{TV}}(D,\mathcal{P}) \geq \varepsilon$, $\mathcal{T}$ returns 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 3, 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. [16]. These ideas, for the case of *label-invariant* properties[7] 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. [16] 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 atoms, denoted $\mathrm{At}(\mathcal{A})$, is the *partition* generated by $\mathcal{A}$.

[6]We further observe that, besides the general COND_D oracle which allows these arbitrary query sets, the authors of [14] introduce two weaker variants of the conditional model (the "pair-cond" $\mathsf{PAIRCOND}_D$ and "interval-cond" $\mathsf{INTCOND}_D$ oracles) which restrict algorithms to "simple" queries.

[7]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. $(D_1\circ\sigma, D_2\circ\sigma) \in \mathcal{P}$).

This definition essentially captures "all sets (besides the A_i) about which something can be learned 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 $\mathbf{s} = ((s_j^{(1)},s_j^{(2)}))_{1\le j\le t}$ as above, the *t-configuration of* $\mathbf{s}$ consists of the $6t^2$ bits indicating, for all $1 \le i, j \le t$, 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 mentioned before 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.

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 $s_1^{(k)},\ldots,s_{i-1}^{(k)}$ will be included in the next query. The labels of these samples are unknown, but are indexed by the index of the query which returned them.)
- 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
  $$\left\{ \Lambda \subseteq [n] \setminus \{s_j^{(1)},s_j^{(2)}\}_{1\le j\le i-1} \,:\, \forall A \in \mathrm{At}(A_1,\ldots,A_{i-1}),\ |\Lambda \cap A| = k_i^A \right\}. \tag{2.2}$$
  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:

$$\Gamma_i \stackrel{\text{def}}{=} \left\{ s_j^{(1)} \,:\, j \in K_i^{(1)} \right\} \cup \left\{ s_j^{(2)} \,:\, j \in K_i^{(2)} \right\} ; \qquad A_i \stackrel{\text{def}}{=} \Lambda_i \cup \Gamma_i \,. \tag{2.3}$$

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 $A_1,\dots,A_i$ and $(s_1^{(1)},s_1^{(2)}),\dots,(s_i^{(1)},s_i^{(2)})$, which is revealed to $\mathcal{T}$. Put differently, the algorithm only learns (i) to which of the A_ℓ 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, $\mathcal{T}$ returns either ACCEPT or REJECT, based only on the configuration of $A_1,\dots,A_t$ and $(s_1^{(1)},s_1^{(2)}),\dots,(s_t^{(1)},s_t^{(2)})$ (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 [31]), 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 lower-bounded 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 3 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 [31]).

2.4 Chernoff bounds for binomials and hypergeometrics

We will make extensive use of Chernoff-style bounds in this article. Recall that the $\mathrm{Binomial}(n,p)$ distribution describes the distribution of the number of successes when we run n independent Bernoulli trials, each with success probability p.

Lemma 2.6 (Chernoff Bound for Binomials)**.** *Let $X \sim \mathrm{Binomial}(n,p)$ and $\mu = \mathbb{E}[X] = np$. Then*

$$\forall \delta \in (0,1), \qquad \Pr[\,|X-\mu| \ge \delta\mu\,] \le 2\exp\left(-\frac{\delta^2\mu}{3}\right).$$

We will also need a similar Chernoff-style bound for the *hypergeometric* distribution. The $\mathrm{Hypergeometric}(n,K,N)$ distribution describes the distribution of the number of successes when we draw n times without replacement from a population of size N, in which K objects have the pertinent feature (and thus count as successes). Note that if the drawing were done with replacement, and $K/N = p$, then this would be equivalent to $\mathrm{Binomial}(n,p)$. Sampling without replacement introduces *negative correlation* between the probability of each draw being successful. This type of negative correlation generally "helps" with concentration, allowing one to prove similar concentration bounds (see, e. g., [19, 23], Theorem 1.17 of [6]).

Lemma 2.7 (Chernoff Bound for Hypergeometrics)**.** *Let $X \sim \mathrm{Hypergeometric}(n,K,N)$ and $\mu = \mathbb{E}[X] = nK/N$. Then,*

$$\forall \delta \in (0,1), \qquad \Pr[\,|X-\mu| \ge \delta\mu\,] \le 2\exp\left(-\frac{\delta^2\mu}{3}\right).$$

3 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 $\mathrm{d_{TV}}(D_1,D_2) \ge 1/4$. We then show that no algorithm $\mathcal{T}$ making $O\big(\sqrt{\log\log n}\big)$ 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 3.1, and provide a detailed analysis in Section 3.2.

3.1 Construction

We now summarize how a pair of distribution is constructed under $\mathcal{Y}$ and $\mathcal{N}$. (Each specific step will be described in more detail in the subsequent paragraphs.)

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) Choose ρ and r such that $\sum_{i=1}^{2r} \rho^i = n^{1/4}$.

   (b) Divide $\{1,\dots,m\}$ into intervals $B_1,\dots,B_{2r}$ with $|B_i| = b \cdot \rho^i$.

3. **Distributions**

   (a) For each $i \in [2r]$, assign probability mass $1/(2r)$ uniformly over B_i to generate distribution D_1.

   (b) For each $i \in [r]$ independently, pick π_i to be a Bernoulli trial with $\Pr(\pi_i = 0) = 1/2$; if $\pi_i = 0$ then assign probability mass $1/(4r)$ and $3/(4r)$ over B_{2i-1} and B_{2i}, respectively, else $3/(4r)$ and $1/(4r)$, 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.

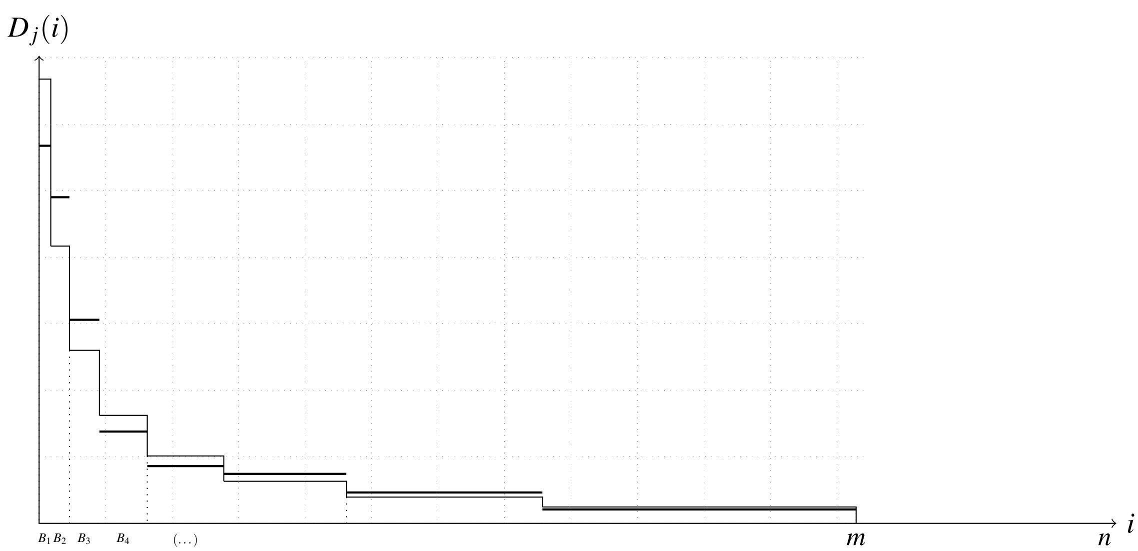

Figure 1: A no-instance (D_1, D_2) (before permutation).

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 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. The test would (roughly) go as follows. Since $|B_i|$ is known, one can choose a random subset S of the domain which (with high probability) has no intersection with B_i for $i \leq 2r-2$, a $O(1)$ size intersection with B_{2r-1}, and a $O(\rho)$ size intersection with B_{2r}. Perform $O(1)$ conditional queries over the set S, for both distributions. Given these queries, we can then identify which elements of S belong to B_{2r-1} or B_{2r}—namely, those which occur at most once belong to B_{2r}, and those which occur at least twice belong to B_{2r-1}. In a $\mathcal{Y}$ instance, then in both distributions, a $1/2$ fraction of queries will belong to B_{2r-1}, whereas in a $\mathcal{N}$ instance, one distribution will have either a $1/4$ or $3/4$ fraction of queries in B_{2r-1}, allowing us to distinguish the two cases.

Buckets. Our construction is inspired by the lower bound of Canonne, Ron, and Servedio [14, Theorem 8] for the more restrictive PAIRCOND access model. We partition the support into $2r$ consecutive intervals (henceforth referred to as *buckets*) $B_1,\dots,B_{2r}$, 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 $1/2r$ to each bucket. Therefore, points within B_i have the same probability mass $1/2rb\rho^i$. 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* $\Pi_1,\dots,\Pi_r$, 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 $1/4r$ while the other gets $3/4r$ (so that the probability of every point is either decreased by a factor $1/2$ or increased by $3/2$). 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 the π_i.

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 $(\pi_1,\dots,\pi_r)\in\{0,1\}^r$ (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, $\vec{\pi}$ 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 $r = \log n/(8\log\rho) + O(1)$, 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 $r = (1/8)\sqrt{\log n} + O(1)$. (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.)

3.2 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 3.2.1, 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 3.2.2, 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 3.1. *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 $q = (1/10)\sqrt{\log\log n}$ (adaptive) queries; and will show that it can only distinguish between yes and no-instances with probability $o(1)$.

3.2.1 Banning "bad queries"

As mentioned in Section 3.1, 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 [16], 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 3.2. *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}—in other words, the set is likely to be disjoint from the support. 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, is either disjoint from B_i or has an intersection with B_i of size close to the expected value. In the latter 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 3.1, 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 3.3. Let q be an integer, and let $\varphi = \Theta(q^{5/2})$. A number β is said to be *small* if $\beta < n/(b\rho^{2r})$; it is *large* (with relation to some integer q) if $\beta \geq n/(b\rho^{2r-2\varphi})$.

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 $\beta \left|B_{2r-2\varphi}\right| / n \geq 1$). 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 3.4. *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 $\alpha = \mathrm{poly}(q)$.)

Definition 3.5. 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 3.6. *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 3.7. 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 $1/2rb\rho^i$ of a single element from any of the $2r$ buckets.

