Optimal mass estimation in the conditional sampling model

Tomer Adar* Eldar Fischer† Amit Levi‡

October 12, 2025

Abstract

The conditional sampling model, introduced by Canonne, Ron and Servedio (SODA 2014, SIAM J. Comput. 2015) and independently by Chakraborty, Fischer, Goldhirsh and Matsliah (ITCS 2013, SIAM J. Comput. 2016), is a common framework for a number of studies concerning strengthened models of distribution testing. A core task in these investigations is that of estimating the mass of individual elements. The above mentioned works, and the improvement of Kumar, Meel and Pote (AISTATS 2025), provided polylogarithmic algorithms for this task.

In this work we shatter the polylogarithmic barrier, and provide an estimator for the mass of individual elements that uses only $O(\log\log N) + O(\mathrm{poly}(1/\varepsilon))$ conditional samples. We complement this result with an $\Omega(\log\log N)$ lower bound.

We then show that our mass estimator provides an improvement (and in some cases a unifying framework) for a number of related tasks, such as testing by learning of any label-invariant property, and distance estimation between two (unknown) distributions. In light of some known lower bounds for common restricted models, our results imply that the full power of the conditional model is indeed required for the doubly-logarithmic upper bound.

Finally, we exponentially improve the previous lower bound on testing by learning of label-invariant properties from double-logarithmic to $\Omega(\log N)$ conditional samples, whereas our testing by learning algorithm provides an upper bound of $O(\mathrm{poly}(1/\varepsilon) \cdot \log N \log\log N)$.

*Technion - Israel Institute of Technology, Israel. Email: tomer-adar@campus.technion.ac.il.

†Technion - Israel Institute of Technology, Israel. Email: eldar@cs.technion.ac.il. Research supported by an Israel Science Foundation grant number 879/22.

‡University of Haifa, Israel. Email: alevi@cs.haifa.ac.il.

Contents

1 Introduction

The property testing framework [GGR98, RS96] deals with approximate decision making in situations where the input data cannot be read in its entirety. Instead, the algorithm is only allowed to read a very small fraction of the data and deduce some global property based on the observed information.

A well-investigated area of property testing focuses on examining the properties of distributions. In this context, the algorithm can access independent samples from a discrete distribution over $\{1, \ldots, N\}$, and must determine whether to accept or reject the input based on these samples. Specifically, the algorithm receives a parameter $\varepsilon > 0$ and is required to accept any input that meets the property to be tested (with high probability), while rejecting any input that is ε-far (in terms of total variation) from any distribution that fulfills the property (again, with high probability). This model was explicitly defined in [BFR+00, BFF+01, GR11] and has garnered considerable attention over the past few decades.

Somewhat unsurprisingly, a typical sample complexity for distribution testing algorithms is $\widetilde{O}(N^{\delta})$ for some constant $\delta < 1$. Even for testing whether a distribution is uniform, one of the most basic and simple distribution properties, a tight bound of $\Theta(\sqrt{N}/\varepsilon^2)$ is known [Pan08, GR11]. When studying distributions supported over extremely large domains, this sample complexity effectively makes testing intractable. To circumvent this problem, several competing approaches were considered.

The first approach involves restricting the class of input distributions (e.g., restricting the input distribution to be monotone [RS09] or a product distribution [CDKS17, DDK19]). The second approach considers a model equipped with a more relaxed distance metric (usually coupled with an even weaker query model), such as the Huge Object Model [GR23, CFG+23, AF24, AFL24b, CFG+24], which uses the earth-mover distance metric (as defined in those works). The third approach, which is the main focus of this work, investigates stronger query models.

One of the earliest models suggested to tackle the scaling problem is the *conditional sampling* model. This model was introduced independently by Chakraborty, Fischer, Goldhirsh, and Matsliah [CFGM16], and Canonne, Ron, and Servedio [CRS15]. The conditional model allows more general queries: namely, the algorithm may specify an arbitrary subset of the domain and request a sample from the distribution conditioned on it belonging to the subset. In many cases, the conditional sampling model circumvents sample-complexity lower bounds. Since its introduction, there has been significant study into the complexity of testing a number of properties of distributions under conditional samples, in both adaptive and non-adaptive settings [Can20, FJO+15, ACK18, BCG19, BC18, KT19, FLV19]. Beyond distribution testing, this model of conditional sampling has found applications in sublinear algorithms [GTZ17], group testing [ACK15], and crowdsourcing [GTZ18].

In this work we concentrate on a core task that is useful to many investigations of distribution testing. Consider a task that, given $x \in \{1, \ldots, N\}$, attempts to provide a multiplicative approximation of the probability of drawing x according to the input distribution μ, denoted by $\mu(x)$. It was first described as the *evaluation oracle* in [RS09]. If we are able to do it efficiently for all but a small probability set of the possible elements, then we can solve other tasks. Algorithms that simulate the multiplicative estimation task appear in [CRS15] and in [CFGM16]. Both works independently define an implementation with $\mathrm{poly}(\log(N), 1/\varepsilon)$ sample-complexity of the evaluation oracle and use it to show their results (equivalence testing in [CRS15], a universal tester for label-invariant

properties in [CFGM16]). Our main contribution is a radically improved algorithm for this task, that uses only $\log(\log(N))$ many samples, with a polynomial dependency on ε and an additional approximation parameter that we de-couple from ε and specify later.

As an example application, consider the task of *distance estimation* in the conditional sampling model. In this task the algorithm receives (conditional) sampling access to two unknown distributions μ and τ supported over $\{1, \ldots, N\}$, and is required to estimate their total-variation distance within an additive error parameter. In the standard sampling model, tight bounds of $\Theta(N/\log N)$ are known even for estimating the distance from uniformity [VV11, VV10a, VV10b].

When considering the conditional sampling model, one can do much better. In [CRS15, MKP25], an algorithm using $O(\mathrm{poly}(\log N)/\mathrm{poly}(\varepsilon))$ conditional samples was established. Later, [FJO+15] improved the complexity in the easier task of equivalence testing, which is distinguishing between zero distance and a distance greater than the approximation parameter, to $\tilde{O}(\log\log N/\varepsilon^5)$, and [CCK24] obtained a corresponding lower bound of $\tilde{\Omega}(\log\log N)$.

In this paper we apply our improved distribution approximator to drastically improve the upper bound for estimating the distance between two unknown distributions using conditional samples to $O(\log\log N/\varepsilon^2 + 1/\varepsilon^7) \cdot \mathrm{poly}\log(\varepsilon^{-1})$. Based on our core approximator, we also improve the polynomial ε-factors of [FJO+15] and eliminate the polynomial triple-logarithmic N-factors. We also use this enahnced estimation module to approximate the histogram of an unknown distribution, being optimal up to poly-double-logarithmic N-factors and polynomial ε-factors, and use it to obtain a universal tester for every label-invariant property at this cost.

To complement the picture, we show that there exists a label-invariant property that requires $\Omega(\log N/\varepsilon)$ samples to test, which implies that the above-mentioned universal tester is optimal up to polynomial double logarithmic N-factors and polynomial ε-factors. We also show that the core approximation task in itself is nearly optimal in its number of samples.

1.1 Summary of our results

Table of results The following table summarizes our results, except for the two lower bounds marked by "(∗)" which are due to [CCK24]. The following paragraphs provide more details. In the estimation task of $\mu(x)$, a correct output (with high probability) is guaranteed for every x in some set $G \subseteq \Omega$ satisfying $\mu(G) > 1 - c$. This task has two sample-complexity upper bounds: the first holds for every x in the domain of μ (even if it does not belong to G), and the second is the expected sample complexity when x is unconditionally drawn from μ.

Task	Lower Bound	Upper Bound	
Estimate $(1 \pm \varepsilon)\mu(x)$	$\Omega(\log\log N)$	all x	$O(\log\log N) + \tilde{O}(\frac{1}{\varepsilon^2 c} + \frac{1}{\varepsilon^5})$
		$x \sim \mu$	$O(\log\log N) + O(\frac{1}{\varepsilon^4}) \cdot \mathrm{polylog}(c^{-1}, \varepsilon^{-1})$
$\mu = \tau$ vs. $d_{\mathrm{TV}}(\mu, \tau) > \varepsilon$	(∗) $\tilde{\Omega}(\log\log N)$	$O\left(\frac{\log\log N}{\varepsilon} + \frac{1}{\varepsilon^5}\right) \cdot \mathrm{polylog}\,\varepsilon^{-1}$	
Estimate $d_{\mathrm{TV}}(\mu, \tau) \pm \varepsilon$	(∗) $\tilde{\Omega}(\log\log N)$	$O\left(\frac{\log\log N}{\varepsilon^2} + \frac{1}{\varepsilon^7}\right) \cdot \mathrm{polylog}\,\varepsilon^{-1}$	
Learn histogram of μ	$\Omega(\log N/\varepsilon)$	$\tilde{O}(1/\varepsilon^7) \cdot \log N \log\log N$	
Label-invariant universal tester	$\Omega(\log N/\varepsilon)$ (worst property)		

Single-element mass estimation This is our core contribution. We use a new approach to show a tight upper bound for estimating the mass of a given element within $(1 \pm \varepsilon)$-factor in the fully conditional (adaptive) model, where the mass of the set of eligible elements is at least $1 - c$ for a second approximation parameter c (all prior works implicitly set $c = O(\varepsilon)$, which is sufficient for most applications). The old approaches (for example descending the tree of dyadic intervals as in [CFGM16]) all have poly-logarithmic factors, which are unavoidable since they are also implementable in more restricted conditional models (such as interval conditioning [CRS15] and subcube conditioning [BC18]) in which equivalence testing is known to be poly-logarithmic hard.

Theorem 1.1 (Informal statement of Theorem 4.1)**.** *Let μ be a distribution over $\Omega = \{1, \ldots, N\}$ that is accessible through the fully conditional oracle, and let $\varepsilon, c > 0$ be the given approximation parameters. There exists a set $G \subseteq \Omega$ of mass $\mu(G) > 1 - c$ such that for every $x \in G$, we can algorithmically estimate $\mu(x)$ within $(1 \pm \varepsilon)$-factor with probability $2/3$. The sample complexity is bounded by $O(\log \log N) + \tilde{O}\left(\frac{1}{\varepsilon^2 c} + \frac{1}{\varepsilon^5}\right)$. If x is drawn from μ, then the expected sample complexity is only $O(\log \log N) + O\left(\frac{1}{\varepsilon^4}\right) \cdot \mathrm{poly}(\log c^{-1}, \log \varepsilon^{-1})$.*

The "$O(\log \log N)$" part comes from a binary search performed over a range of size $O(\log N)$. By a reduction from a binary search problem back to an estimation task, we also show that this part of the bound is tight (even if we only need to estimate a "typical" element from its support).

Theorem 1.2 (Informal statement of Theorem 10.22)**.** *There exists $\varepsilon > 0$, so that every algorithm, that can with probability at least $2/3$ estimate the probability mass of an element drawn (unconditionally) from μ within $(1 \pm \varepsilon)$-multiplicative factor, must draw at least $\Omega(\log \log N)$ conditional samples in expectation.*

We leverage Theorem 1.1 to obtain the following upper-bound results.

Equivalence ε-testing We improve the state-of-the-art upper bound of [FJO+15] that uses $\tilde{O}(\log \log N/\varepsilon^5)$ for ε-testing equivalence of two distributions over $\{1, \ldots, N\}$ in the fully conditional sampling model. We provide two asymptotical speedups: first, the $1/\varepsilon^5$-factor becomes additive rather than multiplicative, whereas $\log \log N$ is only multiplied by $1/\varepsilon$. This is an ε^4-speedup over the previous best for N that is large enough with respect to ε. Second, we remove the poly-triple-logarithmic factors over N, leaving a clean $O(\log \log N)$ dependency on the domain size.

Theorem 1.3 (Almost-tight upper bound for equivalence testing)**.** *Let μ, τ be two distributions over $\Omega = \{1, \ldots, N\}$ and $\varepsilon > 0$. There exists an algorithm for distinguishing between the case where $\mu = \tau$ and the case where $d_{\mathrm{TV}}(\mu, \tau) > \varepsilon$, using $O((\log \log N/\varepsilon + 1/\varepsilon^5) \cdot \mathrm{poly}(\log \varepsilon^{-1}))$ conditional samples.*

Distance estimation between two distributions We show an almost-tight upper bound for estimating the total variation distance between two distributions over $\{1, \ldots, N\}$ in the fully conditional model. The surprising aspect of this result is the almost-tight gap between the lower bound of the equivalence testing problem and the upper bound of the (harder) distance estimation problem. Having only a small gap depending on the domain size between testing and estimation tasks is relatively uncommon, and in various models there exist examples for polynomial gaps (for example, uniformity in the sampling model [Pan08, VV10a, VV11]), and even examples for constant-cost tests with non-constant cost corresponding estimation tasks (for example, [FF06] in the string model, later improved in [BEFLR20]).

Theorem 1.4 (Almost-tight upper bound for distance estimation)**.** *Let μ, τ be two distributions over $\Omega = \{1, \ldots, N\}$ and $\varepsilon > 0$. There exists an algorithm for estimating $d_{\mathrm{TV}}(\mu, \tau)$ within ε-additive error using $O((\log\log N/\varepsilon^2 + 1/\varepsilon^7) \cdot \mathrm{poly}(\log \varepsilon^{-1}))$ conditional samples.*

Since the distance estimation task cannot be easier than equivalence testing, whose lower bound is $\tilde{\Omega}(\log\log N)$ [CCK24], this upper bound for distance estimation is optimal, up to poly-triple-logarithmic factors of N and polynomial ε-factors.

Learning the histogram of μ We show an upper bound for learning the histogram of a given distribution μ up to a threshold parameter ε using a quasi-logarithmic in N number of conditional samples. Since histogram learning directly implies testing of any label-invariant property without additional samples, this improves over [CFGM16], whose label-invariant tester only guarantees a polylogarithmic upper bound. A nearly-matching lower bound for this task follows from specially constructed label-invariant properties (see below).

Theorem 1.5 (Informal statement of Theorem 9.21)**.** *There exists an algorithm that approximates the histogram of an input distribution μ with accuracy ε, using $\tilde{O}(1/\varepsilon^7) \cdot \log N \log\log N$ conditional samples.*

Theorem 1.6 (Informal statement of Corollary 10.52)**.** *For every sufficiently small $\varepsilon > 0$, every algorithm that approximates the histogram of its input distribution with accuracy ε must draw at least $\Omega(\log N/\varepsilon)$ conditional samples.*

Almost tight label-invariant testing The histogram learning algorithm immediately implies a corresponding test for any label-invariant property: one can just perform the histogram approximation up to a distance of $\varepsilon/2$, and then accept if and only if this histogram corresponds to a distribution that is $\varepsilon/2$-close to satisfying the property. We complement this with an existence proof of label-invariant properties with a nearly matching lower bound on the number of required samples, an exponential improvement over the $\tilde{\Omega}(\log\log N)$ bound recently shown in [CCK24].

Theorem 1.7 (Informal statement of Corollary 9.22)**.** *There exists a universal tester for ε-testing every label-invariant property using $\tilde{O}(1/\varepsilon^7) \cdot \log N \log\log N$ conditional samples.*

Theorem 1.8 (Informal statement of Theorem 10.51)**.** *For every sufficiently small $\varepsilon > 0$, there exists a label-invariant property $\mathcal{P}$ such that every ε-testing algorithm for $\mathcal{P}$ draws at least $\Omega(\log N/\varepsilon)$ conditional samples.*

1.2 Related work

Closely related to the distance estimation problem is the problem of *equivalence testing*, which asks to determine whether two unknown distributions are equal or far from each other. In the standard sampling model, the sample complexity of the problem was pinned down to $\Theta(N^{2/3}/\varepsilon^{4/3} + \sqrt{N}/\varepsilon^2)$ [BFR+00, Val08, CDVV14]. In the conditional sampling model, Canonne Ron and Servedio [CRS15] designed a testing algorithm with query complexity $\tilde{O}(\log^5 N/\varepsilon^4)$. This was later improved to $\tilde{O}(\log\log N/\varepsilon^5)$ by [FJO+15], and complemented with an almost matching lower bound of $\tilde{\Omega}(\log\log N)$ [CCK24]. That lower bound can be used for a relatively easy derivation of a $\tilde{\Omega}(\log\log N)$ lower bound on the (c, ε)-estimation task for small enough $\varepsilon > 0$ and $c > 0$, but we directly prove a clean $\Omega(\log\log N)$ bound.

One interesting special case of the distance estimation problem is the case where one of the distributions is explicitly given to the algorithm. In this setting, [CRS15] showed that one can estimate the distance to the known distribution using $\tilde{O}(\log^5 N/\varepsilon^5)$ conditional queries, which was later improved by [Nar20] to $\tilde{O}(1/\varepsilon^4)$. In contrast, in the standard sampling model, estimating the distance to the uniform distribution requires at least $\Omega(N/\log N)$ samples [VV11, VV10a, VV10b].

A special type of conditional access which gained popularity in recent years is the *subcube* conditioning model [BC18, CRS15]. In this model, the distributions are given over a product set $\{0,1\}^n$, and the algorithm can query subcube subsets, which are sets of the form $\prod_{i=1}^{n} D_i$ where $D_i \subseteq \{0,1\}$ for every $1 \leq i \leq n$ (note that here $N = 2^n$). In this model, uniformity can be tested using $\tilde{\Theta}(\sqrt{n}/\varepsilon^2) = \tilde{\Theta}\left(\sqrt{\log N}/\varepsilon^2\right)$ samples [CCK+21], and the best-known test for equivalence is $\tilde{O}(n/\varepsilon^2)$ [AFL24a], with a lower bound of $\Omega(n^{3/4}/\varepsilon + \sqrt{n}/\varepsilon^2)$ [CDKS17]. Other properties studied under the subcube conditional model include monotonicity [CCR+25], and having a probability density function supported on a low-dimensional subspace [CJLW21].

A related line of work aims to circumvent the polynomial dependency in the domain's size by considering restricted classes of input distributions. Some of the cases studied are those where the distribution is known to be monotone [RS09, CCR+25], a low-degree Bayesian Network [CDKS17, DP17, ABDK18], a Markov Random Field [DDK19, GLP18, BBC+20], or having a "histogram by intervals" structure [DKP19]. Considering a distribution having a histogram structure, a learning algorithm was given in [FLV19] under several sampling models (for such distributions there is little difference between learning the histogram and learning the entire distribution).

2 Overview

2.1 Technical overview

The core result

Given a distribution μ over $\Omega = \{1, \ldots, N\}$, an element $x \in \Omega$ and two estimation parameters $\varepsilon, c \in (0,1)$, our task is to estimate $\mu(x)$ within a $(1 \pm \varepsilon)$-factor or to indicate that it is among the smallest elements, whose cumulative mass is at most c. Note that if $\mu(x) = \Omega(c)$ then it can easily be estimated directly using unconditional sampling, and therefore, this overview focuses on the case where $\mu(x) = O(c)$ (with an appropriate hidden constant factor).

At top level, our algorithm looks for a reference set R whose probability mass (as an event) is both comparable to $\mu(x)$ (that is, $\Pr_\mu[x|R \cup \{x\}]$ lies in a reasonable range, between two constants) and estimable with high accuracy. This way we can arithmetically estimate $\mu(x)$ using estimations of $\Pr_\mu[x|R \cup \{x\}]$ and $\mu(R)$. For estimating $\Pr_\mu[x|R \cup \{x\}]$, it should be possible to efficiently draw samples from $R \cup \{x\}$. This can be directly done under the conditional model if we know R in its entirety, but as we describe below there are ways around this problem when we do not have complete access to R.

Our construction refers to two sets: the *target set* V_x, which is a set that includes all elements with mass smaller than $\mu(x)$ and no element whose mass is significantly higher than $\mu(x)$, and the *filter set* A_α, which is the result of independently choosing every $z \in \Omega$ with probability α. The intersection $V_x \cap A_\alpha$ is a good reference set whenever the order of α is about the quotient of $\mu(x)$ and $\mu(V_x)$. The algorithm spends most of its effort on finding a good α. In fact, if a good α is

already given, then the rest of the algorithm can complete its estimation of $\mu(x)$ using a number of samples that does not depend on N at all.