Proposition 3.8. *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

$$\sum_{j=1}^{\ell}\beta_j\Delta_j b=b\sum_{j=1}^{\ell}\beta_j(\lambda_j+O(1))+\frac{b\log b}{\log\rho}\sum_{j=1}^{\ell}\beta_j\,. \tag{3.1}$$

- 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 equation (3.1) above, requiring that

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

  Recalling that $b\log b=k_b2^{k_b}$, 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

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

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

- Finally, the two terms are comparable only if

$$\log b=\Theta\left(\log\rho\cdot\sum_{j=1}^{\ell}\beta_j(\lambda_j+O(1))\cdot\left(\sum_{j=1}^{\ell}\beta_j\right)^{-1}\right);$$

  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 3.9. *With probability at least*

$$1 - O\left(\frac{2^{6q^2+q}(r\log\alpha + \varphi\log\rho) + 2^{6q^2}(r\log q)}{\log n}\right),$$

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,*[8] *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 3.2, 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 Proposition 3.4 and Proposition 3.6 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 3.8 and applying the union bound over all 2^{6q^2+1} nodes in the tree (each containing a single k_i^A vector). □

3.2.2 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.[9] $\alpha \stackrel{\text{def}}{=} q^7$, $\varphi \stackrel{\text{def}}{=} q^{5/2}$ and $\gamma \stackrel{\text{def}}{=} 1/\varphi = q^{-5/2}$. (Recall further that $q = (1/10)\sqrt{\log\log n}$.)

Lemma 3.10. *Conditioned on the events of Lemma 3.9, 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 $o(1)$.*

Proof. This proof is by induction on $1 \le i \le q$. 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

[8] 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.

[9] 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.

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 $|S'|b\rho^{\ell}/n \le 1/\alpha$, 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 $j \le \ell^{S'}$, $\left|S' \cap B_j\right| = 0$;
  - for $j > \ell^{S'}$, $\left|S' \cap B_j\right|$ 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

  $$\frac{p_1}{p_1+p_2} \le O\left(\frac{1}{q^2}\right) \qquad \text{or} \qquad \frac{p_2}{p_1+p_2} \le O\left(\frac{1}{q^2}\right)$$

  (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[10] 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

  $$\mathrm{d_{TV}}\big(T_i^{\mathsf{yes}}, T_i^{\mathsf{no}}\big) \le O\big(1/q^2 + 1/\rho + \gamma + 1/\varphi\big) = o(1)\,.$$

We will show that $\boldsymbol{E_1}(i)$ holds with probability $1 - O\big(2^i \exp(-2\gamma^2\alpha/3)\big)$ 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, which $\boldsymbol{E_3}(q)$ bounds. We can then take a union bound over all $i \in [q]$ to upper bound the probability that $\boldsymbol{E_1}(i)$ and $\boldsymbol{E_2}(i)$ do not hold, and use $\boldsymbol{E_3}(i)$ and the coupling interpretation of total variation distance to upper bound the probability that T_i^{yes} and T_i^{no} ever differ. Any of these “failure” events happens with probability at most

$$O\left(2^q \exp\left(-\frac{2\gamma^2\alpha}{3}\right) + \frac{q^2}{\varphi} + \frac{1}{q} + \frac{q}{\rho} + q\gamma + \frac{q}{\varphi}\right) = o(1)$$

[10]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.

(from our choice of α, γ, φ). This upper bounds the total variation distance between T^{yes} and T^{no}, 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.

Lemma 3.11. *Assume 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 $\boldsymbol{E_1}(i)$ (the statement prior to "Furthermore"). Let S (and the corresponding S') be any atom as in $\boldsymbol{E_1}(i)$. First, we note that $\ell^{S'} \in \{0, \ldots, 2r - \varphi - 2\} \cup \{2r\}$ since we are conditioning on Lemma 3.9: $|S'|$ is α-stable and either large or small, which enforces this condition. Next, suppose S' is contained in some other atom T generated by $A_1, \ldots, A_{i-1}$, and let

$$T' = T \setminus \{s_1^{(1)}, s_1^{(2)}, \ldots, s_{i-1}^{(1)}, s_{i-1}^{(2)}\}.$$

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, $\left|T' \cap B_j\right| = 0$, so $\left|S' \cap B_j\right| = 0$ with probability 1.

- The next case is $\ell^{T'} < j \le \ell^{S'}$. Recall from the definition of a core adaptive tester that S' will be chosen uniformly at random from all subsets of T' of appropriate size. By the inductive hypothesis,

  $$\frac{\left|T' \cap B_j\right|}{|T'|} \in [1-(i-1)\gamma, 1+(i-1)\gamma]\,\frac{b\rho^j}{n},$$

  and therefore

  $$\mathbb{E}\left[\left|S' \cap B_j\right|\right] \in [1-(i-1)\gamma, 1+(i-1)\gamma]\,\frac{|S'|\,b\rho^j}{n}, \quad \text{implying} \quad \mathbb{E}\left[\left|S' \cap B_j\right|\right] \le \frac{2}{\alpha\rho^{\ell^{S'}-j}};$$

  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 hypergeometric random variables (Lemma 2.7), and writing $\mu \stackrel{\text{def}}{=} \mathbb{E}\left[\left|S' \cap B_j\right|\right]$ for conciseness,

  $$\begin{aligned}\Pr\left[\left|S' \cap B_j\right| \ge 1\right] &= \Pr\left[\left|S' \cap B_j\right| \ge \left(1 + \frac{1-\mu}{\mu}\right)\mu\right] \\ &\le \exp\left(-\frac{(1-\mu)^2}{3\mu}\right) \\ &\le \exp\left(-\frac{1}{12}\alpha\rho^{\ell^{S'}-j}\right),\end{aligned}$$

  where the second inequality holds because $\mu \le 2/\alpha\rho^{\ell^{S'}-j}$ and $(1-\mu)^2 \ge 1/2$ for n sufficiently large.

- The final case is $j > \ell^{S'}$. As in the previous one,

$$\mathbb{E}\left[\left|S' \cap B_j\right|\right] \in [1-(i-1)\gamma, 1+(i-1)\gamma]\,\frac{|S'|\,b\rho^j}{n}, \quad \text{implying} \quad \mathbb{E}\left[\left|S' \cap B_j\right|\right] \geq \frac{\alpha\rho^{j-\ell^{S'}-1}}{2};$$

  where the inequality is by the definition of $\ell^{S'}$, α-stability, and using the fact that $(i-1)\gamma \leq 1/2$. Using again a Chernoff bound for hypergeometric random variables (Lemma 2.7),

$$\begin{aligned}
\Pr\left[\left|S' \cap B_j\right| - \mathbb{E}\left[\left|S' \cap B_j\right|\right] \geq \gamma\frac{|S'|\,b\rho^j}{n}\right] &\leq \Pr\left[\left|S' \cap B_j\right| - \mathbb{E}\left[\left|S' \cap B_j\right|\right] \geq \gamma 2\mathbb{E}\left[\left|S' \cap B_j\right|\right]\right] \\
&\leq 2\exp\left(-\frac{(2\gamma)^2\mathbb{E}\left[\left|S' \cap B_j\right|\right]}{3}\right) \\
&\leq 2\exp\left(-\frac{2}{3}\gamma^2\alpha\rho^{j-\ell^{S'}-1}\right),
\end{aligned}$$

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

Since we wish to prove the statement for all buckets B_j simultaneously, we take a union bound over all j. Recall that we want to bound the probability that S' satisfies two conditions, for $j \leq \ell^{S'}$, $\left|S' \cap B_j\right| = 0$; and for $j > \ell^{S'}$, $\left|S' \cap B_j\right|$ lies in

$$[1-i\gamma, 1+i\gamma]\,\frac{|S'|\,b\rho^j}{n}.$$

Using bounds from all three of these regimes of j, and a union bound, the probability that S' does not satisfy the conditions of $\boldsymbol{E_1}(i)$ 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 remainder of $\boldsymbol{E_1}(i)$ (the statement following "Furthermore"); this will follow from the definition of incomparability (Definition 3.7).

- 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

$$p_2 \le p_2' + \frac{3p_2'}{\rho} + \frac{3p_2'}{\rho^3} + \cdots \le p_2' + \frac{3p_2'}{\rho} \sum_{k=0}^{\infty} \frac{1}{\rho^{2k}} = \left(1 + \frac{3}{\rho}\frac{\rho^2}{\rho^2 - 1}\right) p_2'$$
$$\le (1+o(1))p_2' .$$

Therefore, $p_2 \in [p_2', (1+o(1))p_2']$. Supposing this heaviest element belongs to bucket j, we can say that

$$p_2 \in \left[\frac{1}{2}, (1+o(1))\frac{3}{2}\right] \frac{1}{2rb\rho^j} .$$

- 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 $|A' \cap B_k| \in [1 - i\gamma, 1 + i\gamma]\,|A'|\, b\rho^k/n$. In the latter case, noting that $i\gamma \le 1/2$ 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 3.7, as shown earlier in this proof, this non-empty intersection happens for exactly Δ_A buckets. 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 3.9 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 Lemma 3.11. □

Lemma 3.12. *Assume that $\boldsymbol{E_1}(t), \boldsymbol{E_2}(t), \boldsymbol{E_3}(t)$ hold for all $1 \le t \le i-1$, and additionally $\boldsymbol{E_1}(i)$ holds. Then $\boldsymbol{E_2}(i)$ holds with probability at least $1 - O(i/\varphi)$.*

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 let us condition on the fact 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 $(1+i\gamma)(2i-1)|S|/rn$. Similarly, using the definition of a large atom, the minimum probability mass contained in S is $(1-i\gamma)\varphi|S|/rn$. 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 which large atom was arbitrary, we can remove the conditioning. Taking a union bound for $s_i^{(1)}$ and $s_i^{(2)}$ gives the result. □

Lemma 3.13. *Assume that $\boldsymbol{E_1}(t), \boldsymbol{E_2}(t), \boldsymbol{E_3}(t)$ hold for all $1 \le t \le i-1$, and additionally $\boldsymbol{E_1}(i)$ holds. Then $\boldsymbol{E_3}(i)$ holds.*

Proof. We fix some setting of the intersection history, 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 happens with probability $1 - O\big(1/q^2\big)$.