The target set V_x, whose mass is $\Omega(c)$ (for x whose cumulative mass is at least c and for which $\mu(x) = O(c)$), is estimable directly by virtue of having a high mass. Since it does not contain elements much heavier than x, we can use a large deviation inequality to deduce that the mass of $V_x \cap A_\alpha$ is highly concentrated around $\alpha\mu(V_x)$.

However, V_x cannot be found explicitly. We can only construct a "membership-oracle" by comparing the weight of potential elements to the weight of x using pair conditionals. In particular, V_x is probabilistic, but for any element y with strict demands (either lighter than x or much heavier than x) there is a very small probability to misclassify the membership of y. For medium-weight elements, which are only slightly heavier than x, our analysis embraces the probabilistic nature of belonging to V_x.

Another problematic consequence of the oracle-membership implicit-construction of V_x (and thereby of $R = V_x \cap A_\alpha$) is the inability to use it as a condition, since we can only restrict to explicit sets. Instead, we restrict to A_α (whose construction uses internal randomness and no samples from μ), and use rejection-sampling to simulate the restriction to $V_x \cap A_\alpha$. Since V_x has a globally high weight but contains no elements whose mass is too high, the relative weight of $V_x \cap A_\alpha$ as a subset of A_α is usually high as well. This allows a lazy construction of V_x, where we only query candidate elements drawn from A_α for belonging to V_x, instead of drawing V_x in its entirety in advance.

To find an α of the correct magnitude, we first observe that it suffices to consider powers of $\frac{1}{2}$ that lie between 1 and $\frac{1}{O(N)}$. This observation reduces the search range to $O(\log N)$ possible choices. To reduce the needed work to $O(\log \log N)$, we show a monotone estimable function of α that can characterize the range of good αs based on their respective values of this function. This allows a binary search, but since the estimation of the function is probabilistic, we construct a binary search scheme that allows the comparator to be wrong with small fixed probability. Our binary search scheme removes the triple-logarithmic penalty required by the straight-forward approach of amplifying the success probability of the comparator to the point that even a single error is unlikely to occur during the binary search.

The result of the binary search is a $\Theta(1)$-approximation of the ideal choice of α. Referring to R constructed using such an α, a simple arithmetical function of $\mu(R)/\mu(R \cup \{x\})$ (which can be approximated by inspecting a sequence of samples from $R \cup \{x\}$) gives us the missing factor that allows us to calculate the approximation of $\mu(x)$. Our procedure uses the roughly-estimated α to estimate this expectation within $1 \pm O(\varepsilon)$-factor, which we then use to obtain a $(1 \pm \varepsilon)$-factor estimation of $\mu(x)$.

The applications

We present three applications of our core estimator. All of them are the result of plugging our estimator (each time with different parameters and under different circumstances) into an algorithm that achieves the corresponding task when it has some access to the actual values of the distribution function μ.

For the task of histogram learning, knowing the exact value of $\mu(x)$ for each x that was received as an unconditional sample would have allowed us to just approximate the weight of each "bucket" B_i that

contains all x of weight between $(1-\varepsilon)^{i-1}$ and $(1-\varepsilon)^i$ (ignoring buckets with $i > O(\varepsilon^{-1}\log N)$). From the bucket weights one can then write down a distribution that is an approximation of a permutation of μ, providing the histogram of μ. Receiving only approximate values can cause some "bucket shift" to (say) $i \pm 2$, but the resulting error is not significant.

Estimating the distance between two distributions, for which element mass estimations are provided, is generally achievable by (unconditionally) drawing elements from the distributions, and for each drawn element examining the ratio of its masses according to the two distributions. To implement this in the conditional sampling model, we plug in our estimator. Since we need to use it also for elements which were not drawn from the distribution to be queried, this increases the dependency on ε to one that follows from the "all x" bound. However, when we only want to solve the equivalence testing task, we show that we can still use the "$x \sim \mu$" bound as long as both distributions are identical to the same μ, which allows us to automatically reject if the algorithm happens to require more samples than that bound. For both tasks, the lower bound is that of [CCK24].

The lower bounds

The tight lower bound for the estimation task is essentially an ad-hoc reduction from the task of finding an unknown value k, whose range of possible values has size $O(\log N)$, through binary search. Such a value can be "encoded" by a uniform distribution over a subset of $\{1,\ldots,N\}$ whose size is $(1 \pm o(1))2^{-k} \cdot N$, and then retrieved by successfully approximating the mass of any of the support elements. Since the bottleneck of the upper bound algorithm is a binary search task as well, this implies that a binary search task (in an appropriate range) is indeed a crucial component of the estimation task.

The demonstration that the full conditional model is essential for a doubly logarithmic algorithm follows from using the framework of some of our applications "in the other direction": a doubly logarithmic solution to the estimation task in a weak model would have implied a solution to a testing task that contradicts a known lower bound.

A label-invariant property with a logarithmic in N lower bound is constructed by encoding maximally hard to test linear codes as histograms, and proving that a test for such a code can be converted to a classical string property test in this case.

2.2 Organization of the paper

Section 3 describes the sampling model and the notation scheme that we use throughout this paper. Within it, Subsection 3.2 defines the quantities and constructed sets used for our algorithm. Appendix A provides a concise table for these, for the reader's convenience.

Sections 4 through 8 contain the proof of our core result. Appendix B provides a simplified chart of the calling structure and dependencies between the various procedures defined in these sections.

Section 4 provides the top layer of the algorithm, which is the procedure Estimate-element.

Section 5 provides the implementation of the target test scheme (Target-test in Subsection 5.1), along with a few algorithmic tools to assess the target sets, and specifically tools related to the quantity $\beta_{x,\alpha} = \mu(R)/\mu(R \cup \{x\})$: a "cheaper" Target-test-gross in Subsection 5.1, a virtualization of the target

set (Initialize-new-V_x, V_x-Query) in Subsection 5.2, Estimate-E$[\beta_{x,\alpha}]$ in Subsection 5.3, and Estimate-$\beta_{x,\alpha}$ in Subsection 5.4 (for a virtual single instance of V_x). The estimator here, while satisfying the optimal asymptotic guarantees, has an unrealistic numerical constant factor. In Appendix C we show how to reduce this by adjusting the target test, at the cost of a small asymptotic penalty which carries over to the estimator.

The following sections provide the three main components of the Estimate-element procedure: Reference-estimation in Section 6, Find-good-α in Section 7 and Estimate-scaled-result (an estimator for $\mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right]$) in Section 8. The estimator for $\mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right]$, whose precise definition is $\mathrm{E}_{A_\alpha,V_x}\left[\frac{\beta_{x,\alpha}(A_\alpha,V_x)}{1-\beta_{x,\alpha}(A_\alpha,V_x)}\right]$, uses the procedures for draws of V_x and estimations of $\beta_{x,\alpha}$ (for a provided A_α) of Section 5.

We provide the uncertain-comparator binary search used in Find-good-α, which is a probabilistic variant of binary search that can use a comparator which is allowed to be wrong with a small fixed probability, in Subsection 7.3.

In Section 9 we provide applications of our core result in the fully conditional model: in Subsection 9.2 we provide an almost-tight histogram learning algorithm, and as a corollary a universal ε-testing algorithm for label-invariant properties. In Subsection 9.3 we provide an almost tight ε-estimation for total-variation distance, and in Subsection 9.4 we provide an improved ε-test for equivalence. These applications use a trio of general-purpose application lemmas, Lemmas 9.10, 9.23 and 9.24. In Appendix D we provide another lemma of this type, for which we hope to be useful in future applications.

In Section 10 we provide lower bounds for the tasks discussed in this paper. In Subsection 10.1 we prove the tight lower bound on the $\mu(x)$ estimation task (as a function of N). Then, in Subsection 10.2 we provide quick lower bounds for this task under more restricted conditional models, mostly derived from known lower bounds on equivalence testing in conjunction with interim algorithms from Section 9. In Subsection 10.3 we construct a specific label-invariant property, for which we prove an almost-tight testing lower bound.

The most technical (and mechanical) proofs across the paper are deferred to Appendix E (bounds relevant to $\mathrm{E}[\beta_{x,\alpha}]$ and $\mathrm{E}[\beta_{x,\alpha}/(1-\beta_{x,\alpha})]$) and Appendix F (miscellaneous ad-hoc proofs).

3 Preliminaries

3.1 Distribution access oracles and tasks

In this work we consider algorithms that access an input *indirectly* through oracles. In particular, the complexity of our algorithms is measured in terms of the number of calls to the provided oracle. In all oracles, the output distribution is determined by the distribution μ and the arguments of the call, and is completely independent of past calls and any other algorithmic behavior.

The following is the weakest oracle, the one allowed in the traditional distribution testing model.

Definition 3.1 (Sampling oracle)**.** Let μ be an input distribution over a set Ω. The *sampling oracle* for μ has no additional input, and outputs an element $x \in \Omega$ that distributes like μ.

The following oracle corresponds to the algorithms that we analyze in this paper.

Definition 3.2 (Conditional sampling oracle)**.** Let μ be an input distribution over a set Ω. The *conditional sampling oracle* for μ gets a set $A \subseteq \Omega$ as input, and outputs an element $x \in A$ that distributes like μ when conditioned on x belonging to A.

The above definition still leaves open the question as to what happens when a sample conditioned on A is requested for a probability zero set A (two common variants are the oracle returning a special error symbol [CRS15] or the oracle returning a value uniformly drawn from A [CFGM16]). Our estimation algorithm is designed to never ask for such a sample, and hence works for all variants of this model.

We next define some notions of distances and approximations.

Definition 3.3 (Total variation distance)**.** Let μ and τ be two distributions over Ω. Their *total variation distance* is defined as

$$d_{\mathrm{TV}}(\mu,\tau) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{x\in\Omega}|\mu(x)-\tau(x)| = \sum_{x\in\Omega:\mu(x)>\tau(x)}(\mu(x)-\tau(x)) = \max_{E\subseteq\Omega}(\mu(E)-\tau(E))$$

Definition 3.4 (ε-test)**.** Let $\mathcal{R}$ be a metric space and let $\mathcal{P}$ be a closed set, which we call a *property*. For an input element $x \in \mathcal{R}$ and a parameter $\varepsilon > 0$, the goal of an *ε-test* for $\mathcal{P}$ is to distinguish between the case where $x \in \mathcal{P}$ and the case where $d(x, y) > \varepsilon$ for every $y \in \mathcal{P}$.

The following notion is a major building block in our algorithms.

Definition 3.5 (Saturation-aware estimator)**.** Let $f : [0, 1] \to [0, 1]$ be a non-decreasing monotone function. An algorithm is an $(\varepsilon; p_\ell, p_m)$-$f$-saturation-aware estimator of an unknown probability p if the following hold:

- If $f(p) \le p_\ell$, then with probability at least $2/3$ the output is a special value LOW.
- If $p_\ell < f(p) < p_m$, then with probability at least $2/3$ the output is either the special value LOW or in the range $(1 \pm \varepsilon)p$.
- If $f(p) \ge p_m$, then with probability at least $2/3$ the output is in the range $(1 \pm \varepsilon)p$.

We use "$(\varepsilon; p_\ell, p_m)$-saturation-aware estimator" to denote the case where f is the identity function.

Usually, a test can distinguish between the first and the third cases, and an estimator can guarantee the third part. The saturation-aware estimator also requires correctness in the "middle" case.

Recall that our main goal is to estimate the probability mass of individual elements. The exact mass cannot be provided since it can be any value in the continuous range $[0, 1]$. Moreover, the effort needed to estimate the mass of an extremely rare element is unbounded. Hence, every estimation algorithm must allow a small mass of elements whose probabilities cannot be approximated at all, and in this paper these are characterized by the notion of the *cumulative distribution function*.

Definition 3.6 (Cumulative distribution function)**.** Let μ be a distribution over Ω. The *cumulative distribution function* of μ is the function $\mathrm{CDF}_\mu : \Omega \to [0, 1]$ defined as $\mathrm{CDF}_\mu(x) = \Pr_{y\sim\mu}[\mu(y) \le \mu(x)]$.

Definition 3.7 (The (ε, c)-estimation task)**.** Let $\varepsilon > 0$ and $c > 0$ be our parameters. For a distribution μ over a finite domain Ω, let $\mathcal{A}$ be an algorithm that gets $x \in \Omega$ and outputs some $\hat{p}$. The goal is an $(\varepsilon; 0, c)$-CDF_μ-saturation-aware estimation of $\mu(x)$.

Our main result is an algorithm that solves the estimation task, whose dependency on N is doubly-logarithmic for fixed ε and c. Note that in particular the set $G = \{x \in \Omega : \mathrm{CDF}_\mu(x) \geq c\}$ has mass strictly larger than $1 - c$. We next describe a major application of our estimator.

Definition 3.8 (Histogram divergence $D_{\mathrm{H}}(\cdot;\cdot)$)**.** Let μ and τ be two distributions over Ω, and let $S(\Omega)$ denote the set of all permutations over Ω. The *histogram divergence* of μ and τ is defined as:

$$D_{\mathrm{H}}(\mu;\tau) = \min\{\varepsilon \geq 0 : \min_{\pi \in S(\Omega)} \Pr_{x\sim\mu}[\mu(x) \notin (1 \pm \varepsilon)\tau(\pi(x))] \leq \varepsilon\}$$

The following lemma, which we prove in Appendix F, states that distributions with low histogram divergence are close up to a permutation of the labels.

Lemma 3.9. *For every two distributions μ, τ over Ω there exists a permutation π over Ω for which $d_{\mathrm{TV}}(\mu, \pi\tau) \leq 2D_{\mathrm{H}}(\mu;\tau)$.*

Definition 3.10 (The ε-histogram learning task)**.** Let μ be a distribution over Ω. The *ε-histogram learning task* requires finding a distribution τ for which $D_{\mathrm{H}}(\mu;\tau) \leq \varepsilon$.

3.2 Paper-specific notations

To describe our estimation algorithm we need various ad-hoc notations. Most of them involve $x \in \Omega$, $c \in (0,1)$ and $\varepsilon \in (0,1)$, and some additionally involve $0 < \alpha \leq 1$. We usually use short-form notations ignoring c and ε, but never ignore x and α. See Appendix A for a concise table that summarizes these notations.

Given $x \in \Omega$ for which we would like to assess $\mu(x)$, our proofs rely heavily on a categorization of $\Omega \setminus \{x\}$ by masses.

Definition 3.11 (The three scale-sets with respect to x)**.** Let $x \in \Omega$. We divide the rest of the domain Ω according to their probability masses as compared to $\mu(x)$ as follows:

- The x-light set is $L_x = \{y \in \Omega \setminus \{x\} : \mu(y) \leq \mu(x)\}$.
- The x-medium set is $M_x = \{y \in \Omega : \mu(x) < \mu(y) < 1.2\mu(x)\}$.
- The x-heavy set is $H_x = \{y \in \Omega : \mu(y) \geq 1.2\mu(x)\}$.

Distinguishing between L_x-elements and H_x-elements cannot be certain since it uses random samples. We require the probability to be very small. The affect of the target error on our algorithm's complexity is logarithmic.

Definition 3.12 ($\eta_{c,\varepsilon}$, the target error)**.** The *target error* is $\eta_{c,\varepsilon} = \min\left\{\frac{1}{4}c\varepsilon, \frac{1}{10^9}, \frac{\varepsilon^5}{10^{20}(\ln \varepsilon^{-1})^5}\right\}$.

We define the constraints of the categorization algorithm based on the target error.

Definition 3.13 (An (x,c,ε)-target test)**.** An algorithm $\mathcal{T}$ is an *(x,c,ε)-target-test* if:

- The probability to accept $y \in \Omega \setminus \{x\}$ only depends on x and y (and μ), and in particular is independent of past executions.
- For every $y \in L_x$, the probability to accept y is greater than $1 - \eta_{c,\varepsilon}$.
- For every $y \in H_x$, the acceptance probability is less than $\eta_{c,\varepsilon}$.

Definition 3.14 (An (x, c, ε)-target-test scheme)**.** A mapping from every triplet (x, c, ε) to an (x, c, ε)-target-test $\mathcal{T}_{x,c,\varepsilon}$ is called an (x, c, ε)-*target-test scheme.*

The target-test scheme is a crucial part in our proof, and can be implemented algorithmically.

Lemma 3.15 (Informal statement of Lemma 5.3)**.** *Procedure* Target-test *(Algorithm 3 in Section 5), with parameters $(\mu, c, \varepsilon; x, y)$, is an (x, c, ε)-target-test scheme.*

We prove this lemma in Section 5, right after the implementation of Target-test. Based on this lemma, we have a canonical target-test for every triplet (x, c, ε), and can omit the recurring parameter of "the specific (x, c, ε)-test we refer to", encapsulating it as a function.

Definition 3.16 ($f_{x,c,\varepsilon}$, Target function)**.** The target function $f_{x,c,\varepsilon} : \Omega \setminus \{x\}$ is the probability of the (canonical) (x, c, ε)-target-test to accept y.

The "ideal reference set" is defined similarly to the output of the target test, only here we do not allow any error with respect to the "important" sets L_x and H_x.

Definition 3.17 ($V_{x,c,\varepsilon}$, the target set)**.** Let $x \in \Omega$. The target set $V_{x,c,\varepsilon} \subseteq \Omega \setminus \{x\}$ is a random set that contains L_x, is disjoint from H_x, and contains every element $y \in M_x$ with probability $f_{x,c,\varepsilon}(y)$, independently.

We define two masses based on the expected mass of the target set corresponding to x, with or without x itself. Both masses play a role.

Definition 3.18 ($s_{x,c,\varepsilon}$, the scale mass)**.** The scale mass with respect to x is denoted by $s_{x,c,\varepsilon} = \mathrm{E}[\mu(V_{x,c,\varepsilon})] = \mu(L_x) + \sum_{y \in M_x} \mu(y) f_{x,c,\varepsilon}(y)$. The expectation is over the choice of $V_{x,c,\varepsilon}$ as a random set.

Definition 3.19 ($w_{x,c,\varepsilon}$, the weight of x)**.** The weight of x is denoted by $w_x = \mu(x) + s_{x,c,\varepsilon}$.

As mentioned in the technical overview, we also define a filter set that is randomly constructed according to a parameter α. Since the filter set only depends on the internal coin-tosses of the algorithm, we can fully characterize it and use it for drawing conditional samples. Our reference set is the intersection of the target set and the filter set.

Definition 3.20 (A_α, the α-filter set)**.** Let $0 < \alpha \leq 1$. The α-filter set, A_α, is a random set where every element in Ω belongs to A_α with probability α, independently.

Definition 3.21 ($V_{x,c,\varepsilon,\alpha}$, the α-filtered target set)**.** The α-filtered target set is denoted by $V_{x,c,\varepsilon,\alpha} = V_{x,c,\varepsilon} \cap A_\alpha$.

The next definition, $\gamma_{x,c,\varepsilon}$, describes the "best" value of α with respect to x, c and ε, which our algorithm looks for. The rest of the algorithm works by first finding a suitable $\alpha = \Theta(\gamma_{x,c,\varepsilon})$

Definition 3.22 ($\gamma_{x,c,\varepsilon}$, the goal magnitude)**.** The goal magnitude of the filtering parameter α is denoted by $\gamma_{x,c,\varepsilon} = \mu(x) / \mathrm{E}[\mu(V_{x,c,\varepsilon})]$.

For a good α, the distribution of the following probability is concentrated around a value bounded away from both 0 and 1.

Definition 3.23 ($\beta_{x,c,\varepsilon,\alpha}$, the filtered density)**.** Let $x \in \Omega$ and $0 < \alpha \leq 1$. The filtered density of x, with respect to the choices of $V_{x,c,\varepsilon}$ and A_α, is $\beta_{x,c,\varepsilon,\alpha} = \Pr_\mu[\neg x | V_{x,c,\varepsilon,\alpha} \cup \{x\}] = \frac{\mu(V_{x,c,\varepsilon,\alpha})}{\mu(x) + \mu(V_{x,c,\varepsilon,\alpha})}$.

The algorithm first finds a good α by performing a binary search, where for values too close to 0 or 1 the expectation of $\beta_{x,c,\varepsilon,\alpha}$ are in the "too low" and "too high" ranges respectively. A good α allows us to complete our assessment of $\mu(x)$ using $\beta_{x,c,\varepsilon,\alpha}$.

Observation 3.24. *For every $0 < \alpha \leq 1$, $\mu(x) = \alpha \operatorname{E}[\mu(V_{x,c,\varepsilon})] / \operatorname{E}\left[\frac{\beta_{x,c,\varepsilon,\alpha}}{1-\beta_{x,c,\varepsilon,\alpha}}\right]$.*

Proof. For every fixed x,

$$\operatorname{E}\left[\frac{\beta_{x,c,\varepsilon,\alpha}}{1-\beta_{x,c,\varepsilon,\alpha}}\right] = \operatorname{E}\left[\frac{\frac{\mu(V_{x,c,\varepsilon,\alpha})}{\mu(x)+\mu(V_{x,c,\varepsilon,\alpha})}}{1-\frac{\mu(V_{x,c,\varepsilon,\alpha})}{\mu(x)+\mu(V_{x,c,\varepsilon,\alpha})}}\right] = \operatorname{E}\left[\frac{\mu(V_{x,c,\varepsilon,\alpha})}{\mu(x)}\right] = \frac{\operatorname{E}[\mu(V_{x,c,\varepsilon,\alpha})]}{\mu(x)} = \frac{\alpha \operatorname{E}[\mu(V_{x,c,\varepsilon})]}{\mu(x)} \quad \square$$

3.3 Technical lemmas

Definition 3.25 (Binomial distribution, $\operatorname{Bin}(n,p)$)**.** The distribution of the sum of n independent trials with success probability p is denoted by $\operatorname{Bin}(n,p)$. Explicitly, $\Pr_{\operatorname{Bin}(n,p)}[k] = \binom{n}{k} p^k (1-p)^{n-k}$.

Lemma 3.26 (Additive Chernoff bound)**.** *Let $X \sim \operatorname{Bin}(n,p)$. For every $t > 0$, $\Pr[X - \operatorname{E}[X] > t] \leq e^{-2t^2/n}$ and $\Pr[X - \operatorname{E}[X] < -t] \leq e^{-2t^2/n}$.*

Lemma 3.27 (Multiplicative Chernoff bound)**.** *Let $X \sim \operatorname{Bin}(n,p)$. For every $0 < r \leq 1$, $\Pr[X > (1+r)\operatorname{E}[X]] \leq e^{-\frac{1}{3}r^2 \operatorname{E}[X]}$ and $\Pr[X < (1-r)\operatorname{E}[X]] \leq e^{-\frac{1}{3}r^2 \operatorname{E}[X]}$.*

Definition 3.28 (Geometric distribution, $\operatorname{Geo}(p)$)**.** The distribution of the number of independent trials, each with success probability p, until the first success (including the successful trial itself) is denoted by $\operatorname{Geo}(p)$. Explicitly, $\Pr_{\operatorname{Geo}(p)}[k] = (1-p)^{k-1}p$.

Lemma 3.29 (Well-known)**.** *Let X be a random variable that is geometrically distributed with parameter p. Then $\operatorname{E}[X] = p^{-1}$, $\operatorname{Var}[X] = \frac{1-p}{p^2}$, and $\operatorname{E}[e^{\lambda X}] = \frac{pe^{\lambda}}{1-(1-p)e^{\lambda}}$ for $\lambda < -\ln(1-p)$.*

Observation 3.30 (Folklore)**.** *Let $A_\alpha \subseteq \Omega$ be a random set such that, given α, every element $y \in \Omega$ belongs to A_α with probability $p_{y,\alpha}$, where $p_{y,\alpha}$ is non-decreasing monotone with respect to α (but possibly not the same for different choices of y). Let $f : 2^\Omega \to \mathbb{R}$ be a non-decreasing monotone function (that is, $U \subseteq V \to f(U) \leq f(V)$). In this setting, the mapping $\alpha \to \operatorname{E}[f(A_\alpha)]$ is non-decreasing monotone as well.*

Observation 3.31 (Generic)**.** *Let $f : 2^\Omega \to [a,b]$ be a bounded function. Assume that $A_\alpha \subseteq \Omega$ is drawn such that every element $y \in U$ is drawn with probability $p_{y,\alpha}$, which is continuous with respect to a parameter α (but possibly not the same for different choices of y). The mapping $\alpha \to \operatorname{E}[f(A_\alpha)]$ is continuous.*

Proof. Let $\alpha_1 < \alpha_2$. Then:

$$|\operatorname{E}[f(A_{\alpha_2})] - \operatorname{E}[f(A_{\alpha_1})]| \leq \left(\max_U f(U) - \min_U f(U)\right) \sum_{y \in \Omega} |p_{y,\alpha_2} - p_{y,\alpha_1}| \leq (b-a) \cdot \sum_{y \in \Omega} |p_{y,\alpha_2} - p_{y,\alpha_1}|$$

This expression tends to zero for $\alpha_2 \to \alpha_1$ since all $p_{y,\alpha}$s are continuous and a and b are fixed. $\square$

Observation 3.32 (Median amplification)**.** *Let X be a random variable, and $[a,b]$ be a range such that $\Pr[X \in [a,b]] \geq 2/3$. We use "median-of-$M$" to denote the process of drawing M independent samples of X and taking their median value. Then:*

(a) Median-of-9 amplifies the probability of obtaining a value in $[a,b]$ to $5/6$.

(b) Median-of-13 amplifies it to $8/9$.

(c) Median-of-47 amplifies it to $99/100$.

(d) Median-of-$\lceil 30 \ln c^{-1} \rceil$ amplifies it to $1 - \frac{1}{2}c$ for $c < 1/3$.

(e) Median-of-$\lceil 30 \ln c^{-1} \rceil$ amplifies it to $1 - \frac{1}{24}c$ for $c < 1/150$.

We prove Observation 3.32 in Appendix F.

Lemma 3.33. *Let $r > 0$. For every distribution μ, $\mathrm{E}_{x\sim\mu}\left[\frac{1}{w_x + r}\right] \leq \mathrm{E}_{x\sim\mu}\left[\frac{1}{\mathrm{CDF}_\mu(x)+r}\right] = O(\log r^{-1})$.*

Proof. By definition, $w_x \geq \mathrm{CDF}_\mu(x)$ for every $x \in \Omega$.

$$\begin{aligned}
\mathrm{E}\left[\frac{1}{w_x + r}\right] \leq \mathrm{E}\left[\frac{1}{\mathrm{CDF}_\mu(x) + r}\right] &\leq \sum_{x\in\Omega} \mu(x) \cdot \frac{1}{\max\{\mathrm{CDF}_\mu(x), r\}} \\
&\leq \Pr[\mathrm{CDF}_\mu(x) \leq r] \cdot \frac{1}{r} + \sum_{t=0}^{\lfloor \log_2 r^{-1} \rfloor} \Pr[\mathrm{CDF}_\mu(x) \leq 2^{-t}] \cdot 2^t \\
&\leq r \cdot \frac{1}{r} + \sum_{t=0}^{\lfloor \log_2 r^{-1} \rfloor} 2^{-t} \cdot 2^t \\
&= 1 + \sum_{t=0}^{\lfloor \log_2 r^{-1} \rfloor} 1 = O(\log r^{-1})
\end{aligned}$$

□

Due to the length of some expressions in our proofs, we use here the contribution notation introduced in [AFL24b]:

Definition 3.34 (Contribution of X over B)**.** Let X be a random variable and B be an event. We denote the *contribution of X over B* by $\mathrm{Ct}[X|B] = \sum_{x\in B} \Pr[x] \cdot X(x) = \Pr[B]\,\mathrm{E}[X|B]$.

We quickly summarize some equalities of the contribution notation:

- $\mathrm{Ct}[\alpha X + \beta Y|B] = \alpha\,\mathrm{Ct}[X|B] + \beta\,\mathrm{Ct}[Y|B]$.
- If $B_1 \cap B_2 = \emptyset$ then $\mathrm{Ct}[X|B_1 \cup B_2] = \mathrm{Ct}[X|B_1] + \mathrm{Ct}[X|B_2]$.
- If $\Pr[X = Y|B] = 1$ then $\mathrm{E}[X] - \mathrm{E}[Y] = \mathrm{Ct}[X - Y|\neg B]$.

4 Our algorithm

Our upper-bound statements assume that ε and c are "sufficiently small". More concretely, $\varepsilon < \frac{1}{10}$ and $c < \frac{1}{16}$. The acronym "SP" appearing in some of the algorithms refers to "Success Probability".

We state the main theorem of this paper, which refers to the correctness of the procedure Estimate-element.

Theorem 4.1. *For every individual $x \in \Omega$, Algorithm 1 solves the (ε, c)-estimation task with expected sample complexity $O(\log\log N) + O\left(\log\frac{1}{\varepsilon c} \cdot \left(\frac{1}{\varepsilon^2(w_x+c)} + \frac{\log^5 \varepsilon^{-1}}{\varepsilon^4(w_x+\varepsilon/\log\varepsilon^{-1})}\right)\right)$ (the expectation is over the random choices of the algorithm), where $N = |\Omega|$ is the size of the domain of μ.*

Corollary 4.2. *The expected complexity of Algorithm 1 is $O(\log\log N) + O\left(\log\frac{1}{\varepsilon c} \cdot \left(\frac{1}{\varepsilon^2 c} + \frac{\log^6 \varepsilon^{-1}}{\varepsilon^5}\right)\right)$ for the worst-case choice of $x \in \Omega$.*

Proof. The worst case is trivially $w_x = 0$. □

Corollary 4.3. *The expected complexity of Algorithm 1, where x is the result of an unconditional sample from μ, is $O(\log\log N) + O\left(\log\frac{1}{\varepsilon c} \cdot \left(\frac{\log c^{-1}}{\varepsilon^2} + \frac{\log^6 \varepsilon^{-1}}{\varepsilon^4}\right)\right)$.*

Proof. By Lemma 3.33, the expected value of $\frac{1}{w_x+c}$ is bounded by $O(\log c^{-1})$ and the expected value of $\frac{1}{w_x+\varepsilon/\log\varepsilon^{-1}}$ is $O(\log\varepsilon^{-1})$. □

The algorithmic demonstration of Theorem 4.1 (Algorithm 1 below) relies on three core subroutines, whose interface is stated in the following lemmas.

The first lemma, proved in Section 6, provides an estimation of the expected mass of the target set. Additionally, for the edge-case of elements with very high mass, it estimates this mass directly.

Lemma 4.4 (Reference-estimation)**.** *For every $x \in \Omega$, Algorithm 10 is a joint estimator of $(\mu(x), s_x)$ which is:*

- *An $\left(\varepsilon; \max\left\{\frac{1}{400}c, \frac{1}{400}s_x\right\}, \max\left\{c, \frac{1}{4}s_x\right\}\right)$-saturation-aware estimator for $\mu(x)$.*
- *A $\left(\frac{1}{3}\varepsilon; \max\left\{\frac{1}{400}c, \frac{1}{400}\mu(x)\right\}, \max\left\{c, \frac{1}{4}\mu(x)\right\}\right)$-saturation-aware estimator for s_x.*

Its expected cost is $O\left(\log\frac{1}{\varepsilon c} \cdot \frac{1}{\varepsilon^2(w_x+c)}\right)$ samples.

The second lemma, proved in Section 7, provides us with a parameter α that would be usable for comparing $\mu(x)$ with the mass of the filtered target set $V_{x,\alpha}$.

Lemma 4.5 (Find-good-α)**.** *Assume that $\mu(x) \le \frac{1}{4}s_x$. The output of Algorithm 12 is a random variable α for which, with probability $2/3$, $\gamma_x \le \alpha \le 41\gamma_x$, at the cost of $O(\log\log N)$ samples at worst case.*

The third lemma, proved in Section 8, produces the relative estimation of the ratio of $\mu(x)$ to the expected mass of the filtered target set, provided we have started with a suitable α (essentially a very rough approximation of the ratio between $\mu(x)$ and the scale mass).

Lemma 4.6 (Estimate-scaled-result)**.** *Let $0 < \alpha \le 1$ be an explicitly given input, and assume that $\gamma_x \le \alpha \le 50\gamma_x$. The output of Algorithm 14 is a random variable whose value, with probability $2/3$, is $(1 \pm \varepsilon/2)\alpha s_x/\mu(x)$, at the expected cost of $O\left(\log \frac{1}{\varepsilon c} \cdot \frac{\log^5 \varepsilon^{-1}}{\varepsilon^4(w_x+\varepsilon/\log \varepsilon^{-1})}\right)$ samples.*

At this point we provide Algorithm 1, and prove its correctness, which implies Theorem 4.1.

Algorithm 1: Procedure Estimate-element$(\mu, c, \varepsilon; x)$
Input: x.
Output: $\hat{p} \in \{\text{LOW}\} \cup (1 \pm \varepsilon)\mu(x)$.

1. Let $M \leftarrow 13$. For median amplification (Observation 3.32(b)).
2. For i from 1 to M:
   (a) Let $(\hat{p}_i, \hat{s}_i) \leftarrow$ Reference-estimation$(\mu, c, \varepsilon; x)$.
3. Set $\hat{p} \leftarrow \text{median}(\hat{p}_1, \ldots, \hat{p}_M)$ and $\hat{s} \leftarrow \text{median}(\hat{s}_1, \ldots, \hat{s}_M)$. SP: $\frac{8}{9}$
4. If $\hat{p} = \text{LOW}$ and $\hat{s} \neq \text{LOW}$:
   (a) For i from 1 to M:
      i. Let $\alpha_i \leftarrow$ Find-good-$\alpha(\mu, c, \varepsilon; x)$.
   (b) Let $\alpha \leftarrow \text{median}(\alpha_1, \ldots, \alpha_M)$. SP: $\frac{8}{9}$
   (c) For For i from 1 to M:
      i. Let $\hat{b}_i \leftarrow$ Estimate-scaled-result$(\mu, c, \varepsilon; x)$.
   (d) Let $\hat{b} \leftarrow \text{median}(\hat{b}_1, \ldots, \hat{b}_M)$. SP: $\frac{8}{9}$
   (e) Set $\hat{p} \leftarrow \alpha\hat{s}/\hat{b}$.
5. Return $\hat{p}$. **Total success probability: $\frac{2}{3}$**

Proof of Theorem 4.1. For the definition of the (ε, c)-estimation task with the required saturation-awareness bounds p_1 and p_2, let $p_1 = \min_x(\{c\} \cup \{\mu(x) : w_x \ge c\})$, and $p_2 = \max_x\{\mu(x) : w_x \le \frac{1}{300}c\}$ with a fallback of $p_2 = \frac{1}{2}p_1$ if this set is empty. Also, let $G = \{x : w_x \ge c\} \supseteq \{\text{CDF}_\mu(x) \ge c\}$ be the set of "good" xs.

For a given x, Algorithm 1 draws $O\left(\log\log N + \log \frac{1}{\varepsilon c} \cdot \left(\frac{1}{\varepsilon^2(w_x+c)} + \frac{\log^5 \varepsilon^{-1}}{\varepsilon^4(w_x+\varepsilon/\log \varepsilon^{-1})}\right)\right)$ samples in expectation:

- $O(1)$ calls to Reference-estimation (Lemma 4.4) at the expected cost of $O\left(\log \frac{1}{\varepsilon c} \cdot \frac{1}{\varepsilon^2(w_x+c)}\right)$.
- $O(1)$ calls to Find-good-α (Lemma 4.5) at the cost of $O(\log\log N)$.
- $O(1)$ calls to Estimate-scaled-result (Lemma 4.6), at total cost of $O\left(\log \frac{1}{\varepsilon c} \cdot \frac{\log^5 \varepsilon^{-1}}{\varepsilon^4\left(w_x+\frac{\varepsilon}{\log \varepsilon^{-1}}\right)}\right)$ in expectation.

Correctness (main case): assume that we obtain $\hat{p}$, $\hat{s}$ where at least one of them is not LOW, and each of them which is not LOW is a $(1\pm\varepsilon/3)$-factor estimation of its goal ($\mu(x)$ or s_x). This happens with probability at least $8/9$ if $x \in G$ ($w_x \ge c$).

If $\hat{p}$ is not LOW, then we just return it as a correct estimation. This happens with probability $8/9$ if $\mu(x) \ge \max\{c, \frac{1}{4}s_x\}$.

If $\hat{p}$ is LOW, then we have $\hat{s} = (1 \pm \varepsilon/3)s_x$. This happens with probability at least 8/9 if $s_x \geq \max\{c, \frac{1}{4}\mu(x)\}$, which is a superset of the constraint $(x \in G) \wedge (\mu(x) < \frac{1}{4}s_x)$. In this case, α is correct with probability 8/9 and $\hat{b}$ is correct with probability 8/9 as well. Overall, with probability 2/3, we have $\hat{s} = (1 \pm \varepsilon/3)s_x$ and $\hat{b} = (1 \pm \varepsilon/2)\alpha s_x/\mu(x)$, which means that Step 4e sets $\hat{p} = (1 \pm \varepsilon/3)(1 \pm \varepsilon/2)\mu(x) = (1 \pm \varepsilon)\mu(x)$ as desired.

Correctness (reject case): assume that we obtain $\hat{p}$, $\hat{s}$ which are both LOW. This happens with probability at least 8/9 if $w_x \leq \frac{1}{100}c$. In this case, we return LOW, which is a correct output for $x \notin G$.

Correctness (middle case): the saturation-awareness of Lemma 4.4 guarantees that, with probability 8/9, we obtain a pair $(\hat{p}, \hat{s})$ that correctly matches the main case or the reject case. □

Avoiding probability zero sets In Section 6 we explain how Reference-estimation$(\mu, c, \varepsilon; x)$ in itself takes samples only from sets that include an element that was already sampled (unconditionally) from μ, thus ensuring that they have positive probability. We also explain there why, in the case where $\mu(x) = 0$, Reference-estimation$(\mu, c, \varepsilon; x)$ always answers (LOW, LOW), causing Estimate-element$(\mu, c, \varepsilon; x)$ to skip the next steps and immediately return LOW. The other procedures used by Estimate-element$(\mu, c, \varepsilon; x)$ only take samples from sets that include x itself, and hence if they are invoked they only take samples from positive probability sets.

5 Target test assessment

5.1 The target test

We formulate the algorithm whose (randomized) output is used as part of the definition of the target function f_x. Procedure Target-test uses $O(\log \frac{1}{\varepsilon c})$ pair conditional samples to distinguish between $\mu(y) \leq \mu(x)$ and $\mu(y) \geq 1.2\mu(x)$ with probability at least $1 - \eta_{c,\varepsilon}$.

First, we provide Target-test-explicit (Algorithm 2) as a common logic for the actual target test, and a cheaper approximation of the target test that is used for finding a good α. For this implementation we use a hard-coded tuning parameter $\kappa = 10^{-9}/45$. In Appendix C we provide an alternative implementation of the target test with reasonable constant factors (removing the dependency on κ) but with an additional asymptotic penalty, which carries over to the estimator.

We use the common logic to define the target test and the approximate target test.

We provide some essential bounds of the explicit test.

Lemma 5.1. *In Algorithm 2,* $\Pr\left[\textsf{Target-test-explicit}(\mu, \eta; x, y) = \textsc{accept} \middle| t \notin \frac{\mu(y)}{\mu(x)+\mu(y)} \pm \kappa\right]$ *is at least* $1 - \eta$ *if* $\frac{\mu(y)}{\mu(x)+\mu(y)} \leq t - \kappa$ *and at most* η *if* $\frac{\mu(y)}{\mu(x)+\mu(y)} \geq t + \kappa$.

Proof. If $t \geq \frac{\mu(y)}{\mu(x)+\mu(y)} + \kappa$, then by Chernoff bound,

$$\Pr[Y \geq t\ell | t] \leq \Pr\left[\mathrm{Bin}\left(\ell, \frac{\mu(y)}{\mu(x)+\mu(y)}\right) \geq \left(\frac{\mu(y)}{\mu(x)+\mu(y)} + \kappa\right)\ell\right] \leq e^{-2\kappa^2\ell} \leq e^{-\ln \eta^{-1}} = \eta$$

Algorithm 2: Procedure Target-test-explicit$(\mu, \eta; x, y)$
Input: $y \in \Omega$.
Output: ACCEPT or REJECT.