Proposition 3.14. *In either a yes-instance or a no-instance, s_i will either come from Λ_i with probability $1 - O\big(1/q^2\big)$, or Γ_i with probability $1 - O\big(1/q^2\big)$, 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." □

We now try to bound the total variation distance between T_i^{yes} and T_i^{no} conditioning on this event. In the case when it does not hold, we trivially bound the total variation distance by 1, incurring a cost of $O\big(1/q^2\big)$ to the total variation distance between the unconditioned variables. Since our target was for this quantity was $O\big(1/q^2 + 1/\rho + \gamma + 1/\varphi\big)$, it remains to show, in the conditioned space, that the total variation distance in either case is at most $O(1/\rho + \gamma + 1/\varphi) = O(1/q^{5/2})$. 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 3.15. *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 $1 - O(1/\rho)$, 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

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

Since the ratio of this value and p is $O(1/\rho)$, with probability $1-O(1/\rho)$ the sample returned is the heaviest element in Γ_i. ☐

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

Proposition 3.16. *Condition that s_i comes from Λ_i. Then either:*

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

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

$$\begin{aligned}\mathrm{d_{TV}}(T_i^{\mathsf{yes}},T_i^{\mathsf{no}})&\leq\mathrm{d_{TV}}(T_i^{\mathsf{yes}},T_i^{\mathsf{no}}\mid\mathcal{X})\Pr(\mathcal{X})+\mathrm{d_{TV}}(T_i^{\mathsf{yes}},T_i^{\mathsf{no}}\mid\bar{\mathcal{X}})\Pr(\bar{\mathcal{X}})\\&\leq\mathrm{d_{TV}}(T_i^{\mathsf{yes}},T_i^{\mathsf{no}}\mid\mathcal{X})+O(1/\varphi),\end{aligned}\tag{3.2}$$

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

$$N\stackrel{\text{def}}{=}\sum_{A\in\mathrm{At}(A_1,\dots,A_i)}N^A.$$

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 $O(\gamma N^A/N)$. Summing over all atoms, we get the desired result of $O(\gamma)$.

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

$$(1+\gamma)\frac{|A|}{rn}\left((1-\gamma)\frac{|A|}{rn}N^A\right)^{-1},$$

and the minimum is

$$(1-\gamma)\frac{|A|}{rn}\left((1+\gamma)\frac{|A|}{rn}N^A\right)^{-1},$$

so the total variation distance in the distribution incurred by this bucket-pair is at most $O(\gamma/N^A)$. Summing this difference over all N^A bucket-pairs, we get $2\gamma/(1-\gamma^2) = O(\gamma)$. 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. This concludes the proof of Proposition 3.16. □

We note that in both cases, the cost in total variation distance which is incurred is $O(1/\rho + \gamma + 1/\varphi)$, which implies $\boldsymbol{E}_3(i)$—proving Lemma 3.13. □

This concludes the proof of Lemma 3.10. □

With Lemma 3.10 in hand, the proof of the main theorem is straightforward.

Proof of Theorem 1.1. Conditioned on Lemma 3.9, Lemma 3.10 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 3.9 occurs with probability $1-o(1)$, the difference between the unconditional probabilities is also $o(1)$. □

4 Lower bounds for non-adaptive algorithms

In this section, we prove our lower bounds for non-adaptive support-size estimation and uniformity testing, restated here.

Theorem 1.2 (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 $\gamma \geq \sqrt{2}$ must have query complexity $\Omega\big((\log n)/\log^2\gamma\big)$.*

Theorem 1.3 (Non-Adaptive Uniformity Testing)**.** *Any non-adaptive algorithm which, given* COND *access to an unknown distribution D on $[n]$ and parameter $\varepsilon \in (0,1/4)$, distinguishes with probability at least $2/3$ between* (a) *$D = \mathcal{U}_{[n]}$ and* (b) *$\mathrm{d_{TV}}(D,\mathcal{U}_{[n]}) \geq \varepsilon$, must have query complexity $\Omega((\log n)/\varepsilon)$.*

These two theorems follow from the same argument, which we outline below before turning to the proof itself. (Note that we will, in this section, establish Theorem 1.3 for *constant* ε, i. e., $\varepsilon = 1/4$; before explaining in Section 4.1 how to derive the general statement as a corollary.)

Structure of the proof. By Yao's Minimax Principle, we consider deterministic tests and study their performance over random distributions, chosen to be uniform over a random subset of carefully picked size. The proof of Theorem 1.2 then proceeds in 3 steps: in Lemma 4.2, we first argue that all queries made by the deterministic tester will (with high probability over the choice of the support size s) behave very "nicely" with regard to s, i. e., not be concentrated around it. Then, we condition on this to bound the total variation distance between the sequence of samples obtained in the two cases we "oppose," a random distribution from a family $\mathcal{D}_1$ and the corresponding one from a family $\mathcal{D}_2$. In Lemma 4.4 we show that the part of total variation distance due to samples from the small queries is zero, except with probability $o(1)$ over the choice of s. Similarly, Lemma 4.4 allows us to say (comparing both cases to a third "reference" case, a honest-to-goodness uniform distribution over the whole domain, and applying a triangle inequality) that the remaining part of the total variation distance due to samples from the big queries is zero as well, except again with probability $o(1)$. Combining these three lets us to conclude by properties of the total variation distance, as (since the queries are non-adaptive) the distribution over the sequence of samples is a product distribution. (Moreover, applying Lemma 4.4 as a stand-alone enables us, with little additional work, to obtain Theorem 1.3, as our argument in particular implies distributions from $\mathcal{D}_1$ are hard to distinguish from uniform.)

The families $\mathcal{D}_1$ and $\mathcal{D}_2$. Fix $\gamma \geq \sqrt{2}$ as in Theorem 1.2; writing $\beta \stackrel{\text{def}}{=} \gamma^2$, we define the set

$$\mathcal{S} \stackrel{\text{def}}{=} \left\{ \beta^k n^{1/4} \,:\, 0 \leq k \leq \frac{\log n}{2\log\beta} \right\} = \{n^{1/4}, \beta n^{1/4}, \beta^2 n^{1/4}, \dots, , n^{3/4}\}. \tag{4.1}$$

A no-instance $(D_1,D_2) \in \mathcal{D}_1 \times \mathcal{D}_2$ is then obtained by the following process:

- Draw s uniformly at random from $\mathcal{S}$.
- Pick a uniformly random set $S_1 \subseteq [n]$ of size s, and set D_1 to be uniform on S_1.
- Pick a uniformly random set $S_2 \subseteq [n]$ of size βs, and set D_2 to be uniform on S_2.

(Similarly, a yes-instance is obtained by first drawing a no-instance (D_1,D_2), and discarding D_2 to keep only $(D_1,D_1) \in \mathcal{D}_1 \times \mathcal{D}_1$.)

We will argue that no algorithm can distinguish with high probability between the cases $(D_1,D_2) \sim \mathcal{D}_1 \times \mathcal{D}_2$ and $(D_1,D_2) \sim \mathcal{D}_1 \times \mathcal{D}_1$, by showing that in both cases D_1 and D_2 generate transcripts indistinguishable from those the *uniform* distribution $\mathcal{U}_{[n]}$ would yield. This will imply Theorem 1.2, as any algorithm to estimate the support within a multiplicative γ would imply a distinguisher between instances of the form (D_1, D_1) and (D_1, D_2) (indeed, the support sizes of D_1 and D_2 differ by a factor $\beta = \gamma^2$). (As for Theorem 1.3, observe that any distribution $D_1 \in \mathcal{D}_1$ has constant distance from the uniform distribution on $[n]$, so that a uniformity tester must be able to tell D_1 apart from $\mathcal{U}_{[n]}$.)

Small and big query sets. Let $\mathcal{T}$ be any deterministic non-adaptive algorithm making

$$q_{\mathcal{T}} \leq q = \frac{1}{80000} \frac{\log n}{\log^2 \beta}$$

queries. Without loss of generality, we can assume $\mathcal{T}$ makes exactly q queries, and denote them by $A_1, \ldots, A_q \subseteq [n]$. Moreover, we let $a_i = |A_i|$, and (again without loss of generality) write $a_1 \geq \cdots \geq a_q$.

As a preliminary observation, note that for any $A \subset [n]$ and $0 \leq s \leq n$ we have

$$\mathbb{E}_S |S \cap A| = \frac{|A| s}{n}$$

where the expectation is over a uniform choice of S among all $\binom{n}{s}$ subsets of size s. This observation will lead us to divide the query sets A_i in two groups, depending on the expected size of their intersection with the (random) support.

With this in mind, the following definition will be crucial to our proof. Intuitively, it captures the distribution of *sizes of intersection* of various query sets with the randomly chosen set S.

Definition 4.1. Let $s \geq 1$, and $\mathcal{A} = \{a_1, \ldots, a_q\}$ be any set of q integers. For a real number $t > 0$, define

$$C_t(s) \stackrel{\text{def}}{=} \left| \left\{ i \in [q] : \frac{a_i s}{n} \in \left(\beta^{-t}, \beta^t\right) \right\} \right| \tag{4.2}$$

to be the number of *t-hit points of $\mathcal{A}$ (for s)*.

The next result will be crucial to prove our lower bounds: it roughly states that if we consider the a_i and scale them by the random quantity s/n, then the distribution of the random variable generated has an exponential tail with respect to t.

Lemma 4.2 (Hitting Lemma)**.** *Fix $\mathcal{A}$ as in the previous definition. If s is drawn uniformly at random from $\mathcal{S}$, then with probability at least* $99/100$.

$$\sup_{t>0} \frac{C_t(s)}{t} < \frac{2}{100}. \tag{4.3}$$

Proof. Without loss of generality, assume that the a_i are in decreasing order. We will work in the logarithmic domain: a number a_i contributes to $C_t(s)$ if and only if $\log s \in [\log(n/a_i) - t \log \beta, \log(n/a_i) + t \log \beta]$. Indeed, we can restate the lemma in an additive form. Let $\mathcal{A} = \{\alpha_1, \ldots, \alpha_m\}$ be any set of numbers in $[0, \log n]$. These are defined as transformations of a_i's: $\alpha_i \triangleq \log(n/a_i)$. In the additive restating, x will play the role of $\log s$, or equivalently, $s = 2^x$. For a point $x \in [0, \log n]$, let ℓ_j^x and r_j^x denote the distance of x from the jth point to its left and right, respectively, from the set $\mathcal{A}$. More precisely, if $\alpha_\gamma \leq x \leq \alpha_{\gamma+1}$,

$\ell_j^x \triangleq x - \alpha_{\gamma+1-j}$. If we consider only points to the left of x, $\ell_j^x/\log\beta$ is the least value of t such that $C_t(2^x) = j$. Therefore, if

$$t_j^x \triangleq \frac{1}{\log\beta} \min\left\{\ell_j^x, r_j^x\right\} ,$$

then we are guaranteed that $j \le C_{t_j^x}(2^x) \le 2j$.