1. If $y = x$: Technical guarantee
   (a) REJECT
2. Let $\kappa = 10^{-9}/45$.
3. Let $\ell \leftarrow \left\lceil \ln \eta^{-1}/2\kappa^2 \right\rceil$.
4. Draw $t \sim [\frac{1}{2} + \kappa, \frac{6}{11} - \kappa]$ uniformly.
5. Draw $z_1, \ldots, z_\ell$ independent samples from μ conditioned on $\{x, y\}$.
6. Let $Y = |\{i : z_i = y\}|$.
7. If $Y < t \cdot \ell$:
   (a) ACCEPT.
8. Else:
   (a) REJECT.

Algorithm 3: Procedure Target-test$(\mu, c, \varepsilon; x, y)$
Input: $y \in \Omega$.
Output: ACCEPT or REJECT.

1. Call Target-test-explicit$(\mu, \eta_{c,\varepsilon}; x, y)$ and return its answer.

Algorithm 4: Procedure Target-test-gross$(\mu; x, y)$
Input: $y \in \Omega$.
Output: ACCEPT or REJECT.

1. Call Target-test-explicit$(\mu, 10^{-9}; x, y)$ and return its answer.

If $t \le \frac{\mu(y)}{\mu(x)+\mu(y)} - \kappa$, then by Chernoff bound,

$$\Pr[Y < t\ell | t] \le \Pr\left[\mathrm{Bin}\left(\ell, \frac{\mu(y)}{\mu(x)+\mu(y)}\right) < \left(\frac{\mu(y)}{\mu(x)+\mu(y)} - \kappa\right)\ell\right] \le e^{-2\kappa^2\ell} \le e^{-\ln \eta^{-1}} = \eta \quad \square$$

Lemma 5.2. *In the setting of Algorithm 2,* $\Pr\left[t \in \frac{\mu(y)}{\mu(x)+\mu(y)} \pm \kappa\right] \le 10^{-9}$.

Proof. t is uniformly drawn in $[\frac{1}{2}+\kappa, \frac{6}{11}-\kappa]$. Hence, the probability that the segment $t \pm \kappa$ contains a given number p is at most $\frac{2\kappa}{6/11-1/2-2\kappa} \le 10^{-9}$. $\square$

These bounds allow us to prove the following two lemmas.

Lemma 5.3 (Target-test)**.** *Procedure Target-test uses* $O(\log \frac{1}{\varepsilon c})$ *conditional samples, accepts with probability at least* $1 - \eta_{c,\varepsilon}$ *if* $y \in L_x$ *and rejects with probability at least* $1 - \eta_{c,\varepsilon}$ *if* $y \in H_x$.

Proof. For $y \in L_x$: $\mu(y) \le \mu(x)$ and hence $t \ge \frac{1}{2} + \kappa \ge \frac{\mu(y)}{\mu(x)+\mu(y)} + \kappa$ with probability 1. By Lemma 5.1, Target-test$(\mu; x, y)$ accepts with probability at least $1 - \eta_{c,\varepsilon}$. That is, the difference from $\Pr[y \in V_x] = 1$ is bounded by $\eta_{c,\varepsilon}$.

For $y \in H_x$: $\mu(y) \geq 1.2\mu(x)$ and hence $t \leq \frac{6}{11} - \kappa \leq \frac{\mu(y)}{\mu(x)+\mu(y)} - \kappa$ with probability 1. By Lemma 5.1, Target-test$(\mu; x, y)$ accepts with probability at most $\eta_{c,\varepsilon}$. That is, the difference from $\Pr[y \in V_x] = 0$ is bounded by $\eta_{c,\varepsilon}$. □

Lemma 5.4 (Target-test-gross)**.** $|\Pr[y \in V_x] - \Pr[\textit{Target-test-gross}(\mu; x, y) = \textsc{accept}]| \leq 10^{-8}$ *for every* $x, y \in \Omega$.

Proof. We consider every y individually.

For $y \in L_x$: then $\Pr[y \in V_x] = 1$ and Target-test-gross accepts (x, y) with probability at least $1 - 10^{-9}$. The difference is bounded by 10^{-9}.

For $y \in H_x$: then $\Pr[y \in V_x] = 0$ and Target-test-gross accepts (x, y) with probability at most 10^{-9}. The difference is bounded by 10^{-9}.

For $y \in M_x$: then $\Pr[y \in V_x]$ is the probability of Target-test to accept (x, y). By Lemma 5.2, the probability that $t \in p_y \pm \kappa$ is bounded by 10^{-9}. In this case, our best bound for the variation in behaviors is 1. Otherwise, $t \notin p_y \pm \kappa$, and hence the variation in behaviors is bounded by $\max\{\eta_{\varepsilon,c}, 10^{-9}\}$ by Lemma 5.1.

Combined, for every $x, y \in \Omega$, the difference between the accept probability of Target-test and Target-test-gross is bounded by $10^{-9} + \max\{\eta_{c,\varepsilon}, 10^{-9}\} \leq 10^{-9} + 10^{-9} \leq 10^{-8}$. □

5.2 Individual drawing of V_x

The goal of this subsection is to provide membership query access to a drawing of a set V_x through the following interface:

- Initialize-new-$V_x(c, \varepsilon; x, q)$ – draws a secret set V according to a distribution that is $\eta_{c,\varepsilon} q$-close to the correct distribution of V_x, and returns an object that supports up to q queries.
- V_x-Query(obj, y) – reports whether $y \in V$ or not, where V is the set being held by the object obj. If the initialization parameter of the object is q, then only the first q calls are guaranteed to be meaningful. A query may affect the contents of obj.
- The overall sample complexity of the initialization followed by at most q queries is bounded by $O(\log \frac{1}{\varepsilon c}) \cdot q$.

The implementation of the interface is straightforward: during initialization, we initialize an empty list of "historical records", which we denote by *hist*. In every query of an element we first look for it in the list. If it exists there, then we report (again) the recorded result, and if it is missing, then we run the Target-test procedure to determine whether it belongs to the set, and record the answer in the list.

Algorithm 5: Procedure Initialize-new-$V_x(c, \varepsilon; x, q)$
1. Return $(c, \varepsilon, x, \textit{hist})$, where *hist* is an empty list.

In the rest of this subsection we show that Algorithm 5 and Algorithm 6 implement their desired interface guarantees.

Algorithm 6: Procedure V_x-Query(obj, y)
Input: An object *obj* created by Initialize-new-V_x representing a subset of Ω, and $y \in \Omega$.
Output: Whether or not y belongs to the set represented by the object.
Side effects: the *hist* component of *obj* may change.

1. Let c, ε, x, *hist* be the components of *obj* as a 4-tuple.
2. If $y = x$:
   (a) Return REJECT. ($x \in V_x$ never happens)
3. If *hist* contains (y, ans) for any *ans*: (y was queried before)
   (a) Return *ans*.
4. Else: (new y)
   (a) Let $ans \leftarrow$ Target-test$(c, \varepsilon; x, y)$.
   (b) Add (y, ans) to *hist*.
   (c) Return *ans*.

Lemma 5.5 below states that, from the caller's perspective, drawing V_x while answering one query at a time is logically equivalent to drawing it all at once.

Lemma 5.5 (The local-simulation lemma)**.** *Given $p_z \in [0, 1]$ for every $z \in \Omega$, Consider the following three methods of drawing a random set $U \subseteq \Omega$.*

1. *Every $z \in U$ is chosen to be in U with probability p_z, independently of all other members of U.*
2. *We start with an empty U. For up to q iterations, in the ith iteration we receive z_i (which may depend on the results of previous iterations), and then with probability p_{z_i} (independently of all previous results) add z_i to Ω. After this phase is over, every $y \in \Omega \setminus \{z_1, \ldots, z_q\}$ is added to U independently with probability p_y.*
3. *Same as Item 2, only here in the first phase we use p'_{z_i} instead of p_{z_i}, where $|p_{z_i} - p'_{z_i}| \leq \delta$ (in the second phase we still use the original p_y).*

The distribution of the sets as drawn in the first item or the second item (both phases) are identical, and are δq-close to the distribution of the set as drawn in the third item.

Proof. Methods 1 and 2 are identical since, regardless of the order of choices, every element z belongs to U with probability p_z independently of the others.

In every step $1 \leq i \leq q$, if the first $i - 1$ steps in the first phase of Method 2 and Method 3 were the same, then the probability of the ith step of Method 2 to deviate from the ith step of Method 3 is bounded by δ. By the union bound (and the second phase of both methods being the same), the distributions of a set drawn by Method 2 and a set drawn by Method 3 are are δq-close to each other. □

Lemma 5.6 (Initialize-new-V_x, V_x-Query)**.** *The distribution of the output of a sequence starting with a single call to Initialize-new-V_x (Algorithm 5), followed by q calls to V_x-Query (Algorithm 6) over the produced object with the queries $y_1, \ldots, y_q \in \Omega$, is $\eta_{c,\varepsilon} q$-close to the distribution of the output of a sequence that draws $V \sim V_{x,c,\varepsilon}$ and determines whether y_i belongs to V or not for every*

$1 \le i \le q$. *This bound holds also for an adaptive choice of every* y_i *based on the answers to the queries* $y_1, \ldots, y_{i-1}$. *The call to* Initialize-new-V_x *has no sample cost, and each call to* V_x-Query *costs* $O(\log \frac{1}{\varepsilon c})$ *conditional samples.*

Proof. Observe that Algorithm 5 and Algorithm 6 together implement Method 3 of Lemma 5.5 for drawing V_x with error parameter $\delta = \eta_{c,\varepsilon}$. Hence, its behavior is $\eta_{c,\varepsilon}$-close to an object that is initialized with an explicit drawing of V_x as a whole and then answers all membership queries (whether $y \in V_x$ for some y) with the same deviation probability as Method 3. □

5.3 Estimation of $\mathrm{E}[\beta_{x,\alpha}]$

Recall that, given A_α and V_x, we define $\beta_{x,\alpha} = \Pr_\mu[\neg x | V_{x,\alpha} \cup \{x\}]$.

Algorithm 7 estimates $\mathrm{E}[\beta_{x,\alpha}]$ as the expected value of the following indicator: first we draw A_α, and then we repeatedly draw y from μ conditioned on $A_\alpha \cup \{x\}$, until we hit an instance where $y = x$ or Algorithm Target-test accepts, or until we exceed a pre-defined iteration limit.

Algorithm 7: Procedure Estimate-$\mathrm{E}[\beta_{x,\alpha}]$ $(\mu, c, \varepsilon; x, \alpha)$
Output: $\hat{b} \in \mathrm{E}[\beta_{x,\alpha}] \pm \frac{1}{200}$.

1. Let $M \leftarrow 70000$.
2. Set $m \leftarrow 0$.
3. For i from 1 to M:
   (a) Draw A_α according to its definition.
   (b) Set $b_i \leftarrow 0$.
   (c) For 10000 iterations or until explicitly terminated:
      i. Draw y from μ, conditioned on $A_\alpha \cup \{x\}$.
      ii. If $y = x$:
         A. Exit FOR loop.
      iii. If Target-test-gross$(\mu; x, y)$ accepts:
         A. Set $b_i \leftarrow 1$.
         B. Exit FOR loop.
4. Let $\hat{b} \leftarrow \frac{1}{M} \sum_{i=1}^{M} b_i$.
5. Return $\hat{b}$.

Lemma 5.7 (Estimate-$\mathrm{E}[\beta_{x,\alpha}]$)**.** *Given* $0 < \alpha \le 1$, *Algorithm 7 estimates* $\mathrm{E}[\beta_{x,\alpha}]$ *within* $\frac{1}{200}$*-additive error with probability* $2/3$ *using* $O(1)$ *conditional samples.*

Proof. The worst-case cost of the algorithm is $O(1)$ calls to Target-test-gross, each costing $O(1)$ samples.

Consider the following hypothetical variants of Algorithm 7:

- Variant A. Algorithm 7 as written (the realizable variant).
- Variant B. A variant where instead of Target-test-gross$(\mu; x, y)$ we use a hypothetical (non-realistic) procedure that accepts y with probability equal to $\Pr[y \in V_x]$.

- Variant C. A variant where additionally to the hypothetical procedure used in Variant B, we also remove the iteration limit of the loop in Step 3c, running it until it is explicitly terminated.

Let r_A, r_B and r_C be the expected values of each of $b_1, \ldots, b_M$ (which are all distributed the same) in each respective variant. By definition (and $\hat{b}$ being the average of $b_1, \ldots, b_M$ and hence having the same expectation), r_A is also the expected output of Algorithm 7, and $r_C = \mathrm{E}[\beta_{x,\alpha}]$.

By the triangle inequality, $|\hat{b} - \mathrm{E}[\beta_{x,\alpha}]| = |\hat{b} - r_C| \leq |\hat{b} - r_A| + |r_B - r_A| + |r_C - r_B|$.

For every $1 \leq i \leq M$, b_i is an indicator and hence its variance is bounded by $\frac{1}{4}$. Since the b_is are independent, the variance of $\hat{b}$ is bounded by $\frac{1}{4M}$, and by Chebyshev's inequality,

$$\Pr\left[\left|\hat{b} - r_A\right| > \frac{1}{300}\right] = \Pr\left[\left|\hat{b} - \mathrm{E}[\hat{b}]\right| > \frac{1}{300}\right] \leq \frac{\frac{1}{4M}}{(1/300)^2} = \frac{22500}{70000} < \frac{1}{3}$$

By the union bound over 10000 iterations of the loop in Step 3c of Variant A and the corresponding one of Variant B, $|r_B - r_A| \leq 10000\,|\Pr[y \in V_x] - \Pr[\mathsf{Target\text{-}test\text{-}gross}(x, y) = \textsc{accept}]|$.

We use Lemma 5.4 to obtain that $|\Pr[y \in V_x] - \Pr[\mathsf{Target\text{-}test\text{-}gross}(x, y) = \textsc{accept}]|$ is bounded by $\frac{1}{10^8}$. Hence, $|r_B - r_A| \leq 10000 \cdot \frac{1}{10^8} < \frac{1}{5400}$.

The behaviors of Variant B and Variant C with respect to the definitions differ only when 10000 iterations of Step 3c are exceeded, and always $r_B \leq r_C$.

Let r be the probability to explicitly terminate the loop in an individual iteration. Observe that $r \geq r_C$ since we always terminate after writing $b_i \leftarrow 1$ (which happens with probability r_C), but we can also terminate when $y = x$. The number of iterations in Variant C distributes geometrically with parameter r, hence the total variation distance between a run of this loop in Variant B and in Variant C is also bounded by $\Pr[\mathrm{Geo}(r) > 10000]$. Combined with $0 \leq r_B \leq r_C \leq r$, we obtain $|r_C - r_B| \leq \min\{r, \Pr[\mathrm{Geo}(r) > 10000]\}$.

We use $\frac{1}{675}$ as an approximate break-even parameter. If $r \geq \frac{1}{675}$, then

$$\begin{aligned}
|r_B - r_C| \leq \Pr\left[\mathrm{Geo}(1/675) \geq 10000\right] &\leq \Pr\left[\mathrm{Geo}(1/675) \geq 14 \cdot 675\right] \\
&= \Pr\left[e^{\frac{1}{675}\mathrm{Geo}(1/675)} \geq e^{14}\right] \\
[\text{Markov}] \quad &\leq e^{-14}\,\mathrm{E}\left[e^{\frac{1}{675}\mathrm{Geo}(1/675)}\right] \\
[\text{Lemma 3.29}] \quad &= e^{-14}\frac{\frac{1}{675}e^{1/675}}{1 - \left(1 - \frac{1}{675}\right)e^{1/675}} \leq \frac{1}{675}
\end{aligned}$$

Hence, $|r_C - r_B| \leq \min\left\{\frac{1}{675}, \Pr\left[\mathrm{Geo}(1/675) \geq 10000\right]\right\} = \frac{1}{675}$.

Overall, with probability at least $\frac{2}{3}$, $\left|\hat{b} - \mathrm{E}[\beta_{x,\alpha}]\right| \leq \frac{1}{300} + \frac{1}{5400} + \frac{1}{675} = \frac{1}{200}$. □

5.4 Individual estimation of $\beta_{x,\alpha}$

The previous subsection provided a rough approximation of $\mathrm{E}[\beta_{x,\alpha}]$ where the expectation is taken over random choices of A_α and V_x, but in Section 8 we define some function h and need to approximate $\mathrm{E}[h(\beta_{x,\alpha})]$. Hence, we also need to estimate $\beta_{x,\alpha}$ for *specific* draws of A_α and V_x, where

the later is given through an output of a call to Initialize-new-V_x. In the following we show how to estimate $\beta_{x,\alpha}$ with respect to A_α and the set V_x that is virtually held by the V_x-object.

Algorithm 8: Procedure Estimate-$\beta_{x,\alpha}(\mu, c, \varepsilon; x, \alpha, \delta, A_\alpha, obj)$
Input: $\delta < \frac{1}{4}$.
Input: *obj* represents a subset of Ω that is the output of Initialize-new-V_x.
Output: $\hat{b} \in \beta_{x,\alpha}(A_\alpha, V_x) \pm \delta$.
Complexity: At most $25\frac{\ln(6/\delta)}{\delta^3}$ conditioned samples and one V_x-Query$(obj, \cdot)$ call per sample.

1. Let $M \leftarrow \lceil 8/\delta^2 \rceil$.
2. Set $m \leftarrow 0$.
3. For M times:
   (a) For $\lceil 3\ln(6/\delta)/\delta \rceil$ iterations or until explicitly terminated:
      i. Draw y from μ, conditioned on $A_\alpha \cup \{x\}$.
      ii. If $y = x$:
         A. Exit FOR loop.
      iii. If V_x-Query(obj, y) accepts:
         A. Set $m \leftarrow m + 1$.
         B. Exit FOR loop.
4. Let $\hat{b} \leftarrow m/M$.
5. Return $\hat{b}$.

Lemma 5.8. $\mathrm{E}\left[\frac{\mu(A_\alpha \cup \{x\})}{\mu((V_x \cap A_\alpha) \cup \{x\})}\right] \leq \frac{20}{w_x}$.

We prove Lemma 5.8 in Appendix E.

Lemma 5.9 (Estimate-$\beta_{x,\alpha}$)**.** *Let $0 < \delta < \frac{1}{4}$. Algorithm 8 outputs an estimation of $\beta_{x,\alpha}(A_\alpha, V_x)$ within additive error $\pm\delta$ with probability at least $\frac{2}{3}$. Its expected query complexity is bounded by $O\left(\log \frac{1}{\varepsilon c} \cdot \frac{1}{\delta^2(w_x + (\delta/\log \delta^{-1}))}\right)$. Additionally, the number of V_x-Query$(obj, \cdot)$-calls is bounded by $25\frac{\ln(6/\delta)}{\delta^3}$.*

Proof. The algorithm makes at most one V_x-Query$(obj, \cdot)$-call per sample. The number of samples is bounded by $\lceil 8/\delta^2 \rceil \cdot \lceil 3\ln(6/\delta)/\delta^3 \rceil$, which is at most $25\frac{\ln(6/\delta)}{\delta^3}$ for every $\delta < \frac{1}{4}$. Recall that every V_x-Query-call costs $O(\log \frac{1}{\varepsilon c})$ samples.

Clearly, the worst case cost of Algorithm 8 is $O(1/\delta^2) \cdot O(\ln \delta^{-1}/\delta) \cdot O\left(\log \frac{1}{\varepsilon c}\right) = O\left(\log \frac{1}{\varepsilon c} \cdot \frac{\log \delta^{-1}}{\delta^3}\right)$.

By Lemma 5.8, the expected number of inner-loop iterations is bounded by $O(1/w_x)$ and the expected complexity is bounded by $O\left(\log \frac{1}{\varepsilon c} \cdot \frac{1}{\delta^2 w_x}\right)$. Since the expected cost cannot exceed the worst-case cost, we can reformulate the expected cost as $O\left(\log \frac{1}{\varepsilon c} \cdot \frac{1}{\delta^2(w_x + \delta/\log \delta^{-1})}\right)$.