Observe that $C_t(2^x)$ is a piecewise-constant function which is monotone non-decreasing in t. Therefore, the supremum of $C_t(2^x)/t$ is attained at one of these discontinuities:

$$\sup_{t>0} \frac{C_t(2^x)}{t} = \max_{\{t\,:\,\forall\delta>0, C_{t-\delta}(2^x)<C_t(2^x)\}} \frac{C_t(2^x)}{t} .$$

We can in turn upper bound this by looking at the set of all t_j^x. Note that this may ignore some discontinuous points: for instance, suppose that $\ell_j^x < r_j^x$, then $r_j^x/\log\beta$ will not be considered. However, we note that in this case, $C_{r_j^x/\log\beta}(2^x) \le 2C_{\ell_j^x/\log\beta}(2^x) = 2C_{t_j^x}(2^x)$:

$$\frac{C_{r_j^x/\log\beta}(2^x)}{r_j^x/\log\beta} \le \frac{C_{r_j^x/\log\beta}(2^x)}{\ell_j^x/\log\beta} \le \frac{2C_{\ell_j^x/\log\beta}(2^x)}{\ell_j^x/\log\beta} = \frac{2C_{t_j^x}(2^x)}{t_j^x} .$$

Therefore,

$$\sup_{t>0} \frac{C_t(2^x)}{t} \le \max_{\{t_j^x\,:\,j\in[q]\}} \frac{2C_{t_j^x}(2^x)}{t_j^x} \le \max_{\{t_j^x\,:\,j\in[q]\}} \frac{4j}{t_j^x} = \max_{\{t_j^x\,:\,j\in[q]\}} \frac{4j\log\beta}{\min\{\ell_j^x, r_j^x\}} .$$

We would like to upper bound this term by $2/100$. Equivalently, we satisfy the conditions of the Lemma if

$$\min_j \left\{\frac{t_j^x}{j}\right\} \ge 200\log\beta .$$

For a constant $c > 0$, let S_c be the set of all points x (where recall $s = 2^x$ is selected according to the distribution $\mathcal{S}$) such that violate this inequality for c:

$$\min_j \left\{\frac{t_j^x}{j}\right\} \le c .$$

We would like to upper bound the probability that a randomly selected x violates this inequality for $c = 200\log\beta$ by $1/100$. Equivalently, since x is selected uniformly at random from $(\log n)/(2\log\beta)$ different values, we would like to upper bound the size of $S_{200\log\beta}$ by $\log n/200\log\beta$. We do this with the following claim.

Claim 4.3. $|S_c| \le 2cq$.

Substituting in $c = 200\log\beta$ and $q = \log n/80000\log^2\beta$ will give the desired result.

Proof of Claim 4.3. We consider the set of points in $S_{c,\ell} \subset S_c$ that satisfy $\ell_j^x/j < c$ for some j, and show that their measure is at most cq. An identical bound holds for the set of points of S_c for which $r_j^x/j < c$. Let $S_{c,\ell}^i \subset S_{c,\ell}$ be the set of points in $S_{c,\ell}$ that satisfy $\min_j\{t_j^x/j\} < c$ with respect to the set $\{\alpha_1, \ldots, \alpha_i\}$. We will show by induction that $|S_{c,\ell}^i| < ci$.

For the first point α_1, the set $S^1_{c,\ell} = [\alpha_1, \alpha_1 + c]$. Suppose by induction that $|S^i_{c,\ell}| < ci$. Let x_i be the right-most point in the set $S^i_{c,\ell}$. Then it is clear that $x_i > \alpha_i$, in fact $x_i \geq \alpha_i + c$. Furthermore, either $x_i = \log n$, or $\ell^{x_i}_j / j = c$ for some j. Moreover, we claim that $[\alpha_i, x_i] \in S^i_{c,\ell}$. Indeed, for the same j that $\ell^{x_i}_j / j < c$, all points in $[\alpha_i, x_i]$ satisfy the condition. If $x_i = \log n$, then the result holds trivially. We therefore consider the point α_{i+1} and prove the inductive step for $x_i < \log n$. There are two cases:

If $\alpha_{i+1} \geq x_i$: In this case, $S^{i+1}_{c,\ell} = S^i_{c,\ell} \cup [\alpha_{i+1}, x_{i+1}]$. We have to show that $x_{i+1} \leq \alpha_{i+1} + c$. Suppose to the contrary that $x_{i+1} > \alpha_{i+1} + c \geq x_i + c$. Then there is a point α_h for $h \leq i$, such that

$$\frac{x_{i+1} - \alpha_h}{i+2-h} < c\,, \quad \text{and then} \quad \frac{\alpha_{i+1} + c - \alpha_h}{i+2-h} < c\,,$$

so that

$$\frac{\alpha_{i+1} - \alpha_h}{i+1-h} < c,$$

however, this implies that $\alpha_{i+1} \in S^i_{c,\ell}$, contradicting the assumption of this case.

If $\alpha_{i+1} < x_i$: In this case, $S^{i+1}_{c,\ell} = S^i_{c,\ell} \cup [x_i, x_{i+1}]$. We have to show that $x_{i+1} \leq x_i + c$. Suppose on the contrary that $x_{i+1} > x_i + c > \alpha_{i+1} + c$. Let h be the index such that

$$\frac{x_{i+1} - \alpha_h}{i+2-h} < c\,, \quad \text{and therefore,} \quad \frac{x_i + c - \alpha_h}{i+2-h} < c\,,$$

implying that

$$\frac{x_i - \alpha_h}{i+1-h} < c,$$

contradicting that x_i is the rightmost point of $S^i_{c,\ell}$. □

This concludes the proof of the hitting lemma. □

We proceed to show how to use this lemma to bound the contribution of various types of queries to the distinguishability of D_1 and D_2. In particular, we will apply Lemma 4.2 to the set of query sizes $\{a_1, \ldots, a_q\}$.

Recall that the a_i are non-increasing. If $a_{q'} s/n \leq 1$ let $q' \stackrel{\text{def}}{=} q + 1$, otherwise define q' as the largest integer such that $a_{q'} s/n > 1$. We partition the queries made by $\mathcal{T}$ in two: $A_1, \ldots, A_{q'}$ are said to be *big*, while $A_{q'+1}, \ldots, A_q$ are *small queries*.

Lemma 4.4. *With probability at least $1 - 2^{-10}$, a random distribution from $\mathcal{D}_1$ or from $\mathcal{D}_2$ does not intersect with any small query.*

Proof. Let s be the random size drawn for the definition of the instances. We first claim that

$$\mathbb{E}[|A_{q'+j} \cap S|] \leq \beta^{-50j}$$

for all $j \geq 1$, where the expectation is over the uniform choice of set S_1 for D_1. Indeed, by contradiction suppose there is a $j \geq 1$ such that

$$\mathbb{E}[|A_{q'+j} \cap S|] = \frac{a_{q'+j} s}{n} > \beta^{-50j}\,.$$

By definition of q', for $1 \le i \le j$,

$$1 \ge \frac{a_{q'+iS}}{n} > \beta^{-50j}.$$

Therefore, the queries $A_{q'}, A_{q'+1}, \ldots, A_{q'+j}$ contribute to C_{50j}, and we obtain

$$\frac{C_{50j}}{50j} \ge \frac{j}{50j} = \frac{2}{100},$$

contradicting Lemma 4.2. Thus, the expected intersection can be bounded as follows:

$$\begin{aligned}
\mathbb{E}[\left|(A_{q'+1} \cup A_{q'+2} \cdots \cup A_q) \cap S\right|] &\le \mathbb{E}[\left|A_{q'+1} \cap S\right|] + \mathbb{E}[\left|A_{q'+2} \cap S\right|] + \cdots + \mathbb{E}[\left|A_q \cap S\right|] \\
&\le \beta^{-50} + \beta^{-100} + \ldots \\
&\le 2^{-12},
\end{aligned}$$

since $\beta \ge 2$. From this, we obtain the result holds for $\mathcal{D}_1$ by Markov's inequality. The same applies to $\mathcal{D}_2$ with probability of intersection at most 2^{-10}, proving the lemma. $\square$

We now turn our attention to the sets with *large* intersections. We will show that under $\mathcal{D}_1$ and $\mathcal{D}_2$, the results of querying the sets $A_1, \ldots A_{q'}$ are indistinguishable from simply picking elements uniformly from the sets $A_1, \ldots, A_{q'}$. More precisely, we establish the following.

Lemma 4.5. *Let $\eta^* = 2^{-10}$ and $\eta_s = 1/100$; and fix $\ell \in \{1,2\}$. At least an $1-\eta_s$ fraction of elements $s_1, \ldots, s_{q'} \in A_1 \times A_2, \ldots, A_{q'}$ satisfy*

$$\Pr_{\ell}\left[(s_1, \ldots, s_{q'})\right] \in [1-5\eta^*, 1+5\eta^*] \cdot \frac{1}{|A_1| \ldots |A_{q'}|}, \tag{4.4}$$

where $\Pr_\ell\left[(s_1, \ldots, s_{q'})\right]$ denotes the probability that $s_1 \ldots s_{q'}$ are the results of the queries $A_1, \ldots, A_{q'}$ under COND_{D_ℓ}.

As this holds for most distributions in both $\mathcal{D}_1$ and $\mathcal{D}_2$, this implies the queries are indistinguishable from the product distribution over $A_1 \times A_2, \ldots, A_{q'}$ (which is the one induced by the same queries on the uniform distribution over $[n]$) in either case, with probability at least $1 - \eta_s - 5\eta^*$.

Proof of Lemma 4.5. From standard Chernoff-like concentration bounds for hypergeometric random variables (Lemma 2.7), we obtain the claim below.

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

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

We use this to show that indeed all the $|A_i \cap S|$ concentrate around their expected values for $1 \le j \le q'$. First note that, as a consequence of Lemma 4.2, it is the case that these expected values satisfy

$a_{q'-j}s/n \geq \beta^{50(j+1)}$ for every $0 \leq j \leq q'-1$ (with probability at least $99/100$). Conditioning on this, we first invoke Claim 4.6 on A_j with $\eta = 3 \cdot \beta^{20(j+1)}$, and then apply a union bound to obtain

$$\Pr\left[\exists j \in [q'] \text{ s.t. } \left|A_j \cap S\right| \notin \left[1 - 4 \cdot \beta^{-20(j+1)}, 1 + 4 \cdot \beta^{-20(j+1)}\right] \cdot \frac{a_j s}{n}\right] < e^{-\beta^{10}} \tag{4.5}$$

i. e., with high probability all intersections simultaneously concentrate around their expected values. Note that since s is at most $n^{3/4}$, each A_i under consideration has size at least $n\beta^{50}/n^{3/4} > n^{1/4}$. Therefore, the probability that a random selection of elements from $A_1, \ldots, A_{q'}$ exhibits no collision is at least

$$\prod_{i=1}^{q'} \frac{|A_i| - q'}{|A_i|} \geq \left(1 - \frac{q'}{n^{1/4}}\right)^{q'} \geq 1 - \frac{(q')^2}{n^{1/4}} > 1 - \frac{\log^2 n}{n^{1/4}}.$$

We henceforth condition on this event.

Let $N = \binom{n}{s}$ be the number of outcomes for the set S. We write $N_0 \geq N(1 - e^{-\beta^{10}})$ for the number of such sets for which equation (4.5) holds. Let $s_1^{q'}$ denote $s_1 \ldots s_{q'}$. For a set of distinct $(s_1, \ldots, s_{q'}) \in A_1 \times \cdots \times A_{q'}$, let $N(s_1^{q'}) = \binom{n-q'}{s-q'}$ be the number of sets of size s that contain $s_1^{q'}$, and let $N_0(s_1^{q'})$ of them satisfy equation (4.5).