If the algorithm does not terminate the inner loop after $\lceil 3\ln(6/\delta)/\delta \rceil$ iterations, then the expected gain in m in every inner iteration would be exactly $\mathrm{E}\left[\frac{\mu((V_x \cap A_\alpha) \cup \{x\})}{\mu(A_\alpha \cup \{x\})}\right]$ for the given A_α and the set being held by *obj*. The actual gain is less, since we must consider the possibility to terminate the loop after $\lceil 3\ln(6/\delta)/\delta \rceil$. Let $\hat{\beta}$ be the expectation of this gain.

If $\mathrm{E}\left[\frac{\mu((V_x \cap A_\alpha) \cup \{x\})}{\mu(A_\alpha \cup \{x\})}\right] \leq \frac{1}{3}\delta$, then the additive penalty is at most $\frac{1}{3}\delta$. Otherwise, the distance between $\hat{\beta}$ and $\beta_{x,\alpha}$ is bounded by $\left(1-\frac{1}{3}\delta\right)^{\lceil 3\ln(6/\delta)/\delta \rceil} \leq \left(1-\frac{1}{3}\delta\right)^{3\ln(6/\delta)/\delta} \leq e^{-\ln(6/\delta)} = \frac{1}{6}\delta \leq \frac{1}{3}\delta$ as well.

Note that m is the sum of M independent indicators, hence its variance is bounded by $M/4$. Since $M \geq 8/\delta^2$, Chebyshev's inequality implies that with probability at least $\frac{2}{3}$, $m \in \mathrm{E}[m] \pm \frac{2}{3}\delta M$, and in this case, $\hat{b} = \mathrm{E}[\hat{b}] \pm \frac{2}{3}\delta = \beta_{x,\alpha}(A_\alpha, V_x) \pm \delta$. □

6 The reference estimator

In this section we prove Lemma 4.4 and provide an algorithm that demonstrates it. Towards this proof, we first provide a general saturation-aware estimator for the expectation of an indicator variable.

6.1 Estimation of an unknown probability

We implement here a $\left(\delta; \frac{1}{12}a, a\right)$-saturation-aware estimator for p, where p is only accessible through an oracle that draws an indicator random variable with expected value p. Algorithm 9 draws independent samples of this indicator and then stops after seeing $O(1/\delta^2)$ occurrences of 1 or until reaching the limit of $O(1/(\delta^2 a))$ samples.

Algorithm 9: Procedure SA-Est$(a; \mathcal{A}, \delta)$
Input: An oracle $\mathcal{A}$ for sampling a binary variable.
Output: $\hat{p} \in \{\textsc{low}\} \cup (1 \pm \delta)\,\mathrm{E}[\mathcal{A}]$.

1. Let $M \leftarrow \lceil 48/\delta^2 \rceil$.
2. Let $L \leftarrow \lfloor 6M/a \rfloor$.
3. Set $m \leftarrow 0$, $\ell \leftarrow 0$.
4. While $m < M$ and $\ell < L$:
   (a) Set $\ell \leftarrow \ell + 1$.
   (b) Draw $b \sim \mathcal{A}$. ($b \in \{0,1\}$).
   (c) Set $m \leftarrow m + b$.
5. If $m = M$: (Sufficiently many occurrences observed)
   (a) Set $\hat{p} \leftarrow M/\ell$.
6. Else: (Give-up limit reached)
   (a) Set $\hat{p} \leftarrow$ LOW.
7. Return $\hat{p}$.

Lemma 6.1 (SA-Est). *Assume that we have an oracle $\mathcal{A}$ that draws 1 with probability p and 0 with probability $1-p$, where every call is independent of past calls. For every $0 < a < 1$, Algorithm 9 with parameters $(a; \mathcal{A}, \delta)$ is a $(\delta; a/12, a)$-saturation-aware estimator for $p = \mathrm{E}[\mathcal{A}]$, at the expected cost of $O\left(\frac{1}{\delta^2(p+a)}\right)$ oracle calls. Moreover, $\mathrm{E}[\hat{p}^{-1} | \hat{p} \neq \textsc{low}] \leq p^{-1}$, where $\hat{p}$ is the output of the estimator.*

Proof. Let ℓ' be the theoretical value of ℓ if we would not terminate the loop after L iterations but let it continue until $m = M$. Observe that ℓ' distributes the same as the sum of M independent

geometric variables with parameter p. Also, $\Pr[\ell' = \ell | m = M] = 1$ (since we terminate the loop due to $m = M$ regardless of whether ℓ exceeds L or not) and $\Pr[\ell \le \ell'] = 1$ (since $\ell = \min\{\ell', L\}$).

Observe that the events "$m = M$" and "$\ell' \le 6M/a$" are equivalent: in the last iteration, we increased m from $M-1$ to M and also increased ℓ and ℓ' together (regardless whether the result reached L or not).

Using standard facts about geometric variables, $\mathrm{E}[\ell'] = M/p$ and $\mathrm{Var}[\ell'] \le M/p^2$. Thus the expected number of samples taken by the algorithm is $O\left(\frac{1}{\delta^2 p}\right)$, and the introduction of the hard bound $\ell \le L$ brings it down to the required $O\left(\frac{1}{\delta^2(p+a)}\right)$.

By Chebyshev's inequality,

$$\begin{aligned}\Pr\left[M/\ell' \notin (1\pm\delta)p\right] = \Pr\left[\ell' \notin \frac{1}{1\pm\delta}M/p\right] &\le \Pr\left[\ell' \notin \left(1 \pm \frac{1}{2}\delta\right)\mathrm{E}[\ell']\right] \\ &\le \frac{M/p^2}{(\delta/2)^2(M/p)^2} = \frac{4}{\delta^2 M} \le \frac{4}{\delta^2(48/\delta^2)} < 1/6\end{aligned}$$

We now go over the cases in the definition of a saturation-aware approximation.

Case I. $p \ge a$: by Markov's inequality, the probability to stop due to the give-up limit is bounded by $\mathrm{E}[\ell]/\lceil 6M/a\rceil \le (M/p)/(6M/p) = 1/6$, and the overall probability to have the correct output is at least $1 - \Pr[m = M] - \Pr[M/\ell' \notin (1\pm\delta)p] \ge 1 - 1/6 - 1/6 = 2/3$.

Case II. $a/12 < p < a$: the probability to return the wrong output is $\Pr[m = M \wedge M/\ell' \notin (1\pm\delta)p] \le \Pr[M/\ell' \notin (1\pm\delta)p] < \frac{1}{6} < \frac{1}{3}$.

Case III. $p \le a/12$: the "bad event" is $m = M$, which is equivalent to $\ell' \le 6M/a$. Hence, the probability to return any real number instead of LOW is:

$$\begin{aligned}\Pr[M = m] = \Pr[\ell' \le 6M/a] &= \Pr[\ell' - \mathrm{E}[\ell'] \le 6M/a - M/p] \\ &\le \Pr[\ell' - \mathrm{E}[\ell'] \le M/(2p) - M/p] \\ &= \Pr[\ell' - \mathrm{E}[\ell'] \le -M/(2p)] \\ [\text{Chebyshev}] \quad &\le \frac{M/p^2}{(M/(2p))^2} = \frac{4}{M} \le \frac{4}{48/\delta^2} < \frac{1}{3}\end{aligned}$$

Hence we correctly return LOW with probability at least $2/3$.

Lastly, observe that:

$$\mathrm{E}\left[\hat{p}^{-1} \middle| \hat{p} \ne \textsc{low}\right] = \mathrm{E}\left[\ell'/M \middle| m = M\right] = \frac{1}{M}\mathrm{E}\left[\ell' \middle| \ell' \le 6M/a\right] \le \frac{1}{M}\mathrm{E}\left[\ell'\right] = p^{-1} \qquad \square$$

6.2 The reference estimation procedure

We first roughly estimate $w_x = \mu(x) + s_x$, and based on the result, we estimate $\mu(x)$ and s_x using the magnitude of w_x as a reference.

We recall Lemma 4.4 and then prove it.

Algorithm 10: Procedure Reference-estimation$(\mu, c, \varepsilon; x)$
Output: $\hat{p}, \hat{s}$.

1. Let $M \leftarrow 13$. For median amplification (Observation 3.32(b)).
2. For i from 1 to M:
   (a) Let $\hat{w}_i \leftarrow$ SA-Est$(a = c - \eta_{c,\varepsilon}; \text{Oracle}, \delta = 1/3)$.
      i. Oracle: draw $y \sim \mu$. Return 1 if $y = x$ or Target-test$(\mu, c, \varepsilon; x, y)$ accepts.
3. Let $\hat{w} \leftarrow \text{median}(\hat{w}_1, \ldots, \hat{w}_M)$. SP: $\frac{8}{9}$
4. If $\hat{w} = \text{LOW}$:
   (a) Return (LOW, LOW).
5. For i from 1 to M:
   (a) Let $\hat{s}_i \leftarrow$ SA-Est$(a = \hat{w}/9; \text{Oracle}, \delta = \varepsilon/6)$.
      i. Oracle: draw $y \sim \mu$. Return 1 if $y \neq x$ and Target-test$(\mu, c, \varepsilon; x, y)$ accepts.
   (b) Let $\hat{p}_i \leftarrow$ SA-Est$(a = \hat{w}/9; \text{Oracle}, \delta = \varepsilon)$.
      i. Oracle: draw $y \sim \mu$. Return 1 if $y = x$.
6. Let $\hat{p} \leftarrow \text{median}(\hat{p}_i)$. SP: $\frac{8}{9}$
7. Let $\hat{s} \leftarrow \text{median}(\hat{s}_i)$. SP: $\frac{8}{9}$
8. Return $(\hat{p}, \hat{s})$. **Total success probability:** $\frac{2}{3}$

Lemma 4.4 (Reference-estimation)**.** *For every $x \in \Omega$, Algorithm 10 is a joint estimator of $(\mu(x), s_x)$ which is:*

- *An $\left(\varepsilon; \max\left\{\frac{1}{400}c, \frac{1}{400}s_x\right\}, \max\left\{c, \frac{1}{4}s_x\right\}\right)$-saturation-aware estimator for $\mu(x)$.*
- *A $\left(\frac{1}{3}\varepsilon; \max\left\{\frac{1}{400}c, \frac{1}{400}\mu(x)\right\}, \max\left\{c, \frac{1}{4}\mu(x)\right\}\right)$-saturation-aware estimator for s_x.*

Its expected cost is $O\left(\log\frac{1}{\varepsilon c} \cdot \frac{1}{\varepsilon^2(w_x+c)}\right)$ samples.

Proof. The expected value of the oracle in step 2(a)i (in Algorithm 10) is:

$$\begin{aligned}
\mu(x) + \sum_{y\in\Omega_x} \mu(y)f_x(y) &= \mu(x) + \sum_{y\in L_x} \mu(y)f_x(y) + \sum_{y\in M_x} \mu(y)f_x(y) + \sum_{y\in H_x} \mu(y)f_x(y) \\
(*) &= \mu(x) + (1 \pm \eta_{c,\varepsilon})\mu(L_x) + \sum_{y\in M_x} \mu(y)f(y) \pm \eta_{c,\varepsilon}\mu(H_x) \\
&= \mu(x) + \mu(L_x) + \sum_{y\in M_x} \mu(y)f_x(y) \pm \eta_{c,\varepsilon}(\mu(L_x) + \mu(H_x)) = w_x \pm \eta_{c,\varepsilon}
\end{aligned}$$

$(*)$: since $1 - \eta_{c,\varepsilon} \leq f(y) \leq 1$ for every $y \in L_x$ and $0 \leq f(y) \leq \eta_{c,\varepsilon}$ for every $y \in H_x$.

Also, by Lemma 6.1 (SA-Est), the sample complexity of this estimation is $O\left(\frac{1}{\varepsilon^2(\hat{w}+(c-\eta_{c,\varepsilon}))}\right) = O\left(\frac{1}{\varepsilon^2(w_x+c)}\right)$ oracle calls, since (again by Lemma 6.1) $\mathrm{E}[\hat{w}^{-1}|\hat{w} \neq \text{LOW}] \leq \sum_{i=1}^{M} \mathrm{E}[\hat{w}_i^{-1}|\hat{w}_i \neq \text{LOW}] \leq \frac{M}{w_x} = O\left(\frac{1}{w_x}\right)$.

If $w_x \geq c$, then the oracle's expected value is in the range $w_x \pm \eta_{c,\varepsilon} = w_x \pm \frac{1}{4}c\varepsilon = (1 \pm \varepsilon/2)w_x$, and hence with probability 8/9, $\hat{w} = (1 \pm 1/3)(1 \pm \varepsilon/2)w_x = (1 \pm 1/2)w_x$.

If $s_x \geq \max\{c, \frac{1}{4}\mu(x)\}$, then $c \leq s_x \leq w_x \leq 5s_x$. Hence, with probability at least 8/9, $\hat{w} = (1 \pm 1/2)w_x \leq 8s_x$. The expected value of the oracle in step 5(a)i is $s_x \pm \eta_{c,\varepsilon} = s_x \pm \frac{1}{4}\varepsilon c = (1 \pm \varepsilon/4)s_x \geq (1 - 1/40)s_x$, which is more than $\hat{w}/9 \leq \frac{8}{9}(s_x - \eta_{c,\varepsilon})$. In this case, the estimation outputs an $(1 \pm \varepsilon/6)$-estimation of $s_x \pm \eta_{c,\varepsilon}$ with probability at least 2/3. This estimation is in the range $(1\pm\varepsilon/6)(1\pm\varepsilon/4)s_x = (1\pm\varepsilon)s_x$. The cost of this estimation is bounded by $\mathrm{E}\left[\hat{w}^{-1}\middle|\hat{w} \neq \textsc{low}\right] = O\left(\frac{1}{w_x}\right) = O\left(\frac{1}{w_x+c}\right)$ oracle calls.

If $s_x \leq \frac{1}{400}\max\{c, \mu(x)\}$, then either $\hat{w}$ is LOW (and then we use $\hat{s} = \textsc{low}$ as well), or $\hat{w} \geq \frac{1}{2}w_x = \frac{1}{2} \cdot (s_x + \mu(x)) \geq \frac{1}{2} \cdot 401s_x > 108s_x$. In this case, for $a = \hat{w}/9$, we have $s_x < \frac{1}{108}w_x = \frac{1}{12}a$ and hence the estimation outputs LOW for s_x with probability 2/3.

If $\frac{1}{400}\max\{c, \mu(x)\} \leq s_x \leq \max\{c, \frac{1}{4}\mu(x)\}$, then with probability at least 8/9, either $\hat{w}$ is LOW (and then we correctly return $\hat{s} = \textsc{low}$) or the saturation-aware estimation of s_x returns, with probability at least 2/3, either LOW or an answer in the range $(1 \pm \varepsilon/6)s_x$. As seen before, the latter is in the range $(1 \pm \varepsilon)s_x$.

The analysis for $\mu(x)$ and $\hat{p}$ is analogous (and even a bit stricter, since the indicator for $\mu(x)$ is exact and does not have the $\pm\eta_{c,\varepsilon}$ additive penalty). □

Avoiding probability zero sets Note that all calls made by Reference-estimation$(\mu, c, \varepsilon; x)$ to Target-test$(\mu, c, \varepsilon; x, y)$ involve an element y that was sampled (unconditionally) from μ. Since Target-test$(\mu, c, \varepsilon; x, y)$ is based only on samples drawn from $\{x, y\}$, this means that no probability zero sets are involved. Additionally, if $\mu(x) = 0$ then $y = x$ never happens, and the conditioning on $\{x, y\}$ causes Target-test$(\mu, c, \varepsilon; x, y)$ to reject y with probability 1, forcing the output of Reference-estimation$(\mu, c, \varepsilon; x)$ to be (LOW, LOW).

7 Finding α

We prove Lemma 4.5 in this section. We recall it here.

Lemma 4.5 (Find-good-α)**.** *Assume that $\mu(x) \leq \frac{1}{4}s_x$. The output of Algorithm 12 is a random variable α for which, with probability 2/3, $\gamma_x \leq \alpha \leq 41\gamma_x$, at the cost of $O(\log\log N)$ samples at worst case.*

We look for α using a binary search, adapted to a probabilistic setting as defined below.

Definition 7.1 (Uncertain comparator)**.** Let $\mathcal{A}$ be an oracle to a probabilistic function from $\{1, \ldots, N\}$ to {"low","good","high"}. We say that $\mathcal{A}$ is an *uncertain comparator* if:

- Conviction: there exists a function $f : \{1, \ldots, N\} \to$ {"low","good","high"} such that for every $1 \leq i \leq N$ and every event E about past calls, $\Pr[\mathcal{A}(i) = f(i)|E] \geq 99/100$.
- Monotonicity: the above function f is non-decreasing monotone with respect to the full order "low" < "good" < "high".

Definition 7.2 (Goal range of an uncertain comparator)**.** Let $\mathcal{A}$ be an uncertain comparator over $\{1, \ldots, N\}$. The *goal range* of $\mathcal{A}$ is the set $\{1 \leq i \leq N : \Pr[\mathcal{A}(i) = \text{"good"}] \geq 99/100\}$ (due to monotonicity, this is always a segment).

Definition 7.3 (Appeasement of an uncertain comparator)**.** An uncertain comparator $\mathcal{A}$ is *appeasable* if its goal range is non-empty.

The interface of the binary search is stated in the following lemma:

Lemma 7.4 (Uncertain-binary-search)**.** *Assume that we have access to an appeasable uncertain comparator $\mathcal{A}$. The output of Algorithm 13 is in the goal range of $\mathcal{A}$ with probability at least $2/3$, at the cost of $O(\log N)$ oracle calls.*

We present Algorithm 13 and prove the above lemma in Subsection 7.3. Note that the 99/100 could be substituted by any fixed constant strictly greater than 1/2. We use it rather than the more standard 2/3 bound to eliminate the need for amplification in the implementation of Algorithm 13.

7.1 The uncertain-comparator for α

We provide here a procedure whose guarantees are weaker than those of an uncertain comparator, in the sense that it allows for some "gray areas" where there is more than one correct answer (and hence no probability guarantee). Later, in the proof of Lemma 5.3, we use it in a way that side-steps the issue with the gray areas.

Lemma 7.5 (Weak-uncertain-comparator)**.** *There exist $\alpha_x \in (2.3\gamma_x, 38\gamma_x)$ such that:*

- *If $\alpha \leq 1 \cdot \gamma_x$, then the output of Algorithm 11 is "low" with probability at least $2/3$.*
- *If $\frac{1}{2}\alpha_x \leq \alpha \leq \alpha_x$, then the output of Algorithm 11 is "good" with probability at least $2/3$.*
- *If $\alpha \geq 41\gamma_x$, then the output of Algorithm 11 is "high" with probability at least $2/3$.*

Moreover, the number of samples drawn by Algorithm 11 is $O(1)$.

We prove Lemma 7.5 in this subsection. We essentially show that, if $\alpha = \Theta(1) \cdot \gamma_x$, then the expectations $\mathrm{E}[\beta_{x,\alpha}]$ and $\mathrm{E}[\beta_{x,2\alpha}]$ lie inside a globally fixed range and are well-separated by an additive stride. Hence, it suffices to estimate $\mathrm{E}[\beta_{x,\alpha}]$ within half of this stride to implement the comparator for the binary search, possibly being wrong once at each side of the correct range. We formally state this sketch in one observation and two lemmas.

Observation 7.6. $\mathrm{E}[\beta_{x,\alpha}]$ *is non-decreasing monotone with respect to the choice of α.*

Proof. We can apply Observation 3.30, since the expression that defines $\beta_{x,\alpha}$, which is based on $V_{x,\alpha}$, is non-decreasing monotone. □

Lemma 7.7 (Effective bounds for $\mathrm{E}[\beta_{x,\alpha}]$)**.** *There exists $2.3\gamma_x \leq \alpha_x \leq 38\gamma_x$ for which $\mathrm{E}[\beta_{x,\alpha_x}] = 0.91$. Additionally, if $\alpha \leq 2\gamma_x$ then $\mathrm{E}[\beta_{x,\alpha}] < 0.9$ and if $\alpha \geq 41\gamma_x$ then $\mathrm{E}[\beta_{x,\alpha}] > 0.92$.*

We prove Lemma 7.7 in Appendix E.