By Markov's inequality, with probability at least $1 - e^{-\beta^9}$, for a randomly chosen $s_1^{q'}$ we have $N_0(s_1^{q'})/N(s_1^{q'}) > 1 - e^{2-\beta^9}$. For any such $s_1^{q'}$,

$$\begin{aligned}\Pr\left[s_1^{q'}\right] &\geq \frac{N_0(s_1^{q'})}{N} \cdot \prod_{i=1}^{q'} \frac{1}{|A_i \cap S|} \geq (1 - e^{2-\beta^9}) \frac{s(s-1)\ldots(s-q'+1)}{n(n-1)\ldots(n-q'+1)} \cdot (1 - 4 \cdot \beta^{-19}) \prod_{i=1}^{q'} \frac{n}{a_i s} \\ &\geq (1 - 6 \cdot \beta^{-19}) \prod_{i=1}^{q'} \frac{1}{a_i},\end{aligned}$$

for large n and as $|S| > n^{1/4}$. Since the sum of probabilities of elements is at most 1, the other side of the inequality in Lemma 4.5 follows. □

Proof of Theorem 1.2 and Theorem 1.3. Let T_1 (resp. T_2, T_U) be the distribution over transcripts generated by the queries $A_1, \ldots, A_q$ when given conditional access to the distribution D_1 from a no-instance (resp. D_2, resp. uniform $\mathcal{U}_{[n]}$); that is, a distribution over q-tuples in $A_1 \times \cdots \times A_q$. Since the queries were non-adaptive, we can break T_1 (and similarly for T_2, T_U) in $T_1^{\text{big}} \times T_1^{\text{small}}$ according to q', and use Lemma 4.5 and Lemma 4.4 separately to obtain $\mathrm{d_{TV}}(T_1, T_2) \leq \eta_s + \eta^* + 2^{-10} < 1/50$ and $\mathrm{d_{TV}}(T_1, T_U) \leq \eta_s + \eta^* + 2^{-10} < 1/50$ (for the latter, recalling that queries that do not intersect the support receive samples exactly uniformly distributed in the query set)—thus establishing both theorems. □

4.1 On the dependence on ε in Theorem 1.3

We remark that Theorem 1.3, by establishing a lower bound of $\Omega(\log n)$ queries for non-adaptive testing of uniformity with constant distance parameter $1/4$, immediately implies, by a standard argument, an $\Omega((\log n)/\varepsilon)$ lower bound for distance parameter $\varepsilon \in (0, 1/4)$. In more detail, this is a consequence of the following reduction: any $q(n,\varepsilon)$-query non-adaptive tester for uniformity $\mathcal{T}$ can be used, given conditional access to some distribution D on $[n]$, on the mixture distribution

$$D_\varepsilon \stackrel{\text{def}}{=} 4\varepsilon D + (1-4\varepsilon)\mathcal{U}_{[n]}\,, \tag{4.6}$$

for which a conditional oracle can be easily simulated given a conditional oracle for D. Moreover, answering $q(n,\varepsilon)$ to D_ε can be done with an expected $4\varepsilon q(n,\varepsilon)$ conditional queries to D. As it is immediate to see that $\mathrm{d_{TV}}(D_\varepsilon,\mathcal{U}_{[n]}) = 4\varepsilon\,\mathrm{d_{TV}}(D,\mathcal{U}_{[n]})$, we get that $\mathcal{T}$ can be used to obtain a tester for non-adaptive testing of uniformity with constant distance parameter $1/4$, with query complexity $O(\varepsilon q(n,\varepsilon))$ for every $\varepsilon < 1/4$ (converting the expected query complexity to a worst-case one is straightforward via Markov's inequality followed by success probability amplification by a constant number of repetitions). Therefore, the lower bound of Theorem 1.3 implies that $q(n,\varepsilon) = \Omega((\log n)/\varepsilon)$, as claimed. It is also worth noting that the above argument does *not* yield an analogue statement for support-size estimation via Theorem 1.2. Indeed, mixing the distribution D with the uniform distribution does not preserve the support size in that case (nor the guarantee that every point of the support has probability mass at least τ/n).

5 An upper bound for support-size estimation

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

Theorem 1.4 (Adaptive Support-Size Estimation)**.** *Let $\tau > 0$ be any constant. There exists an adaptive algorithm which, given* COND *access to an unknown distribution D on $[n]$ (guaranteed to have probability mass at least τ/n on every element of its support) and accuracy parameter $\varepsilon \in (0,1)$, makes $\tilde{O}\big((\log\log n)/\varepsilon^3\big)$ queries to the oracle*[11] *and returns a value $\tilde{\omega}$ such that the following holds. With probability at least $2/3$, $\tilde{\omega} \in [(1/(1+\varepsilon))\cdot\omega, (1+\varepsilon)\cdot\omega]$, where $\omega = |\mathrm{supp}(D)|$.*

Before describing and analyzing our algorithm, we shall need the following results, that we will use as subroutines: the first one will help us detecting when the support is already dense. The second, assuming the support is sparse enough, will enable us to find an element with zero probability mass, which can afterwards be used as a "reference" to verify whether any given element is inside or outside the support. Finally, the last one will use such a reference point to check whether a candidate support size σ is smaller or significantly bigger than the actual support size.

Lemma 5.1. *Given $\tau > 0$ and* COND *access to a distribution D such that each support element has probability at least τ/n, as well as parameters $\varepsilon \in (0,1/2), \delta \in (0,1)$, there exists an algorithm* TESTSMALLSUPPORT *(Algorithm 2) that makes*

$$\tilde{O}\left(\frac{1}{\tau\varepsilon^2} + \frac{1}{\tau^2}\right)\cdot\log(1/\delta)$$

queries to the oracle, and satisfies the following. (i) If $\mathrm{supp}(D) \geq (1-\varepsilon/2)n$, then it returns ACCEPT with probability at least $1-\delta$; (ii) if $\mathrm{supp}(D) \leq (1-\varepsilon)n$, then it returns REJECT with probability at least $1-\delta$.

[11] We remark that the constant in the $\tilde{O}$ depends polynomially on $1/\tau$.

Lemma 5.2. *Given* COND *access to a distribution D, an upper bound $m < n$ on* $\operatorname{supp}(D)$*, as well as parameter $\delta \in (0,1)$, there exists an algorithm* GETNONSUPPORT *(Algorithm 3) that makes*

$$\tilde{O}\left(\log^2 \frac{1}{\delta} \log^{-2} \frac{n}{m}\right)$$

queries to the oracle, and returns an element $r \in [n]$ such that $r \notin \operatorname{supp}(D)$ with probability at least $1-\delta$.

Lemma 5.3. *Given* COND *access to a distribution D, inputs $\sigma \geq 2$ and $r \notin \operatorname{supp}(D)$, as well as parameters $\varepsilon \in (0,1/2), \delta \in (0,1)$, there exists an algorithm* ISATMOSTSUPPORTSIZE *(Algorithm 4) that makes $\tilde{O}(1/\varepsilon^2)\log(1/\delta)$ queries to the oracle, and satisfies the following. The algorithm returns either* yes *or* no*, and (i) if $\sigma \geq \operatorname{supp}(D)$, then it returns* yes *with probability at least $1-\delta$; (ii) if $\sigma > (1+\varepsilon)\operatorname{supp}(D)$, then it returns* no *with probability at least $1-\delta$.*

We defer the proofs of these 3 lemmata to the next subsections, and now turn to the proof of the theorem.

Proof. The algorithm is given in Algorithm 1, and at a high-level works as follows: if first checks whether the support size is big (an $1-O(\varepsilon)$ fraction of the domain), in which case it can already stop and return a good estimate. If this is not the case, however, then the support is sparse enough to efficiently find an element r *outside* the support, by taking a few uniform points, comparing and ordering them by probability mass (and keeping the lightest). This element r can then be used as a reference point in a (doubly exponential) search for a good estimate: for each guess $\tilde{\omega}$, a random subset S of size roughly $\tilde{\omega}$ is taken, a point x is drawn from D_S, and x is compared to r to check if $D(x) > 0$. If so, then S intersects the support, meaning that $\tilde{\omega}$ is an upper bound on ω; repeating until this is no longer the case results in an accurate estimate of ω.

Algorithm 1 ESTIMATESUPPORT$_D$

if TESTSMALLSUPPORT$_D(\varepsilon, \frac{1}{10})$ returns ACCEPT **then return** $\tilde{\omega} \leftarrow (1-\varepsilon^2)n$
end if
Call GETNONSUPPORT$_D((1-\frac{\varepsilon}{2})n, \frac{1}{10})$ to obtain a non-support reference point r.
for j **from** 0 **to** $\log_{1+\varepsilon}\log_{1+\varepsilon} n$ **do**
    Set $\tilde{\omega} \leftarrow (1+\varepsilon)^{(1+\varepsilon)^j}$.
    Call ISATMOSTSUPPORTSIZE$_D(\tilde{\omega}, r, \varepsilon, \frac{1}{100\cdot(j+1)^2})$ to check if $\tilde{\omega}$ is an upper bound on ω.
    if the call returned no **then**
        Perform a binary search on $\{(1+\varepsilon)^{j-1}, \ldots, (1+\varepsilon)^j\}$ to find i^*, the smallest $i \geq 2$ such that ISATMOSTSUPPORTSIZE$_D((1+\varepsilon)^i, r, \varepsilon, \frac{1}{10(j+1)})$ returns no.
        return $\tilde{\omega} \leftarrow (1+\varepsilon)^{i^*-1}$.
    end if
end for

In the rest of this section, we formalize and rigorously argue the above. Conditioning on each of the calls to the subroutines TESTSMALLSUPPORT, GETNONSUPPORT and ISATMOSTSUPPORTSIZE being correct (which overall happens except with probability at most

$$\frac{1}{10}+\frac{1}{10}+\sum_{j=1}^{\infty}\frac{1}{100j^2}+\frac{1}{10}<\frac{1}{3}$$

by a union bound), we show that the output $\tilde{\omega}$ of ESTIMATESUPPORT is indeed within a factor $(1+\varepsilon)$ of ω.

- If the test on Step 1 passes, then by Lemma 5.1 we must have $\operatorname{supp}(D)>(1-\varepsilon)n$. Thus, the estimate we return is correct, as $[(1-\varepsilon)n,n]\subseteq[\tilde{\omega}/(1+\varepsilon),(1+\varepsilon)\tilde{\omega}]$.
- Otherwise, if it does not then by Lemma 5.1 it must be the case that $\operatorname{supp}(D)<(1-\varepsilon/2)n$.