At this point we provide Algorithm 11 based on the sketch above, which implements the weak uncertain comparator.

Proof of Lemma 7.5. Case I: if $\alpha \leq \gamma_x$, then by Lemma 7.7, $\mathrm{E}[\beta_{x,2\alpha}] \leq \mathrm{E}[\beta_{x,2\gamma_x}] < 0.9$. In this case, with probability at least 5/6, $\hat{h} \leq \mathrm{E}[\beta_{x,\alpha}] + \frac{1}{200} < 0.905$, and the algorithm outputs "low".

Algorithm 11: Procedure Weak-uncertain-comparator$(\mu, c, \varepsilon; x, \alpha)$
Inaccessible data: γ_x, α_x (of Lemma 7.7).
Output: "low" if $\alpha \le 1\gamma_x$, "good" if $\frac{1}{2}\alpha_x \le \alpha \le \alpha_x$, "high" if $\alpha \ge 41\gamma_x$.

1. Let $M \leftarrow 9$. For median amplification (Observation 3.32(a)).
2. For i from 1 to 9:
   (a) Let $\hat{\ell}_i \leftarrow$ Estimate-E$[\beta_{x,\alpha}](\mu, c, \varepsilon; x, \alpha)$.
   (b) Let $\hat{h}_i \leftarrow$ Estimate-E$[\beta_{x,\alpha}](\mu, c, \varepsilon; x, \min\{1, 2\alpha\})$.
3. Let $\hat{\ell} \leftarrow \text{median}(\hat{\ell}_1, \ldots, \hat{\ell}_m)$. SP: $\frac{5}{6}$
4. Let $\hat{h} \leftarrow \text{median}(\hat{h}_1, \ldots, \hat{h}_m)$. SP: $\frac{5}{6}$
5. If $\hat{h} < 0.905$:
   (a) Return "low". (Inferring that $h < 0.91$)
6. If $\hat{\ell} > 0.915$:
   (a) Return "high". (Inferring that $\ell > 0.91$)
7. Return "good". (Inferring that $\ell \le 0.91 \le h$)

Case II: if $\alpha \ge 41\gamma_x$, then by Lemma 7.7, $\mathrm{E}[\beta_{x,\alpha}] \ge \mathrm{E}[\beta_{x,41\gamma_x}] > 0.92$. In this case, with probability at least 5/6, $\hat{\ell} \ge \mathrm{E}[\beta_{x,\alpha}] - \frac{1}{200} > 0.915$, and the algorithm outputs "high".

Case III. if $\frac{1}{2}\alpha_x \le \alpha \le \alpha_x$, for α_x guaranteed by Lemma 7.7, then with probability at least 5/6, $\hat{\ell} \le \mathrm{E}[\beta_{x,\alpha}] + \frac{1}{200} \le \mathrm{E}[\beta_{x,\alpha_x}] + \frac{1}{200} = 0.915$. Also, with probability at least 5/6, $\hat{h} \ge \mathrm{E}[\beta_{x,2\alpha}] - \frac{1}{200} \ge \mathrm{E}[\beta_{x,\alpha_x}] - \frac{1}{200} = 0.905$. By the union bound, with probability at least $\frac{2}{3}$, the algorithm outputs "good".

The cost of this procedure is the same as the cost of two estimations of $\mathrm{E}[\beta_{x,\alpha}]$ using Lemma 5.7, which is $O(1)$. □

7.2 Proof of Lemma 4.5

At this point we provide Algorithm 12 and use it to prove Lemma 4.5.

Lemma 7.8. *If $\mu(x) \le \frac{1}{4}s_x$, then $\frac{1}{2N} \le \gamma_x \le 1$.*

Proof. Recall that $\gamma_x = \frac{\mu(x)}{s_x}$, hence $\gamma_x \le \frac{1}{4} < 1$ immediately by $\mu(x) \le \frac{1}{4}s_x$.

For the lower bound, observe that

$$s_x = \mathrm{E}[\mu(V_x)] \le \max_{L_x \subseteq V_x \subseteq \Omega \setminus (H_x \cup \{x\})} \mu(V_x) \le \max |V_x| \cdot \max_{y \in V_x} \mu(y) \le (N-1) \cdot 1.2\mu(x) \le 1.2N\mu(x)$$

Hence $\gamma_x = \frac{\mu(x)}{s_x} \ge \frac{1}{1.2N} > \frac{1}{2N}$. □

Proof of Lemma 4.5. At top level, Algorithm 12 looks for $\alpha = 2^{-i}$ in the range $\{0, \ldots, N'\}$, where $N' \ge \log_2(2N)$. By Lemma 7.8, $2^{-N'} \le \gamma_x \le 1$. By Lemma 7.7, $\alpha_x \in (2.3\gamma_x, 41\gamma_x) \subseteq (1/N, 41)$ but also $\alpha_x \le 1$, hence both α_x and $\frac{1}{2}\alpha_x$ lie between the endpoints of our search range ($2^{-N'}$ and 1).

Since the guarantees of Weak-uncertain-comparator are weaker than the constraints of the uncertain comparator that can be used in Uncertain-binary-search, we use an interleaving technique: instead of

Algorithm 12: Procedure Find-good-$\alpha(\mu, c, \varepsilon; x)$
Output: $\alpha \in (1\gamma_x, 41\gamma_x)$.

1. Let $M_1 \leftarrow 47$. For median amplification (Observation 3.32(c)).
2. Let $M_2 \leftarrow 9$. For median amplification (Observation 3.32(a)).
3. Let $N' \leftarrow 1 + \lceil \log_2 N \rceil$.
4. For r from 0 to 5:
   (a) Let $I'_r = \{1, \ldots, \lfloor \frac{1}{6}(N' - r) \rfloor + 1\}$.
   (b) Let $\mathsf{Oracle}_r(i')$ be the procedure that takes the median of M_1 independent calls to Weak-uncertain-comparator$(\mu, c, \varepsilon; x, \alpha = 2^{-(6(i'-1)+r)})$. Oracle SP: $\frac{99}{100}$
   (c) For j from 1 to M_2:
      i. Let $i'_{r,j} \leftarrow$ Uncertain-binary-search$(|I'_r|\,; \mathsf{Oracle}_r(\cdot))$.
      ii. Let $i_{r,j} \leftarrow 6(i'_{r,j} - 1) + r$. (For analysis only)
   (d) Let $i'_r \leftarrow \mathrm{median}(i'_{r,1}, \ldots, i'_{r,M_1})$. SP: $\frac{5}{6}$
   (e) Let $i_r \leftarrow 6(i'_r - 1) + r$ $(= \mathrm{median}(i_{r,1}, \ldots, i_{r,M_1}))$
   (f) If $\mathsf{Oracle}_r(i') =$ "good":
      i. Return $\alpha = 2^{-i_r}$
5. Return $2^{-N'}$. A fallback output

searching the entire range, we partition it to six parts, so that in every part the indexes are arranged in skips of width six. Thus, every two consecutive indexes in each part are far enough apart to allow us to distinguish whether we are below or above the "good" indexes. Also, as together these parts cover the entire range, at least one of the parts contains an index which our comparator explicitly marks as "good".

We apply the uncertain binary search to each part separately, adding an additional "goodness check" to the index resulting from this search. We then greedily select the first index that was both selected by the search and verified by the additional check.

Observe that, for satisfying the requirements of Uncertain-binary-search, we amplify the 2/3-success probability of Weak-uncertain-comparator (for inputs where it is guaranteed) to 99/100 using median-of-M_1 (Observation 3.32(c)).

By Lemma 7.5, the comparator satisfies the requirements of Definition 7.1 (uncertain comparator) when restricted to the union of the ranges $\alpha \le 1\gamma_x$, $\alpha \ge 41\gamma_x$ and $\frac{1}{2}\alpha_x \le \alpha \le \alpha_x$, with respect to an unknown α_x whose existence is guaranteed by Lemma 7.7. There exists some integer $0 \le r_{\mathrm{hit}} \le 5$ for which the range $(\frac{1}{2}\alpha_x, \alpha_x]$ intersects the set $2^{-(6\mathbb{N}+r_{\mathrm{hit}})}$. Let α_{hit} be the only element in this intersection and let $i_{\mathrm{hit}} = -\log_2 \alpha_{\mathrm{hit}}$. Note that the comparator is both valid (with respect to Definition 7.1) in the r_{hit}th range and appeasable (since the majority answer in i_{hit} is "good").

We define the following events:

- G: at the r_{hit}th iteration, the algorithm successfully executes step 4(f)i when $i_{r_{\mathrm{hit}}} = i_{\mathrm{hit}}$.
- B_r $(0 \le r \le 5)$: at the rth iteration, the algorithm successfully, but wrongly, executes step 4(f)i when $2^{-i_r} \notin (1\gamma_x, 41\gamma_x)$.

Clearly, if G happens and none of the B_rs do, then the value of α at the return statement is in the range $(1\gamma_x, 41\gamma_x)$.

The good event: consider the iteration in which $r = r_{\text{hit}}$. Due to the interleave gaps, all choices of $\alpha = 2^{-(6(i'-1)+r)}$ for $i' \in I_r$ are outside the range $(\gamma_x, 41\gamma_x)$ except for a single choice for which $6(i'-1)+r = i_{\text{hit}}$. Therefore, in the r_{hit}th iteration, Weak-uncertain-comparator has the required behavior guarantee for all elements, and in every individual inner-iteration we obtain $i_{r_{\text{hit}},j} = i_{\text{hit}}$ with probability at least $2/3$. The probability that $i_{r_{\text{hit}}} = i_{\text{hit}}$ is at least $5/6$ since we use the median-of-M_2 amplification (Observation 3.32(a)). With probability at least $99/100$, the additional call to the oracle returns "good", and then we return $\alpha_{\text{hit}} \in (1\gamma_x, 41\gamma_x)$. To conclude, $\Pr[G] \geq \frac{5}{6} - \frac{1}{100}$.

Bad events: consider some $0 \leq r \leq 5$ for which $2^{-i_r} \notin (1\gamma_x, 41\gamma_x)$. In this case, the oracle returns either "low" or "high" with probability at least $99/100$, hence $\Pr[B_r] \leq \frac{1}{100}$.

By the union bound,

$$\Pr\left[\alpha \in (\gamma_x, 41\gamma_x)\right] \geq \Pr\left[G \wedge \bigwedge_{r=0}^{5} \neg B_r\right] \geq \Pr[G] - \sum_{r=0}^{5} \Pr[B_i] \geq \left(\frac{5}{6} - \frac{1}{100}\right) - 6 \cdot \frac{1}{100} > \frac{2}{3} \quad \square$$

7.3 The uncertain-comparator binary search

In this subsection we prove the correctness of the uncertain-comparator binary search costing $O(\log n)$ uncertain-comparator calls (Lemma 7.4). Note that a standard amplification of the uncertain comparator allows binary search at the cost of an additional $O(\log \log n)$ factor.

Recall Lemma 7.4:

Lemma 7.4 (Uncertain-binary-search)**.** *Assume that we have access to an appeasable uncertain comparator $\mathcal{A}$. The output of Algorithm 13 is in the goal range of $\mathcal{A}$ with probability at least $2/3$, at the cost of $O(\log N)$ oracle calls.*

The binary search algorithm executes a Markov chain over a range tree in a way that can be seen as a random-walk over a *line*. To prove the correctness of our binary search variant, we recap common definitions.

Definition 7.9 (Dyadic range tree)**.** A *dyadic range tree* is a tree whose root holds the dyadic interval $\{1, \ldots, 2^k\}$, in which every non-leaf node has two children, each holding half of the node's range.

Observation 7.10. *In a dyadic range tree whose root range is $\{1, \ldots, 2^k\}$, all nodes of depth $0 \leq k' \leq k$ (where the root's depth is 0) hold dyadic ranges of length exactly $2^{k-k'}$. In particular, all leaves (which hold singletons) have the same depth, which is k.*

Proof. Trivial for $k' = 0$. By induction for $1 \leq k' \leq k$: an internal node in depth $k'-1$ holds some dyadic range $\{2^{k-k'+1}t + 1, \ldots, 2^{k-k'+1}t + 2^{k-k'+1}\}$. Its left child holds the dyadic range $\{2^{k-k'}(2t) + 1, \ldots, 2^{k-k'}(2t) + 2^{k-k'}\}$ and its right child holds the dyadic range $\{2^{k-k'}(2t+1) + 1, \ldots, 2^{k-k'}(2t+1) + 2^{k-k'}\}$. $\square$

A deterministic binary search can be represented as a walk over a dyadic range tree, where we start with the widest considerable range and in every step we proceed to a narrower range until we reach a leaf, whose singleton range represents the result of the search.

In our setting the comparator is probabilistic, hence the search is a random walk. If we fully trust our comparator and only go forward, as in the deterministic version, then the worst-case guarantee for the success probability is $(99/100)^{\log_2 n}$ which is too low. Alternatively, we can amplify the confidence of the uncertain comparator by considering the majority vote of $O(\log\log n)$ independent calls. This reduces the error probability to $o(\log n)$ in each step, and hence by the union bound, the run of the algorithm is $o(1)$-close to the deterministic version. Though correct, this approach brings an $O(\log\log n)$-penalty which we want to avoid.

Instead of these forward-only tree walks, we define a local logic that uses three uncertain comparisons to choose the best edge to use in each step. This edge is possibly the parent edge, which allows us to correct errors that may occur in this setting, as opposed to a deterministic binary search.

To formulate the random walk of Algorithm 13 as a search, we define a set of *good leaves*, corresponding to the the values for which the comparator outputs "good" with high probability. A random walk of a predefined length is considered *successful* if we reach one of the good leaves at the very last step. We insist that we only consider the last step: it does not suffice to pass through a good leaf during the walk.

To prove our formulation, we consider the edge-distance of every node from the set of good leaves (where a distance from a set is the minimum distance from a leaf in this set). Even though the sequence of distances is not memoryless, its guarantees suffice to apply the following lemma.

Lemma 7.11 (Linear random walk)**.** *Consider the following random walk with parameter k: we start with $X_1 = k$. In every step, we choose $B_i = 1$ with probability at least $1 - \frac{3}{100}$, and otherwise $B_i = 0$. We allow $\Pr[B_i = 1]$ to depend on the history (the choices of $B_1, \ldots, B_{i-1}$), but the lower bound of $1 - \frac{3}{100}$ holds for every condition on individual histories. After choosing B_i, if $B_i = 1$ then $X_{i+1} = \max\{0, X_i - 1\}$, and if $B_i = 0$, then $0 \le X_{i+1} \le X_i + 1$ (but it must be an integer). In this setting, if $n \ge 20k + 1$, then $\Pr[X_n = 0] \ge \frac{2}{3}$.*

Proof. For $1 \le i \le n$, let G_i be the event "$X_i = 0$". Also, for $1 \le i < j \le n$, let $G_{i,j}$ be the event "$\sum_{t=i}^{j-1} B_t \ge \frac{1}{2}(j-i)$".

A key observation is that $\bigvee_{i=1}^{n-1} G_i \wedge \bigwedge_{i=1}^{n-1} G_{i,n}$ implies G_n. If we assume the contrary, then there exists $1 \le i \le n-1$ for which $G_i \wedge G_{i,n}$ holds. Consider the maximal such i, and for every $i \le j \le n$ let $Y_j = X_i + \sum_{t=i}^{j-1}(-1)^{B_t}$. From the assertions on $X_1, \ldots, X_n$ and the maximality of i, $Y_j \ge X_j$ for every $i+1 \le j \le n$, a contradiction since $0 < X_n \le Y_n = X_i + \sum_{t=i}^{n-1}(-1)^{B_t} = X_i + (n-i) - 2\left|\{i \le t \le n-1 : B_j = 1\}\right| \le X_i + (n-i) - 2 \cdot \frac{1}{2}(n-i) = X_i = 0$.

It remains to show that $\Pr\left[\bigvee_{i=1}^{n-1} G_i \wedge \bigwedge_{i=1}^{n-1} G_{i,n}\right] \ge \frac{2}{3}$. Consider the negation of each part:

$$\begin{aligned}
\Pr\left[\neg \bigvee_{i=1}^{n-1} G_i\right] = \Pr\left[\bigwedge_{i=1}^{n-1} \neg G_i\right] \le \Pr\left[\neg G_{n-1}\right] &= \Pr\left[\mathrm{Bin}\left(n-1, \frac{3}{100}\right) \ge \frac{1}{2}(n-1-k)\right] \\
&\le e^{-2\left(\left(\frac{1}{2} - \frac{3}{100}\right)(n-1) - \frac{1}{2}k\right)^2/(n-1)} \\
[n \ge 20k+1] \quad &\le e^{-2\left(\left(\frac{1}{2} - \frac{3}{100} - \frac{1}{40}\right)(n-1)\right)^2/(n-1)} \\
&= e^{-0.39605(n-1)} \le e^{-0.39605 \cdot 20} < \frac{1}{1000}
\end{aligned}$$

Algorithm 13: Procedure Uncertain-binary-search$(n; \mathcal{A})$
Input: An uncertain comparator $\mathcal{A} : \{1, \ldots, n\} \to \{\text{“low”}, \text{“good”}, \text{“high”}\}$.
Promise: $\mathcal{A}$ is appeasable.
Output: i for which $\text{ans}(i) = \text{“good”}$.
WLOG: n is a power of 2.
(The comparator deterministically returns “high” $n + 1 \leq i \leq 2^{\lceil \log_2 n \rceil}$)

1. Initialize $stk \leftarrow \emptyset$. (Empty backtrace stack)
2. Initialize $L_1 = 1$, $R_1 = n$. (Current node is the root)
3. For i from 2 to $20 \log_2 n + 1$: (An integer by assumption)
   (a) If $L_{i-1} = R_{i-1}$: (Currently in a leaf)
      i. Let $ans \leftarrow \mathcal{A}(L_{i-1})$.
      ii. If $ans = \text{“good”}$:
         A. Let $(L_i, R_i) \leftarrow (L_{i-1}, R_{i-1})$ (Stay at leaf)
      iii. Else:
         A. Let $(L_i, R_i) \leftarrow \text{pop}(stk)$ (Move to parent)
   (b) Else: (Currently in an inner node)
      i. Let $M_{i-1} \leftarrow \frac{1}{2}(L_{i-1} + R_{i-1} - 1)$. (Always an integer)
      ii. Let $ans_L \leftarrow \mathcal{A}(L_{i-1})$.
      iii. Let $ans_M \leftarrow \mathcal{A}(M_{i-1})$.
      iv. Let $ans_R \leftarrow \mathcal{A}(R_{i-1})$.
      v. If $ans_L \leq ans_M \leq ans_R$ and $ans_L \leq \text{“good”} \leq ans_R$: (Consistent answers)
         A. $\text{push}(stk, (L_{i-1}, R_{i-1}))$ (Record current node)
         B. If $ans_M = \text{“high”}$: (Middle is too high)
            • Let $(L_i, R_i) \leftarrow (M_{i-1} + 1, R_{i-1})$. (Move to right child)
         C. Else:
            • Let $(L_i, R_i) \leftarrow (L_{i-1}, M_{i-1})$. (Move to left child)
      vi. Else: (Inconsistent answers)
         A. If $(L_{i-1}, R_{i-1}) = (1, n)$:
            • Let $(L_i, R_i) \leftarrow (L_{i-1}, R_{i-1})$. (Stay at root)
         B. Else:
            • Let $(L_i, R_i) \leftarrow \text{pop}(stk)$. (Move to parent)
4. Return $L_{20 \log_2 n + 1}$.