Therefore, if we reach Step 3 then $(1-\varepsilon/2)n$ is indeed an upper bound on ω, and GETNONSUPPORT will return a point $r\notin\operatorname{supp}(D)$ 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 $\omega\in[(1+\varepsilon)^{(1+\varepsilon)^{j-1}},(1+\varepsilon)^{(1+\varepsilon)^{j}}]$; and will be on a set of $(1+\varepsilon)^{j-1}$ values. Similarly, for the value i^* eventually obtained, it must be the case that $(1+\varepsilon)^{i^*}>\omega$ (by contrapositive, as no was returned by the subroutine) but $(1+\varepsilon)^{i^*-1}\leq(1+\varepsilon)\omega$ (again, as the subroutine returned yes). But then, $\tilde{\omega}=(1+\varepsilon)^{i^*-1}\in(\omega/(1+\varepsilon),(1+\varepsilon)\omega]$ as claimed.

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

- the call to TESTSMALLSUPPORT, which from Lemma 5.1 costs $\tilde{O}(1/\varepsilon^2)$ queries;
- the call to GETNONSUPPORT, on Step 3, that from the choice of the upper bound also costs $\tilde{O}(1/\varepsilon^2)$ queries;
- the (at most) $\log_{1+\varepsilon}\log_{1+\varepsilon}n=O((\log\log n)/\varepsilon)$ calls to ISATMOSTSUPPORTSIZE on Step 6. Observing that the query complexity of ISATMOSTSUPPORTSIZE is only $\tilde{O}(1/\varepsilon^2)\cdot\log(1/\delta)$, and from the choice of $\delta=1/(j+1)^2$ at the j-th iteration this step costs at most

$$\tilde{O}\left(\frac{1}{\varepsilon^2}\right)\cdot\sum_{j=1}^{\log_{1+\varepsilon}\log_{1+\varepsilon}n}O(\log(j^2))=\tilde{O}\left(\frac{1}{\varepsilon^2}\log_{1+\varepsilon}\log_{1+\varepsilon}n\right)=\tilde{O}\left(\frac{1}{\varepsilon^3}\log_{1+\varepsilon}\log_{1+\varepsilon}n\right)$$

  queries.
- Similarly, Step 8 results in at most $j\leq\log\log n$ calls to ISATMOSTSUPPORTSIZE with δ set to $1/(10(j+1))$, again costing

$$\tilde{O}\left(\frac{1}{\varepsilon^2}\right)\cdot\log j=\tilde{O}\left(\frac{1}{\varepsilon^2}\log_{1+\varepsilon}\log_{1+\varepsilon}n\right)=\tilde{O}\left(\frac{1}{\varepsilon^3}\log\log n\right)$$

  queries.

Gathering all terms, the overall query complexity is $\tilde{O}\left(\frac{\log\log n}{\varepsilon^3}\right)$, as claimed. □

5.1 Proof of Lemma 5.1

Hereafter, we assume without loss of generality that $\tau < 2$: indeed, if $\tau \geq 2$ then the support is of size at most $n/2$, and it suffices to return REJECT to meet the requirements of the lemma. We will rely on the (easy) fact below, which ensures that any distribution with dense support and minimum non-zero probability τ/n put significant mass on "light" elements.

Fact 5.4. *Fix any $\varepsilon \in [0,1)$. Assume D satisfies both $\operatorname{supp}(D) \geq (1-\varepsilon)n$ and $D(x) \geq \tau/n$ for $x \in \operatorname{supp}(D)$. Then, setting $L_\varepsilon \stackrel{\text{def}}{=} \{\, x \in [n] : D(x) \in [\tau/n, 2/n] \,\}$, we have $|L_\varepsilon| \geq (1/2-\varepsilon)n$ and $D(L_\varepsilon) \geq (1/2-\varepsilon)\tau$.*

Proof. As the second claim follows directly from the first and the minimum mass of elements of L_ε, it suffices to prove that $|L_\varepsilon| \geq (1/2-\varepsilon)n$. This follows from observing that

$$1 = D([n]) \geq D([n] \setminus L_\varepsilon) \geq (|\operatorname{supp}(D)| - |L_\varepsilon|)\frac{2}{n} \geq 2(1-\varepsilon) - \frac{2\,|L_\varepsilon|}{n}$$

and rearranging the terms. □

Description and intuition. The algorithm (as described in Algorithm 2) works as follows: it first takes enough uniformly distributed samples $s_1, \dots, s_m$ to get (with high probability) an accurate enough fraction of them falling in the support to distinguish between the two cases. The issue is now to detect those m_j which indeed are support elements; note that we do not care about underestimating this fraction in case (b) (when the support is at most $(1-\varepsilon)n$, but importantly do not want to underestimate it in case (a) (when the support size is at least $(1-\varepsilon/2)n$). To perform this detection, we take constantly many samples *according to D* (which are therefore ensured to be in the support), and use pairwise conditional queries to sort them by increasing probability mass (up to approximation imprecision), and keep only the lightest of them, t. In case (a), we now from Fact 5.4 that with high probability our t has mass in $[1/n, 2/n]$, and will therefore be either much lighter than or comparable to *any* support element: this will ensure that in case (a) we do detect all of the m_j that are in the support.

This also works in case (b), even though Fact 5.4 does not give us any guarantee on the mass of t. Indeed, either t turns out to be light (and then the same argument ensures that our estimate of the number of "support" elements m_j is good), or t is too heavy—and then our estimate will end up being smaller than the true value. But this is fine, as the latter only means we will reject the distribution (as we should, since we are in the small-support case).

Correctness. Let η be the fraction of the elements s_j that are in the support of the distribution. By a multiplicative Chernoff bound and a suitable constant in our choice of m, we get that (i) if $\operatorname{supp}(D) \geq 1-\varepsilon/2$, then $\Pr[\,\eta < 1-3\varepsilon/4\,] \leq 1/12$, while (ii) if $\operatorname{supp}(D) \leq 1-\varepsilon/2$, then $\Pr[\,\eta \geq 1-3\varepsilon/4\,] \leq 1/12$. We hereafter condition on this (i. e., η being a good enough estimate). We also condition on all calls to COMPARE yielding results as per specified, which by a union bound overall happens except with probability $1/12 + 1/12 = 1/6$, and break the rest of the analysis in two cases.

(a) Since the support size ω is in this case at least $(1-\varepsilon/2)n$, from Fact 5.4 we get that

$$D(L_{\varepsilon/2}) \geq \frac{1-\varepsilon}{2}\tau \geq \frac{\tau}{4}\,.$$

Algorithm 2 TESTSMALLSUPPORT$_D$

Require: COND access to D; accuracy parameter $\varepsilon \in (0,1/2)$, threshold $\tau > 0$, probability of failure δ

1: Repeat the following $O(\log(1/\delta))$ times and return the majority vote.
2: **loop**
3: Draw $m \stackrel{\text{def}}{=} \Theta\left(\frac{1}{\varepsilon^2}\right)$ independent samples $s_1, \dots, s_m \sim \mathcal{U}_{[n]}$.
4: Draw $k \stackrel{\text{def}}{=} \Theta\left(\frac{1}{\tau}\right)$ independent samples $t_1, \dots, t_k \sim D$.
5: **for all** $1 \le i < j \le k$ **do** ▷ Order the t_j
6: Call COMPARE$(\{t_i\}, \{t_j\}, \eta = \frac{1}{2}, K = 2, \frac{1}{4k^2})$ to get a 2-approx. ρ of $\frac{D(t_j)}{D(t_i)}$, High or Low.
7: **if** COMPARE returned High or a value ρ **then**
8: Record $t_i \preceq t_j$
9: **else**
10: Record $t_j \prec t_j$
11: **end if**
12: **end for**
13: Set t to be (any of the) smallest elements t_j according to $\preceq$.
14: **for all** $1 \le j \le m$ **do** ▷ Find the fraction of support elements among the m_j
15: Call COMPARE$(\{t\}, \{s_j\}, \eta = \frac{1}{2}, K = \frac{2}{\tau}, \frac{1}{4m})$ to get either a value ρ, High or Low.
16: **if** COMPARE returned High or a value $\rho \ge 1/2$ **then**
17: Record s_j as "support."
18: **end if**
19: **end for**
20: **if** the number of elements s_j marked "support" is at least $(1 - \frac{3}{4}\varepsilon)m$ **then return** ACCEPT
21: **else return** REJECT
22: **end if**
23: **end loop**

Therefore, except with probability at most $(1-\tau/4)^k < 1/12$, at least one of the t_j will belong to $L_{\varepsilon/2}$. When this happens, and by the choice of parameters in the calls to COMPARE, we get that $t \in L_{\varepsilon/2}$; that is $D(t) \in [\tau/n, 2/n]$. But then the calls to the routine on Step 15 will always return either a value (since t is "comparable" to all $x \in L_{\varepsilon/2}$—i. e., has probability within a factor $2/\tau$ of them) or High (possible for those s_j that have weight greater than $2/n$), unless s_j has mass 0 (that is, is not in the support). Therefore, the fraction of points marked as support is exactly η, which by the foregoing discussion is at least $1 - 3\varepsilon/4$: the algorithm returns ACCEPT at Step 20.

(b) Conversely, if $\omega \le (1-\varepsilon)n$, there will be a fraction $1-\eta > 3\varepsilon/4$ of the s_j having mass 0. However, no matter what t is it will still be in the support and therefore have $D(t) \ge \tau/n$: for these s_j, the call to COMPARE on Step 15 can thus only return Low. This means that there can only be less than $(1-(3/4)\varepsilon)m$ points marked "support" among the s_j, and hence that the algorithm will return REJECT as it should.

Overall, the inner loop of the algorithm thus only fails with probability at most $1/12 + 1/6 + 1/12 = 1/3$, (where the 3 events contributing to the union bound are (i) when η fails to be a good estimate, (ii) when the calls to COMPARE fail to yield results as claimed, and (iii) when no t_j hits $L_{\varepsilon/2}$ in case (a)). Repeating independently $\log(1/\delta)$ times and taking the majority vote boosts the probability of success to $1-\delta$.

Query complexity. The sample complexity comes from the k^2 calls on Step 4 (each costing $O(\log k)$ queries) and the m calls on Step 15 (each costing $O((1/\tau)\log m)$ queries). By the setting of m and because of the $\log(1/\delta)$ repetitions, this results in an overall query complexity

$$O\left(\left(\frac{1}{\tau^2}\log\frac{1}{\tau}+\frac{1}{\tau\varepsilon^2}\log\frac{1}{\varepsilon}\right)\log\frac{1}{\delta}\right). \qquad \square$$

5.2 Proof of Lemma 5.2

As described in Algorithm 3, the subroutine is fairly simple: using its knowledge of an upperbound on the support size, it takes enough uniformly distributed samples to have (with high probability) at least one falling outside the support. Then, it uses the conditional oracle to "order" these samples according to their probability mass, and returns the lightest of them—i. e., one with zero probability mass.

Algorithm 3 GETNONSUPPORT$_D(m,\delta)$

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

```
Set k ≝ ⌈log(2/δ) log^{-1}(n/m)⌉.
Draw independently k points s_1, ..., s_k ~ U_[n]
for all 1 ≤ i < j ≤ k do
    Call COMPARE({s_i}, {s_j}, η = 1/2, K = 2, δ/(2k^2)) to get a 2-approx. ρ of D(s_j)/D(s_i), High or Low.
    if COMPARE returned High or a value ρ then
        Record s_i ⪯ s_j
    else
        Record s_j ≺ s_j
    end if
end for
return argmin_⪯{s_1, ..., s_k}                ▷ Return (any) minimal element for ⪯.
```

Correctness. It is straightforward to see that provided at least one of the s_j falls outside the support and that all calls to COMPARE behave as expected, then the procedure returns one of the "lightest" elements s_j, i. e., a non-support element. By a union bound, the latter holds with probability at least $1-\delta/2$; as for the former, since m is by assumption an upper bound on the support size it holds with probability at least $1-(m/n)^k\geq 1-\delta/2$ (from our setting of k). Overall, the procedure's output is correct with probability at least $1-\delta$, as claimed.

Query complexity. The query complexity of GETNONSUPPORT is due to the $\binom{k}{2}$ calls to COMPARE, and is therefore $O\big(k^2\log k/\delta\big)$ because of our setting for η and K (which is in turn

$\tilde{O}\big(\log^2(1/\delta)\log^{-2}(n/m)\big)$). (In our case, we shall eventually take $m = (1-\varepsilon/2)n$ and $\delta = 1/10$, thus getting $k = O(1/\varepsilon)$ and a query complexity of $\tilde{O}\big(1/\varepsilon^2\big)$.) $\square$

5.3 Proof of Lemma 5.3

Our final subroutine, described in Algorithm 4, essentially derives from the following observation: a random set S of size (approximately) σ obtained by including independently each element of the domain with probability $1/\sigma$ will intersect the support on ω/σ points on expectation. What we *can* test given our reference point $r \notin \operatorname{supp}(D)$, however, is only whether $S \cap \operatorname{supp}(D) = \emptyset$. But this is enough, as by repeating sufficiently many times (taking a random S and testing whether it intersects the support at all) we can distinguish between the two cases we are interested in. Indeed, the expected fraction of times S includes a support element in either cases is known to the algorithm and differs by roughly $\Omega(\varepsilon)$, so $O\big(1/\varepsilon^2\big)$ repetitions are enough to tell the two cases apart.

Algorithm 4 ISATMOSTSUPPORTSIZE$_D(\sigma, r, \varepsilon, \delta)$

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

```
 1: Set α ← (1 − 1/σ)^σ ∈ [1/4, e^{−1}], τ ← α(α^{−ε/2} − 1) = Θ(ε).
 2: Repeat the following O(log(1/δ)) times and return the majority vote.
 3: loop
 4:     for m = O(1/τ²) times do
 5:         Draw a subset S ⊆ [n] by including independently each x ∈ [n] with probability 1/σ.
 6:         Draw x ∼ D_S.
 7:         Call COMPARE({x}, {r}, η = 1/2, K = 1, 1/(100m))    ▷ Low if S ∩ supp(D) ≠ ∅; ρ ∈ [1/2, 2) o.w.
 8:         Record yes if COMPARE returned Low, no otherwise.
 9:     end for
10:     return yes if at least m(α + τ/2) "yes"'s were recorded, no otherwise.    ▷ Thresholding.
11: end loop
```

Correctness. We condition on all calls to COMPARE being correct: by a union bound, this overall happens with probability at least $99/100$. We shall consider the two cases $\sigma \leq \omega$ and $\sigma > (1+\varepsilon)\omega$, and focus on the difference of probability p of recording yes on Step 8 between the two, in any fixed of the m iterations. In both cases, note p is exactly $(1-1/\sigma)^{\omega}$.

- If $\sigma \leq \omega$, then we have $p \leq \left(1-\frac{1}{\sigma}\right)^{\sigma} = \alpha$.
- If $\sigma > (1+\varepsilon)\omega$, then $p > \left(1-\frac{1}{\sigma}\right)^{\sigma/(1+\varepsilon)} > \left(1-\frac{1}{\sigma}\right)^{\sigma(1-\varepsilon/2)} = \alpha^{1-\varepsilon/2}$.

As $\alpha \in [\frac{1}{4}, e^{-1}]$, the difference between the two is $\tau = \alpha(\alpha^{-\varepsilon/2}-1) = \Theta(\varepsilon)$. Thus, repeating the atomic test of Step 4 $O\big(1/\tau^2\big)$ before thresholding at Step 10 yields the right answer with constant probability, then brought to $1-\delta$ by the outer repeating and majority vote.

Query complexity. Each call to COMPARE at Step 7 costs $O(\log m)$ queries, and is in total repeated $O(m\log(1/\delta)$ times. By the setting of m and τ, the overall query complexity is therefore

$$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\log\frac{1}{\delta}\right). \qquad \square$$

5.4 A non-adaptive upper bound

In this section, we sketch how similar—yet less involved—ideas can be used to derive a non-adaptive upper bound for support-size estimation. For simplicity, we describe the algorithm for 2-approximation: adapting it to general $(1+\varepsilon)$-approximation is straightforward.

The high-level idea is to perform a simple binary search (instead of the double exponential search from the preceding section) to identify the greatest lower bound on the support size of the form $k = 2^j$. For each guess $k \in \{2,4,8\ldots,n\}$, we pick uniformly at random a set $S \subseteq [n]$ of cardinality k, and check whether D_S is uniform using the non-adaptive tester of Chakraborty et al. [16, Theorem 4.1.2]. If D_S is found to be uniform for all values of k, we return n as our estimate (as the distribution is close to uniform on $[n]$); otherwise, we return n/k, for the smallest k on which D_S was found to be far from uniform. Indeed, D_S can only be far from uniform if S contains points from the support of D, which intuitively only happens if $n/k = \Omega(1)$.

To be more precise, the algorithm proceeds as follows, where $\tau > 0$ is an absolute constant.

for all $k \in \{2,4,\ldots,n\}$ **do**
  Set a counter $c_k \leftarrow 0$.
  for $m = O(\log\log n)$ times **do**
    Pick uniformly at random a set $S \subseteq [n]$ of k elements.
    Test (non-adaptively) uniformity of D_S on S, with the tester of [16].
    if the tester rejects **then** increment c_k.
    end if
  end for
  if $c_k > \tau \cdot m$ **then return** $\tilde{\omega} \leftarrow \frac{n}{k}$.
  end if
end for
return $\tilde{\omega} \leftarrow n$.

The query complexity is easily seen to be $\operatorname{poly}\log n$, from the $\tilde{O}(\log n)$ calls to the $\operatorname{poly}(\log n)$ tester of [16, Theorem 4.1.2]. As for correctness, it follows from the fact that for any set S with mass $D(S) > 0$ which contains at least an η fraction of points outside the support, it holds that D_S is η-far from $\mathcal{U}_S$.

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 3. The authors are grateful to László Babai, Robert Krauthgamer, and the anonymous reviewers for their helpful and detailed comments.

References

[1] List of Open Problems in Sublinear Algorithms: Problem 66. Bertinoro Workshop on Sublinear Algorithms 2014 (suggested by Eldar Fischer). Available at `http://sublinear.info/66`. 5

[2] JAYADEV ACHARYA, CLÉMENT L. CANONNE, AND GAUTAM KAMATH: Adaptive estimation in weighted group testing. In *IEEE Internat. Symp. Information Theory (ISIT'15)*, pp. 2116–2120, 2015. [doi:10.1109/ISIT.2015.7282829] 3

[3] JAYADEV ACHARYA, CLÉMENT L. CANONNE, AND GAUTAM KAMATH: A chasm between Identity and Equivalence Testing with conditional queries. In *Proc. 19th Internat. Workshop on Randomization and Computation (RANDOM'15)*, volume 40, pp. 449–466. Schloss Dagstuhl, 2015. [doi:10.4230/LIPIcs.APPROX-RANDOM.2015.449, arXiv:1411.7346, ECCC:TR14-156] 1

[4] JAYADEV ACHARYA, HIRAKENDU DAS, ASHKAN JAFARPOUR, ALON ORLITSKY, AND SHENGJUN PAN: Competitive closeness testing. In *Proc. 24th Ann. Conf. on Learning Theory (COLT'11)*, volume 19 of *JMLR Proceedings*, pp. 47–68. JMLR.org, 2011. Accessible at JMLR. 2

[5] JAYADEV ACHARYA, HIRAKENDU DAS, ASHKAN JAFARPOUR, ALON ORLITSKY, SHENGJUN PAN, AND ANANDA THEERTHA SURESH: Competitive classification and closeness testing. In *Proc. 25th Ann. Conf. on Learning Theory (COLT'12)*, pp. 22.1–22.18, 2012. Accessible at JMLR. 2

[6] ANNE AUGER AND BENJAMIN DOERR: *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. Series on theoretical computer science. World Scientific, 2011. 13

[7] MARIA-FLORINA BALCAN, ERIC BLAIS, AVRIM BLUM, AND LIU YANG: Active property testing. In *Proc. 53rd FOCS*, pp. 21–30. IEEE Comp. Soc. Press, 2012. [doi:10.1109/FOCS.2012.64] 3

[8] TUĞKAN BATU, ELDAR FISCHER, LANCE FORTNOW, RAVI KUMAR, RONITT RUBINFELD, AND PATRICK WHITE: Testing random variables for independence and identity. In *Proc. 42nd FOCS*, pp. 442–451. IEEE Comp. Soc. Press, 2001. [doi:10.1109/SFCS.2001.959920] 2

[9] TUĞKAN BATU, LANCE FORTNOW, RONITT RUBINFELD, WARREN D. SMITH, AND PATRICK WHITE: Testing that distributions are close. In *Proc. 41st FOCS*, pp. 189–197. IEEE Comp. Soc. Press, 2000. This is a preliminary version of [10]. [doi:10.1109/SFCS.2000.892113] 2, 42

[10] TUĞKAN BATU, LANCE FORTNOW, RONITT RUBINFELD, WARREN D. SMITH, AND PATRICK WHITE: Testing closeness of discrete distributions. 2010. This is a long version of [9]. [arXiv:1009.5397] 4, 5, 42