For $1 \leq i \leq n - 9$, we can use Chernoff bound to obtain that:

$$\Pr[\neg G_{i,n}] \leq \Pr\left[\text{Bin}\left(n - i, \frac{3}{100}\right) \geq \frac{1}{2}(n - i)\right] \leq e^{-2\left(\frac{1}{2} - \frac{3}{100}\right)^2 (n-i)} = e^{-0.4418(n-i)}$$

By the union bound,

$$\Pr\left[\bigvee_{i=1}^{n-9} \neg G_{i,n}\right] \leq \sum_{i=1}^{n-9} \Pr\left[\neg G_{i,n}\right] \leq \sum_{i=1}^{n-9} e^{-0.4418(n-i)} \leq \sum_{i=9}^{\infty} e^{-0.4418i} = \frac{e^{-0.4418 \cdot 9}}{1 - e^{-0.4418}} < \frac{1}{19}$$

For $n-8 \le i \le n-1$ we use a collective bound:

$$\Pr\left[\bigvee_{i=n-8}^{n-1} \neg G_{i,n}\right] \le \Pr\left[\bigvee_{i=n-8}^{n-1} (B_i \neq 1)\right] \le \sum_{i=n-8}^{n-1} \Pr\left[B_i \neq 1\right] \le 8 \cdot \frac{3}{100} = \frac{6}{25}$$

Combined,

$$\Pr\left[\neg \bigwedge_{i=1}^{n-1} G_{i,n}\right] = \Pr\left[\bigvee_{i=1}^{n-1} \neg G_{i,n}\right] \le \Pr\left[\bigvee_{i=1}^{n-9} \neg G_{i,n}\right] + \Pr\left[\bigvee_{i=n-8}^{n-1} \neg G_{i,n}\right] \le \frac{1}{19} + \frac{6}{25} < \frac{3}{10}$$

If $n \ge 20k+1$ then:

$$\Pr[\neg G_n] \le \Pr\left[\neg\left(\bigvee_{i=1}^{n-1} G_i \wedge \bigwedge_{i=1}^{n-1} G_{i,n}\right)\right] \le \frac{1}{1000} + \frac{3}{10} < \frac{1}{3} \qquad \square$$

At this point we prove Lemma 7.4 (correctness of Algorithm 13 as a binary search).

Proof of Lemma 7.4. Let a and b be the endpoints of the goal range of $\mathcal{A}$ (which is promised to be a non-empty segment). Without loss of generality we can assume that n is a power of 2. Otherwise, we can extend the comparator to answer "high" with probability 1 for every $n+1 \le i \le 2^{\lceil \log_2 n \rceil}$. We use a dyadic range tree with root range $\{1, \ldots, n\}$. Recall that all leaves have the same depth, $\log_2 n$, and let L_{good} be the set of leaves inside the goal range (those for which the comparator answers "good" with high probability).

Algorithm 13 defines a random walk on the dyadic range tree as follows: on a leaf $\{i\}$, we call the comparator to test whether i is good or not. If it answers "good" then we stay on it, and otherwise we move to its parent. On an internal node $\{L, \ldots, R\}$, let $M = (L+R-1)/2$. We use the comparator to see whether the answers about L, M, R make sense with respect to the predicates "the assessments related to L, M and R indeed form a monotone sequence, and also imply that the range $[L, R]$ is not disjoint from the goal range". If the answers make sense, then we move to one of the children based on the answer on M (left child if M is "low" or "good", right child if it is "high"). Otherwise we move to its parent, unless we are already at the root, in which case we stay in place.

Let D_i be the edge-distance between the current node (which is not necessarily an ancestor of a "good" leaf) and the closest good leaf. Observe that at any point:

- If we are in the root or in a "bad" leaf $\{i\}$ (outside the goal range), then with probability at least 97/100 we take the edge which moves us closer to the set of good leaves.
- If we are in an inner node that has a descendant "good" leaf, then with probability 97/100 we take the edge to a child on the path to one of these leaves.
- If we are in an inner node that has only "bad" leaves in its subtree, then with probability 97/100 we take the parent edge, moving closer to the set of good leaves.
- If we are in an "good" leaf, then we stay there with probability $99/100 > 97/100$.

These observations describe a memoryless random-walk on the dyadic tree. If we only consider the sequence $D_1, D_2, \ldots$ (which is not necessarily memoryless), then we satisfy the assertions of Lemma 7.11. Hence, with probability at least 2/3, the $(20 \log_2 n + 1)$st step of the algorithm is a good leaf, as desired. □

8 Estimating $\mu(x)$ using α

In this section we prove Lemma 4.6. For this, we define a function h and show that $\mathrm{E}[h(\beta_{x,\alpha})]$ is a good approximation for $\frac{\alpha\mu(x)}{s_x}$. To estimate $\mathrm{E}[h(\beta_{x,\alpha})]$, we have to draw individual values of $\beta_{x,\alpha}$ and estimate them.

Recall that $\beta_{x,\alpha} = \Pr_\mu[\neg x | V_{x,\alpha} \cup \{x\}]$. In particular, it is fully determined by the choice of A_α and V_x. Since $\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}} = \frac{\mu(V_{x,\alpha})}{\mu(x)}$, we obtain that $\mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right] = \frac{\alpha s_x}{\mu(x)}$. Alternatively, we can use $\mu(x) = \alpha \cdot s_x / \mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right]$, where α is known and s_x is already estimated within a $(1 \pm \varepsilon/3)$-factor. To estimate $\mu(x)$ within a $(1 \pm \varepsilon)$-factor as desired, it suffices to estimate $\mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right]$ within a $(1 \pm \varepsilon/2)$-factor, since $(1 \pm \varepsilon/3)(1 \pm \varepsilon/2) = 1 \pm \varepsilon$.

Since $\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}$ is only bounded by $\frac{1}{\mu(x)}$, which is too large to effectively approximate, we truncate it at $T = 8 \ln \varepsilon^{-1} + 100$ using the function $h(\beta) = \min\left\{T, \frac{\beta}{1-\beta}\right\}$, and use a separate argument to bound the difference that this truncation introduces to the expectation.

At this point we introduce Algorithm 14 and prove its correctness, thereby proving Lemma 4.6.

Algorithm 14: Procedure Estimate-scaled-result$(\mu, c, \varepsilon; x, \alpha)$
Output: $\hat{b} \in (1 \pm \varepsilon/2)\alpha s_x/\mu(x)$.

1. Let $T \leftarrow 8 \ln \varepsilon^{-1} + 100$.
2. Let $M_1 \leftarrow \lceil 9600/\varepsilon^2 \rceil$.
3. Let $M_2 \leftarrow \lceil 30 \ln M_1 \rceil$. For median amplification (Observation 3.32(e)).
4. Let $\delta \leftarrow \frac{\varepsilon}{168 \ln \varepsilon^{-1} + 2163}$.
5. Let $q \leftarrow \left\lfloor 25 M_2 \cdot \frac{\ln(6/\delta)}{\delta^3} \right\rfloor$.
6. For i from 1 to M_1 :
   (a) Draw $A_\alpha^{(i)}$, according to its definition.
   (b) $V_x^{(i)} \leftarrow$ Initialize-new-$V_x(c, \varepsilon; x, q)$.
   (c) For j from 1 to M_2:
      i. Let $\hat{\beta}_{i,j} \leftarrow$ Estimate-$\beta_{x,\alpha}(\mu, c, \varepsilon; x, \alpha, \delta, A_\alpha^{(i)}, V_x^{(i)})$.
   (d) Let $\hat{\beta}_i \leftarrow \mathrm{median}(\hat{\beta}_{i,1}, \ldots, \hat{\beta}_{i,M_2})$. SP: $1 - \frac{1}{24M_1}$
   (e) Define (without computing) $\beta_i \leftarrow \beta(A_\alpha^{(i)}, V_x^{(i)})$. (For analysis only)
   (f) Let $\hat{b}_i \leftarrow \min\left\{\frac{\hat{\beta}_i}{1-\hat{\beta}_i}, T\right\}$. $= h(\hat{\beta}_i)$
7. Let $\hat{b} \leftarrow \frac{1}{M} \sum_{i=1}^{M} \hat{b}_i$.
8. Return $\hat{b}$.

Before diving into the algorithmic logic we state a few arithmetic lemmas.

Observation 8.1. $\eta_{c,\varepsilon} M_1 q < \frac{1}{12}$.

Proof. We use the following bounds for $\varepsilon < \frac{1}{10} < e^{-2}$:

$$\begin{aligned}
M_1 = &\quad \left\lceil \tfrac{9600}{\varepsilon^2} \right\rceil && \leq \frac{9601}{\varepsilon^2} \\
M_2 = &\quad \lceil 30 \ln M_1 \rceil \leq 30 \ln \tfrac{9601}{\varepsilon^2} + 1 = 30 \ln \tfrac{9601 e^{1/30}}{\varepsilon^2} && \leq 200 \ln \varepsilon^{-1} \\
\delta = &\quad \tfrac{\varepsilon}{168 \ln \varepsilon^{-1} + 2163} && \leq \frac{\varepsilon}{2331 \ln \varepsilon^{-1}} \\
\ln(6/\delta) \leq &\quad \ln 6 + \ln 2331 + \ln \ln \varepsilon^{-1} + \ln \varepsilon^{-1} && \leq 12 \ln \varepsilon^{-1}
\end{aligned}$$

Therefore:

$$\begin{aligned}
\eta_{c,\varepsilon} M_1 q &\leq \eta_{c,\varepsilon} \cdot M_1 \cdot 25 M_2 \frac{\ln(6/\delta)}{\delta^3} \\
&= \frac{1}{12} \cdot \eta_{c,\varepsilon} \cdot 300 M_1 M_2 \frac{\ln(6/\delta)}{\delta^3} \\
&\leq \frac{1}{12} \cdot \eta_{c,\varepsilon} \cdot 300 \frac{9601}{\varepsilon^2} \cdot 200 \ln \varepsilon^{-1} \cdot \frac{12 \ln \varepsilon^{-1}}{\varepsilon^3 / (2331 \ln \varepsilon^{-1})^3} \\
&\leq \frac{1}{12} \cdot \frac{\varepsilon^5}{10^{20} (\ln \varepsilon^{-1})^5} \cdot 9 \cdot 10^{19} \frac{(\ln \varepsilon^{-1})^5}{\varepsilon^5} < \frac{1}{12}
\end{aligned}$$

□

Lemma 8.2. *If $\gamma_x \leq \alpha \leq 50\gamma_x$ then $\mathrm{E}[h(\beta_{x,\alpha})] \in \left(1 \pm \frac{1}{10}\varepsilon\right) \alpha s_x / \mu(x)$. In particular, $\mathrm{E}[h(\beta_{x,\alpha})] \geq \frac{9}{10}$ for $\varepsilon < 1$.*

We prove Lemma 8.2 in Appendix E.

Lemma 8.3. *If $\gamma_x \leq \alpha \leq 50\gamma_x$ then $\mathrm{Var}\left[h(\beta_{x,\alpha})\right] \leq 100$.*

Proof. For every $y \in \Omega$, let $\mathbf{1}_{y \in V_{x,\alpha}}$ be an indicator for the event "$y \in V_{x,\alpha}$". Note that:

$$\begin{aligned}
\mathrm{Var}\left[\mu(V_{x,\alpha})\right] &= \sum_{y \in L_x \cup M_x} (\mu(y))^2 \, \mathrm{Var}\left[\mathbf{1}_{y \in V_{x,\alpha}}\right] \\
&= \sum_{y \in L_x \cup M_x} (\mu(y))^2 \Pr[y \in V_{x,\alpha}](1 - \Pr[y \in V_{x,\alpha}]) \\
&\leq \left(\max_{y \in L_x \cup M_x} \mu(y) \right) \cdot \sum_{y \in L_x \cup M_x} \mu(y) \Pr[y \in V_{x,\alpha}] \leq 1.2\mu(x) \cdot \mathrm{E}[\mu(V_{x,\alpha})]
\end{aligned}$$

Since $\frac{\beta_{x,\alpha}}{1-\beta_{x,a}} = \frac{\mu(V_{x,\alpha})}{\mu(x)}$, we can now bound its variance.

$$\mathrm{Var}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right] = \mathrm{Var}\left[\frac{\mu(V_{x,\alpha})}{\mu(x)}\right] \leq \frac{1.2\mu(x)\,\mathrm{E}[\mu(V_{x,\alpha})]}{(\mu(x))^2} = \frac{1.2\,\mathrm{E}[\mu(V_{x,\alpha})]}{\mu(x)} = \frac{1.2\alpha s_x}{\mu(x)} = 1.2\alpha/\gamma_x \leq 100$$

Observe that $\mathrm{Var}[h(\beta_{x,\alpha})] \leq \mathrm{Var}[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}]$, since h is 1-Lipschitz with respect to $\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}$. Hence, $\mathrm{Var}[h(\beta_{x,\alpha})] \leq 100$. □

Lemma 8.4. *For $0 < \delta \leq \frac{\varepsilon}{21(T+3)}$ and $\hat{\beta} = \beta \pm \delta$, $h(\hat{\beta}) = h(\beta) \pm \max\{2\delta, \frac{1}{20}\varepsilon h(\beta)\}$.*

We prove Lemma 8.4 in Appendix E.

Lemma 8.5 (Generic bound)**.** *Let $r, r_1, r_2 > 0$. Let $X_1, \ldots, X_k$ be independent non-negative variables drawn from the same distribution, $\bar{X} = \frac{1}{k}\sum_{i=1}^{k} X_i$ be their average value, and $Y_1, \ldots, Y_k$ be another random sequence for which $|Y_i| \leq \max\{r_1 \operatorname{E}[\bar{X}], r_2 X_i\}$ for every $1 \leq i \leq k$. If $r > r_1 + r_2$ then $\Pr\left[\frac{1}{k}\sum_{i=1}^{k}(X_i + Y_i) \neq (1 \pm r)\operatorname{E}[X]\right] \leq \Pr\left[\bar{X} \neq (1 \pm r')\operatorname{E}[\bar{X}]\right]$ for $r' = \frac{r - r_1 - r_2}{1 + r_2}$.*

Proof. In the following we show that if $\bar{X} \in (1 \pm r')\operatorname{E}[\bar{X}]$ then $\frac{1}{k}\sum(X_i + Y_i) \in (1 \pm r)\operatorname{E}[\bar{X}]$. This implies that the event "$\bar{X} \notin (1 \pm r')\operatorname{E}[\bar{X}]$" contains the event "$\frac{1}{k}\sum(X_i + Y_i) \notin (1 \pm r)\operatorname{E}[\bar{X}]$", which implies that $\Pr\left[\frac{1}{k}\sum(X_i + Y_i) \notin (1 \pm r)\operatorname{E}[\bar{X}]\right] \leq \Pr\left[\bar{X} \notin (1 \pm r')\operatorname{E}[\bar{X}]\right]$.

Let $\bar{Y} = \frac{1}{k}\sum_{i=1}^{k} Y_i$. By the triangle inequality and the assumptions of the lemma,

$$|\bar{Y}| \leq \frac{1}{k}\sum_{i=1}^{k}|Y_i| \leq \frac{1}{k}\sum_{i=1}^{k}\max\{r_1 \operatorname{E}[\bar{X}], r_2 X_i\} \leq \frac{1}{k}\sum_{i=1}^{k}(r_1 \operatorname{E}[\bar{X}] + r_2 X_i) = r_1 \operatorname{E}[\bar{X}] + r_2 \bar{X}$$

Assume that $\bar{X} \in (1 \pm r')\operatorname{E}[\bar{X}]$. Combined with the previous bound we can obtain that:

$$\begin{aligned}
\frac{1}{k}\sum_{i=1}^{k}(X_i + Y_i) = \bar{X} + \bar{Y} &\in (1 \pm r')\operatorname{E}[\bar{X}] \pm r_1 \operatorname{E}[\bar{X}] \pm r_2 \bar{X} \\
&\subseteq (1 \pm r')\operatorname{E}[\bar{X}] \pm r_1 \operatorname{E}[\bar{X}] \pm (1 \pm r')r_2 \operatorname{E}[\bar{X}] \\
&= (1 \pm r' \pm r_1 \pm (1 \pm r')r_2)\operatorname{E}[\bar{X}] \\
&= (1 \pm (r' + r_1 + (1 + r')r_2))\operatorname{E}[\bar{X}] \\
&= (1 \pm ((1 + r_2)r' + r_1 + r_2))\operatorname{E}[\bar{X}] \\
&= (1 \pm r)\operatorname{E}[\bar{X}]
\end{aligned}$$

□

Lemma 8.6. *Let $X_1, \ldots, X_{M_1}$ be a sequence of M_1 independent variables whose distribution is the same as $h(\beta_{x,\alpha})$. In this setting, $\Pr\left[\frac{1}{M_1}\sum_{i=1}^{M_1} X_i \neq \left(1 \pm \frac{1}{4}\varepsilon\right)\operatorname{E}[h(\beta_{x,\alpha})]\right] \leq \frac{50}{243}$.*

Proof. By Chebyshev's bound,

$$\begin{aligned}
\Pr\left[\frac{1}{M_1}\sum_{i=1}^{M_1} X_i \neq \left(1 \pm \frac{1}{4}\varepsilon\right)\operatorname{E}[h(\beta_{x,\alpha})]\right] &\leq \frac{\frac{1}{M_1}\operatorname{Var}[h(\beta_{x,\alpha})]}{\left(\frac{1}{4}\varepsilon \operatorname{E}[h(\beta_{x,\alpha})]\right)^2} \\
\text{[Lemma 8.3]} &\leq \frac{100/M_1}{\frac{1}{16}\varepsilon^2 \left(\operatorname{E}[h(\beta_{x,\alpha})]\right)^2} \\
\text{[Lemma 8.2]} &\leq \frac{1600}{M_1 \varepsilon^2 0.9^2} \leq \frac{1600}{(9600/\varepsilon^2)\varepsilon^2 \cdot 0.81} = \frac{1600}{9600 \cdot 0.81} = \frac{50}{243}
\end{aligned}$$

□

Lemma 8.7. *The sequence $(V_x^{(1)}, \ldots, V_x^{(M_1)})$ drawn by the algorithm is $\frac{1}{12}$-close to a sequence of M_1 independent drawings of V_x.*

Proof. Clearly, the $V_x^{(i)}$s are fully independent. For every $1 \le i \le M_1$, the distribution of $V_x^{(i)}$ is $\eta_{c,\varepsilon} q$-close to the correct distribution of V_x. Hence, by a union bound, the distance between the two sequences is bounded by $\eta_{c,\varepsilon} q M_1$. By Observation 8.1, the latter expression is bounded by $\frac{1}{12}$. □

Recall Lemma 4.6 about the correctness of Algorithm 14.

Lemma 4.6 (Estimate-scaled-result)**.** *Let $0 < \alpha \le 1$ be an explicitly given input, and assume that $\gamma_x \le \alpha \le 50\gamma_x$. The output of Algorithm 14 is a random variable whose value, with probability $2/3$, is $(1 \pm \varepsilon/2)\alpha s_x/\mu(x)$, at the expected cost of $O\left(\log \frac{1}{\varepsilon c} \cdot \frac{\log^5 \varepsilon^{-1}}{\varepsilon^4 (w_x + \varepsilon/\log \varepsilon^{-1})}\right)$ samples.*

Proof of Lemma 4.6. During Step 6, for every i, in the ith iteration we draw $A_\alpha^{(i)}$ and $V_x^{(i)}$. These correspond to some $\beta_i = \beta_{x,\alpha}(A_\alpha^{(i)}, V_x^{(i)})$, which are not accessible to the algorithm, but are estimated by $\hat{\beta}_i$. By Lemma 8.7, the sequence $(V_x^{(1)}, \ldots, V_x^{(M_1)})$ is $\frac{1}{12}$-close to a sequence $(U_x^{(1)}, \ldots, U_x^{(M_1)})$ of M_1 independent samples of V_x drawn according to its correct distribution.

For the analysis, we consider an optimal coupling of $(V_x^{(1)}, \ldots, V_x^{(M_1)})$ with the above hypothetical sequence $(U_x^{(1)}, \ldots, U_x^{(M_1)})$. The good event G_{eqv}, whose probability is at least $\frac{11}{12}$, is defined as the event that $U^{(i)} = V^{(i)}$ for all $1 \le i \le M_1$. This leads to a coupling of Algorithm 14 with a logically-equivalent algorithm that uses the $U_x^{(i)}$ sets instead of the $V_x^{(i)}$ sets, where the behaviors of Algorithm 14 and the hypothetical algorithm are identical when conditioned on G_{eqv}.