[11] TUĞKAN BATU, RAVI KUMAR, AND RONITT RUBINFELD: Sublinear algorithms for testing monotone and unimodal distributions. In *Proc. 36th STOC*, pp. 381–390. ACM Press, 2004. [doi:10.1145/1007352.1007414] 2

[12] Arnab Bhattacharyya, Eldar Fischer, Ronitt Rubinfeld, and Paul Valiant: Testing monotonicity of distributions over general partial orders. In *Proc. 2nd Conf. on Innovations in Theoret. Comput. Sci. (ITCS'11)*, pp. 239–252, 2011. [ECCC:TR10-027] 2

[13] Clément L. Canonne: A survey on distribution testing: Your data is Big, but is it Blue? *Electron. Colloq. on Comput. Complexity (ECCC)*, April 2015. [ECCC:TR15-063] 2

[14] Clément L. Canonne, Dana Ron, and Rocco A. Servedio: Testing probability distributions using conditional samples. *SIAM J. Comput.*, 44(3):540–616, 2015. Preliminary version in SODA'14. [doi:10.1137/130945508, arXiv:1211.2664, ECCC:TR12-155] 2, 3, 4, 5, 7, 9, 10, 15

[15] Clément L. Canonne and Ronitt Rubinfeld: Testing probability distributions underlying aggregated data. In *Proc. 41st Internat. Colloq. on Automata, Languages and Programming (ICALP'14)*, pp. 283–295. Springer, 2014. [doi:10.1007/978-3-662-43948-7_24, arXiv:1402.3835, ECCC:TR14-021] 2

[16] Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah: On the power of conditional samples in distribution testing. *SIAM J. Comput.*, 45(4):1261–1296, 2016. Preliminary version in ITCS'13. [doi:10.1137/140964199, arXiv:1210.8338, ECCC:TR12-154] 2, 3, 4, 6, 7, 9, 10, 16, 41

[17] Chun Lam Chan, Sidharth Jaggi, Venkatesh Saligrama, and Samar Agnihotri: Non-adaptive group testing: Explicit bounds and novel algorithms. *IEEE Trans. Inform. Theory*, 60(5):3019–3035, 2014. Preliminary version in ISIT'12. [doi:10.1109/TIT.2014.2310477, arXiv:1202.0206] 3

[18] Siu-On Chan, Ilias Diakonikolas, Gregory Valiant, and Paul Valiant: Optimal algorithms for testing closeness of discrete distributions. In *Proc. 25th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'14)*, pp. 1193–1203. ACM Press, 2014. [doi:10.1137/1.9781611973402.88, arXiv:1308.3946] 2, 5, 7

[19] Vašek Chvátal: The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285 – 287, 1979. [doi:10.1016/0012-365X(79)90084-0] 13

[20] Sanjoy Dasgupta: Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems 17*, pp. 337–344. MIT Press, 2004–2005. Available at NIPS. 3

[21] Robert Dorfman: The detection of defective members of large populations. *Ann. Math. Statist.*, 14(4):436–440, 1943. [doi:10.1214/aoms/1177731363] 3

[22] Dingzhu Du and Frank K. Hwang: *Combinatorial Group Testing and Its Applications*. Applied Mathematics. World Scientific, 2000. 3

[23] Devdatt P. Dubhashi and Desh Ranjan: Balls and Bins: A study in negative dependence. *Random Structures Algorithms*, 13(2):99–124, 1996. [doi:10.1002/(SICI)1098-2418(199809)13:2<99::AID-RSA1>3.0.CO;2-M] 13

[24] MOEIN FALAHATGAR, ASHKAN JAFARPOUR, ALON ORLITSKY, VENKATADHEERAJ PICHAPATHI, AND ANANDA THEERTHA SURESH: Faster algorithms for testing under conditional sampling. In *Proc. 28th Ann. Conf. on Learning Theory (COLT'15)*, JMLR Proceedings, pp. 607–636, 2015. [arXiv:1504.04103] 4, 5, 7

[25] ELDAR FISCHER: The art of uninformed decisions: A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, 2001. 2

[26] ODED GOLDREICH, editor. *Property Testing: Current Research and Surveys*. Springer, 2010. LNCS 6390. [doi:10.1007/978-3-642-16367-8] 2

[27] ODED GOLDREICH AND DANA RON: On testing expansion in bounded-degree graphs. Technical report, Electron. Colloq. on Comput. Complexity (ECCC), 2000. [ECCC:TR00-020] 2, 4

[28] SUDIPTO GUHA, ANDREW MCGREGOR, AND SURESH VENKATASUBRAMANIAN: Sublinear estimation of entropy and information distances. *ACM Trans. Algorithms*, 2009. Preliminary version in SODA'06. [doi:10.1145/1597036.1597038, arXiv:cs/0508122] 2

[29] PIOTR INDYK, REUT LEVI, AND RONITT RUBINFELD: Approximating and testing k-histogram distributions in sub-linear time. In *Proc. 31st Symp. on Principles of Database Systems (PODS'12)*, pp. 15–22, 2012. [doi:10.1145/2213556.2213561, ECCC:TR11-171] 2

[30] REUT LEVI, DANA RON, AND RONITT RUBINFELD: Testing properties of collections of distributions. *Theory of Computing*, 9(8):295–347, 2013. Preliminary version in ICS'11. [doi:10.4086/toc.2013.v009a008, ECCC:TR10-157] 2

[31] RAJEEV MOTWANI AND PRABHAKAR RAGHAVAN: *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995. 12

[32] HUNG Q. NGO AND DING-ZHU DU: A Survey on Combinatorial Group Testing Algorithms with Applications to DNA Library Screening. In *DIMACS Ser. in Discr. Math. and Theoret. Comput. Sci.*, number 2, 2000. 3

[33] LIAM PANINSKI: A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Trans. Inform. Theory*, 54(10):4750–4755, 2008. [doi:10.1109/TIT.2008.928987] 2, 4, 7

[34] SOFYA RASKHODNIKOVA, DANA RON, AMIR SHPILKA, AND ADAM SMITH: Strong lower bounds for approximating distributions support size and the distinct elements problem. *SIAM J. Comput.*, 39(3):813–842, 2009. Preliminary version in FOCS'07. [doi:10.1137/070701649] 5

[35] DANA RON: Property Testing: A Learning Theory Perspective. *Foundations and Trends in Machine Learning*, 1(3):307–402, 2008. Preliminary version in COLT'07. [doi:10.1561/2200000004] 2

[36] DANA RON: Algorithmic and analysis techniques in property testing. *Found. Trends Theor. Comput. Sci.*, 5:73–205, 2010. [doi:10.1561/0400000029] 2

[37] DANA RON AND GILAD TSUR: The power of an example: Hidden set size approximation using group queries and conditional sampling. *ACM Trans. Comput. Theory*, 8(4):15:1–15:19, 2016. [doi:10.1145/2930657, arXiv:1404.5568] 6, 8

[38] RONITT RUBINFELD: Taming Big Probability Distributions. *XRDS*, 19(1):24–28, 2012. [doi:10.1145/2331042.2331052] 2

[39] RONITT RUBINFELD AND ROCCO A. SERVEDIO: Testing monotone high-dimensional distributions. *Random Structures Algorithms*, 34(1):24–44, January 2009. Preliminary version in STOC'05. [doi:10.1002/rsa.20247] 2

[40] BURR SETTLES: Active learning literature survey. CS Technical Reports 1648, University of Wisconsin–Madison, 2009. Available at MINDS@UW. 3

[41] LARRY J. STOCKMEYER: On approximation algorithms for #P. *SIAM J. Comput.*, 14(4):849–861, 1985. [doi:10.1137/0214060] 8

[42] GREGORY VALIANT AND PAUL VALIANT: A CLT and tight lower bounds for estimating entropy. *Electron. Colloq. on Comput. Complexity (ECCC)*, 2010. [ECCC:TR10-179] 5, 7, 45

[43] GREGORY VALIANT AND PAUL VALIANT: Estimating the unseen: A sublinear-sample canonical estimator of distributions. *Electron. Colloq. on Comput. Complexity (ECCC)*, 2010. [ECCC:TR10-180] 45

[44] GREGORY VALIANT AND PAUL VALIANT: The power of linear estimators. In *Proc. 52nd FOCS*, pp. 403–412, October 2011. See also [42] and [43]. [doi:10.1109/FOCS.2011.81] 5

[45] GREGORY VALIANT AND PAUL VALIANT: An automatic inequality prover and instance optimal identity testing. *SIAM J. Comput.*, 46(1):429–455, 2017. Preliminary version in FOCS'14. [doi:10.1137/151002526] 2, 4

[46] PAUL VALIANT: Testing symmetric properties of distributions. *SIAM J. Comput.*, 40(6):1927–1968, 2011. Preliminary version in STOC'08. [doi:10.1137/080734066, ECCC:TR07-135] 5

[47] WIKIPEDIA: Yes, Virginia, there is a Santa Claus — Wikipedia, The Free Encyclopedia, 2017. `https://en.wikipedia.org/w/index.php?title=Yes,_Virginia,_there_is_a_Santa_Claus&oldid=805832621` [Online; accessed 05-November-2017]. 2

AUTHORS

Jayadev Acharya
Assistant professor
Cornell University
acharya@cornell.edu
`http://people.ece.cornell.edu/acharya/`

Clément L. Canonne
Postdoctoral researcher
Stanford University
ccanonne@cs.stanford.edu
http://www.cs.columbia.edu/~ccanonne/

Gautam Kamath
Microsoft research fellow
Simons Institute for the Theory of Computing
Berkeley, CA
g@csail.mit.edu
http://www.gautamkamath.com/

ABOUT THE AUTHORS

JAYADEV ACHARYA is an assistant professor at Cornell University, where he joined after a postdoc at MIT. He holds a Ph. D. from UC San Diego, where he was advised by Alon Orlitsky. His research interests are in algorithms, information theory, machine learning, and statistics. His awards include a Jack Wolf Student paper award at ISIT'10, and a Best Paper Honorable Mention at ICML'17.

CLÉMENT L. CANONNE is a Motwani Postdoctoral Fellow at Stanford University. He graduated from Columbia University in 2017, where he was advised by Rocco Servedio. His research focuses on the fields of property testing and sublinear algorithms; specifically, on understanding the strengths and limitations of the standard models in property and distribution testing, as well as in related areas. He also really likes elephants.

GAUTAM KAMATH is a Microsoft Research Fellow at the Simons Institute for the Theory of Computing. He graduated from MIT in 2018, where he was advised by Constantinos Daskalakis. His research focuses on algorithms for statistical tasks (such as distribution testing and parameter estimation), particularly with considerations which arise in modern settings for data analysis (including high dimensions, robustness, and privacy). He was awarded the inaugural STOC Best Student Presentation award at STOC 2012.