Recall that $\hat{\beta}_i$ is the median of $\lceil 30 \ln M_1 \rceil$ estimations of $\beta_i \pm \delta$, each of which is successful with probability at least $2/3$ (under the assumption that G_{eqv} happened). Since $M_1 > 150$, $\hat{\beta}_i = \beta_i \pm \delta$ with probability at least $1 - \frac{1}{24 M_1}$ (Observation 3.32(e)). Let G_{est} be the good event $\bigwedge_{i=1}^{M_1} (\hat{\beta}_i = \beta_i \pm \delta)$. Then by the union bound $\Pr[G_{\text{est}} | G_{\text{eqv}}] \ge \frac{23}{24}$.

Assume that $G_{\text{eqv}} \cap G_{\text{est}}$ happens. Let $h(\hat{\beta}_i) = X_i + Y_i$, where $X_i = h(\beta_i)$ and Y_i is the additive error, which according to Lemma 8.4, is bounded by $2\delta + \frac{1}{20}\varepsilon h(\beta_i) = 2\delta + \frac{1}{20}\varepsilon X_i$. Combined,

$$\begin{aligned}
\Pr\left[\hat{b} \ne \left(1 \pm \frac{1}{3}\varepsilon\right) \mathrm{E}\left[h(\beta_{x,\alpha})\right]\right] &\le \Pr[\neg G_{\text{eqv}}] + \Pr[\neg G_{\text{est}} | G_{\text{eqv}}] + \\
&\quad + \Pr\left[\hat{b} \ne \left(1 \pm \frac{1}{3}\varepsilon\right) \mathrm{E}\left[h(\beta_{x,\alpha})\right] \middle| G_{\text{eqv}} \wedge G_{\text{est}}\right] \\
(*) &\le \frac{1}{12} + \frac{1}{24} + \Pr\left[\frac{1}{M_1}\sum_{i=1}^{M_1}(X_i + Y_i) \ne \left(1 \pm \frac{1}{3}\varepsilon\right) \mathrm{E}\left[h(\beta_{x,\alpha})\right] \middle| G_{\text{est}}\right] \\
(**) &\le \frac{1}{8} + \Pr\left[\frac{1}{M_1}\sum_{i=1}^{M_1} X_i \ne \left(1 \pm \frac{1}{4}\varepsilon\right) \mathrm{E}[h(\beta_{x,\alpha})] \middle| G_{\text{est}}\right] \\
[\text{Lemma 8.6}] &\le \frac{1}{8} + \frac{50}{243} \le \frac{1}{3}
\end{aligned}$$

$(*)$: since X_i and Y_i are independent of G_{eqv}.
$(**)$: we use Lemma 8.5 with the parameters $r = \frac{1}{3}\varepsilon$, $r_1 = 2\delta \leq \frac{1}{300}\varepsilon$, $r_2 = \frac{1}{20}\varepsilon$. This results with $r' \geq \frac{1}{4}\varepsilon$. An explicit bound: $r' = \frac{r-r_1-r_2}{1+r_1} \geq \frac{\varepsilon/3-\varepsilon/300-\varepsilon/20}{1+\varepsilon/300} = \frac{(7/25)\varepsilon}{1+\varepsilon/300} \geq \frac{(7/25)\varepsilon}{1+1/300} > \frac{1}{4}\varepsilon$.

Overall, with probability at least 2/3,

$$\hat{b} = (1 \pm \varepsilon/3)\,\mathrm{E}[h(\beta_{x,\alpha})] \underset{\text{Lemma 8.2}}{=} (1 \pm \varepsilon/3)(1 \pm \varepsilon/10)\alpha s_x/\mu(x) = (1 \pm \varepsilon/2)\alpha s_x/\mu(x) \qquad \square$$

9 Applications

In this section we efficiently solve three tasks in the fully conditional model. In each of the tasks, we first construct an algorithm (or adapt an existing one) for an interim model in which one can obtain samples from the distribution along with informational queries about the distribution function itself, where the latter are received with some restrictions on availability and accuracy. Then we plug in our core estimator to provide these queries using conditional samples to complete each task. The lemmas providing this mechanism are Lemmas 9.10, 9.23 and 9.24. Appendix D holds another another such lemma which might be useful in the future.

9.1 Additional notations

The following definitions relate to the restrictions that are imposed on our interim querying model.

Definition 9.1 (ε-approximation function)**.** Let μ be a distribution over Ω. A function $f : \Omega \to [0,1]$ is an *ε-approximation function* with respect to μ if $f(x) \in (1 \pm \varepsilon)\mu(x)$ for every $x \in \Omega$.

Definition 9.2 (CDF c-truncation function)**.** Let μ be a distribution over Ω. A function $f : \Omega \to [0,1]$ is a *CDF c-truncation function* with respect to μ if:

- For every $x \in \Omega$ for which $\mathrm{CDF}_\mu(x) \geq c$, $f(x) = \mu(x)$.
- For every $x \in \Omega$ for which $\mathrm{CDF}_\mu(x) < c$, $f(x) \in \{0, \mu(x)\}$.

Definition 9.3 ((c,ε)-approximation function)**.** Let μ be a distribution over Ω. A function $f : \Omega \to [0,1]$ is a *(c,ε)-approximation function* with respect to μ if:

- $f(x) \in (1 \pm \varepsilon)\mu(x)$ for every $x \in \Omega$ for which $\mathrm{CDF}_\mu(x) \geq c$.
- $f(x) \in (1 \pm \varepsilon)\mu(x) \cup \{0\}$ for every $x \in \Omega$ for which $\mathrm{CDF}_\mu(x) < c$.

Observation 9.4. *Let μ be a distribution over Ω. A (c,ε)-approximation function h can be seen as a $(1 \pm \varepsilon)$-multiplicative approximation of a c-truncated function f (that is, $h(x) \in (1 \pm \varepsilon)f(x)$ for every $x \in \Omega$).*

The following oracle definition is a restricted variation of the "explicit sampler" model of [CFGM16].

Definition 9.5 (r-error (c,ε)-explicit sampling oracle)**.** Let μ be an input distribution over a set Ω. The *r-error (c,ε)-explicit sampling oracle* for μ has no additional input, and outputs a pair (x,p), where $x \in \Omega$ distributes like μ and with probability at least $1 - r$:

- If $\mathrm{CDF}_\mu(x) \geq c$, then p is in the range $(1 \pm \varepsilon)\mu(x)$.
- If $\mathrm{CDF}_\mu(x) < c$, then p is in the range $(1 \pm \varepsilon)\mu(x) \cup \{0\}$.

The probability of error in the estimation of $\mu(x)$, as well as the distribution over the estimated value, is independent of these probabilities for other elements. The oracle guarantees *consistency*, which means that if some element y is drawn more than once, then all pairs of the form $(y, \cdot)$ have the same second entry.

Observation 9.6. *An r-error (c, ε)-explicit sampling oracle can be seen as the following ensemble:*

- *A (c, ε)-approximation function $g_{\text{truth}} : \Omega \to [0, 1]$.*
- *An arbitrary error function $g_{\text{err}} : \Omega \to [0, 1]$.*
- *A random correctness vector $u \in \{0, 1\}^{\Omega}$ whose entries are drawn independently, and for every $x \in \Omega$, $\Pr[u_x = 1] \geq 1 - r$.*
- *The estimation outcome of the drawn $x \sim \mu$ is $h(x) = g_{\text{truth}}(x)$ if $u_x = 1$ and $h(x) = g_{\text{err}}(x)$ if $u_x = 0$.*

The function g_{truth} and g_{err} can be drawn from an arbitrary distribution over such functions.

The following oracle definition is a restricted variation of the "sample and query" model, first defined in [RS09] as the evaluation oracle.

Definition 9.7 ((c, ε)-peek oracle)**.** Let μ be an input distribution over a finite set Ω. The *(c, ε)-peek oracle* for μ gets an element $x \in \Omega$ and returns:

- An arbitrary real number in the range $(1 \pm \varepsilon)\mu(x)$, if $\text{CDF}_\mu(x) \geq c$.
- An arbitrary real number in the range $\{0\} \cup (1 \pm \varepsilon)\mu(x)$, if $\text{CDF}_\mu(x) < c$.

This definition is stricter than the definition commonly used in other works, in that the set of $x \in \Omega$ for which "0" is an allowable answer is fully determined by μ itself, rather than depending on artifacts (and at times probabilistic events) of the algorithm that simulates it.

The oracle guarantees *consistency*, which means that if the algorithm makes more than one query to an element x then it receives the same answer to all of them.

Observation 9.8. *The (c, ε)-peek oracle for a distribution μ can be seen as the querying of an arbitrarily (and possibly probabilistically) predefined (c, ε)-approximation function (Definition 9.3).*

Observation 9.9 (Amplification of testing)**.** *Assume that we have a decision test whose answer is correct with probability at least $5/8$. Then the majority answer of 3 indpendent trials is correct with probability at least $2/3$ and the majority answer of 45 independent trials is correct with probability at least $3/4$.*

We provide a proof for Observation 9.9 in Appendix F.

9.2 Learning of histograms

We first prove a generic lemma about a reduction from the (c, ε)-explicit sampling model to the fully conditional model.

Lemma 9.10. *Consider an algorithm $\mathcal{A}$ whose input is a distribution μ over Ω of size N and its output is an element of a set R, and whose access to μ consists of making at most q calls to the r-error (c, ε)-explicit sampling oracle.*

Assume that for every input μ there exists a set $R_\mu \subseteq R$ for which $\Pr[\mathcal{A}(\mu) \in R_\mu] > \frac{2}{3}$, where the probability is over a draw of the outcome sequence resulting from the algorithm's calls to a valid r-error (c,ε)-explicit sampling oracle (along with the algorithm's internal randomness).

In this setting, there exists an algorithm $\mathcal{A}'$ in the fully conditional model whose sample complexity is $O(q \cdot (\log\log N + 1/\varepsilon^4) \cdot \mathrm{poly}(\log r^{-1}, \log c^{-1}, \log \varepsilon^{-1}))$, such that for every μ, $\Pr[\mathcal{A}'(\mu) \in R_\mu] > \frac{2}{3}$.

Proof. Without loss of generality, we assume that the algorithm draws exactly q samples $x_1, \ldots, x_q \sim \mu$ and receives $p_1, \ldots, p_q$ such that in expectation the fraction of errors is at most r.

We run $\mathcal{A}$ and simulate the outcome estimation of the explicit sampling oracle while keeping "history records". In the ith call to the explicit sampling oracle we:

- Draw $x_i \sim \mu$, independent of past calls.
- Check whether $x_i = x_j$ for some $j \leq i-1$. If such a j exists, then we re-use the estimation for $\mu(x_j)$.
- If $x_i \notin \{x_1, \ldots, x_{i-1}\}$, then we use the median of $\lceil 30 \ln r^{-1} \rceil$ independent calls to Estimate-element (Theorem 4.1) with parameters (c, ε).

Clearly, the sequence $x_1, \ldots, x_q$ is independently and identically distributed like μ. By Observation 3.32(d), the probability to wrongly estimate an individual $\mu(x_i)$ (that is eligible for estimation) is bounded by $\frac{1}{2}r$, independently for every distinct x_i. Hence, we fully simulate the r-error oracle without any additional error.

By Corollary 4.3, the expected complexity of a single estimation of $x_i \sim \mu$ is $O(\log\log N) + \mathrm{poly}(\log \varepsilon^{-1}) \cdot O\left(\frac{\log^2 c^{-1}}{\varepsilon^2} + \frac{\log c^{-1}}{\varepsilon^4}\right)$. We repeat this $\lceil 30 \ln r^{-1} \rceil$ times for amplification for every $1 \leq i \leq q$. Overall, the expected sample complexity is bounded by

$$O(q \cdot (\log\log N + 1/\varepsilon^4) \cdot \mathrm{poly}(\log r^{-1}, \log c^{-1}, \log \varepsilon^{-1})) \qquad \square$$

Most of this subsection is dedicated to an algorithm for ε-learning the histogram of μ at the cost of $O(\log(N/\varepsilon)/\varepsilon^3)$ explicit samples. It works using the bucketing technique of [BFF+01]: instead of considering the distribution itself, we consider a "lower resolution picture" that results from categorizing the possible values of $\mu(x)$ by powers of $(1 - O(\varepsilon))$ that are close to them.

We start by stating a folklore lemma for learning a distribution over a small domain.

Lemma 9.11 (Folklore)**.** *Let μ be a distribution over $\{1, \ldots, n\}$ for $n \geq 16$. Assume that we construct a distribution μ' over $\{1, \ldots, n\}$ as follows: we draw q independent samples from μ, for every $1 \leq i \leq n$ let X_i be the random variable counting the number of occurrences of i in our samples, and let $\mu'(i) = X_i/q$. If $q \geq n/\varepsilon^2$, then with probability at least $8/9$, $d_{\mathrm{TV}}(\mu, \mu') \leq \varepsilon$ and $\mu'(i) \leq 2\max\{\varepsilon^2, \mu(i)\}$ for every $1 \leq i \leq n$.*

Proof. Let $E \subseteq \{1, \ldots, n\}$ be an arbitrary event. By the construction, $\mu'(E) = \frac{1}{q}\sum_{i \in E} X_i$. By Chernoff bound,

$$\Pr\left[\left|\mu'(E) - \mu(E)\right| > \varepsilon\right] = \Pr\left[\mathrm{Bin}(q, \mu(E)) \neq \mu(E)q \pm \varepsilon q\right] \leq 2e^{-2\varepsilon^2 q} \leq 2e^{-2n} < 2^{-n-5}$$

By the union bound, the probability to deviate even once is bounded by $2^n \cdot 2^{-n-5} \leq \frac{1}{32}$. Hence, with probability at least 31/32, $d_{\mathrm{TV}}(\mu, \mu') = \sup_E |\mu(E) - \mu'(E)| \leq \varepsilon$.

Additionally, consider $1 \leq i \leq n$ and let $p_i = \max\{\varepsilon^2, \mu(i)\}$. By Chernoff bound,

$$\Pr\left[\mu'(i) > 2p_i\right] \leq \Pr\left[\mathrm{Bin}(q, p_i) > 2p_i q\right] \leq e^{-\frac{1}{3}p_i q} \leq e^{-\frac{1}{3}\varepsilon^2(n/\varepsilon^2)} = e^{-n/3}$$

By the union bound, the probability that $\mu'(i) \leq 2\max\{\varepsilon^2, \mu(i)\}$ for any $1 \leq i \leq n$ is bounded by $n \cdot e^{-n/3} \leq \frac{1}{12.9}$.

By the union bound, the probability of any "bad event" occurring is at most $\frac{1}{32} + \frac{1}{12.9} \leq \frac{1}{9}$. □

Before we present the learning algorithm, we formally define the histogram buckets we wish to learn.

Definition 9.12 (Bucket function)**.** Let $\varepsilon > 0$, $t \geq 2 + \ln(N/\varepsilon^2)/\varepsilon$, and μ be a distribution over Ω of size N. A function $f : \Omega \to \{1, \ldots, t; \infty\}$ is a *bucket function* if for every $x \in \Omega$:

- If $\mu(x) > e^{-\varepsilon(t-2)}$ and $\mathrm{CDF}_\mu(x) \geq \varepsilon$, then $\mu(x) \in e^{-\varepsilon(f(x)\pm 2)}$.
- Otherwise, $f(x) = \infty$ or $\mu(x) \in e^{-\varepsilon(f(x)\pm 2)}$.

Lemma 9.13. *Let h be an $(\varepsilon, \varepsilon)$-approximation function of μ. The function $f(x) = \lceil -\ln h(x)/(2\varepsilon) \rceil$, where values larger than t are mapped to ∞, is a 2ε-bucket function of μ.*

Proof. For every $x \in \Omega$ for which $\mu(x) > e^{-2\varepsilon(t-2)}$ and $\mathrm{CDF}(x) \geq \varepsilon$ (noting that such a value is never mapped to the ∞-bucket):

$$\begin{array}{rcccl} & & \frac{\ln h(x)}{2\varepsilon} - 1 \leq & -f(x) & \leq \frac{\ln h(x)}{2\varepsilon} \\ e^{-2\varepsilon(f(x)+2)} \leq e^{\ln h(x) - 4\varepsilon} & = e^{-4\varepsilon} h(x) \leq & \mu(x) & \leq e^{2\varepsilon} h(x) = e^{\ln h(x) + 2\varepsilon} & \leq e^{-2\varepsilon(f(x)-2)} \end{array}$$

□

Definition 9.14 (Bucket distribution)**.** Let $\varepsilon > 0$, $t \geq 2 + \ln(N/\varepsilon^2)/\varepsilon$, μ be a distribution over Ω and $f : \Omega \to \{1, \ldots, t; \infty\}$ be an ε-bucket function. The *bucket distribution* of μ with respect to f is the distribution μ_f over $\{1, \ldots, t; \infty\}$ for which $\mu_f(i) = \Pr_{x \sim \mu}[f(x) = i]$.

Definition 9.15 (Bucket-transform $T_{\varepsilon,\varepsilon'}$)**.** Let $\varepsilon' \geq 2\varepsilon$. The *bucket transform from ε to ε'*, $T_{\varepsilon,\varepsilon'} : \mathbb{N} \cup \{\infty\} \to \mathbb{N} \cup \{\infty\}$, maps ∞ to itself and every $i \in \mathbb{N}$ to $\left\lceil \frac{\varepsilon}{\varepsilon'} i \right\rceil$.

Lemma 9.16. *Let f be an ε-bucket function of a distribution μ with respect to some $t \geq 2 + \ln(N/\varepsilon^2)/\varepsilon$. For $\varepsilon' \geq 2\varepsilon$ and $t' = 2 + \left\lfloor \frac{\varepsilon}{\varepsilon'}(t-2) \right\rfloor$, the function $g : \Omega \to \{1, \ldots, t'; \infty\}$ defined as $g(x) = T_{\varepsilon,\varepsilon'}(f(x))$ is an ε'-bucket function of μ with respect to t'.*

Proof. For validity, observe that $\ln\left((\varepsilon'/\varepsilon)^2\right)/\varepsilon' \geq 1$ and hence we can obtain:

$$t' = 2 + \left\lfloor \frac{\varepsilon}{\varepsilon'}(t-2) \right\rfloor \geq 1 + \frac{\varepsilon}{\varepsilon'} \cdot \frac{\ln(N/\varepsilon^2)}{\varepsilon} = 1 + \frac{\ln(N/(\varepsilon')^2) + \ln\left((\varepsilon'/\varepsilon)^2\right)}{\varepsilon'} \geq 2 + \frac{\ln(N/(\varepsilon')^2)}{\varepsilon'}$$

Also, observe that $\varepsilon'(t'-2) \leq \varepsilon' \cdot \frac{\varepsilon}{\varepsilon'}(t-2) = \varepsilon(t-2)$ and that $t' \geq \left\lceil \frac{\varepsilon}{\varepsilon'} t \right\rceil$.

Consider x for which $f(x) \neq \infty$. By definition of f, $f(x) \in \varepsilon^{-1} \ln \frac{1}{\mu(x)} \pm 2$. By definition of g, $g(x) = \frac{\varepsilon}{\varepsilon'} \cdot \left(\varepsilon^{-1} \ln \frac{1}{\mu(x)} \pm 2\right) \pm 1 = (\varepsilon')^{-1} \ln \frac{1}{\mu(x)} \pm \left(2\frac{\varepsilon}{\varepsilon'} + 1\right) \subseteq (\varepsilon')^{-1} \ln \frac{1}{\mu(x)} \pm 2$.

Consider x for which $f(x) = \infty$ (and hence $g(x) = \infty$ as well). If $\mathrm{CDF}_\mu(x) \leq \varepsilon$ then $\mathrm{CDF}_\mu(x) \leq \varepsilon'$ as well. Otherwise, $\mu(x) \leq e^{-\varepsilon(t-2)}$. In this case, by the constraint of t', $\mu(x) \leq e^{-\varepsilon'(t'-2)}$ as well. □

Lemma 9.17. *Let $\varepsilon > 0$, $N \geq 1$, $t \geq 2 + \ln(N/\varepsilon^2)/\varepsilon$. Let μ, τ be two distributions over Ω of size N and $f_\mu, f_\tau : \Omega \to \{1, \ldots, t; \infty\}$ be ε-bucket functions for μ and τ respectively. Let μ_{f_μ} and τ_{f_τ} the bucket distributions corresponding to (μ, f_μ) and (τ, f_τ) respectively. In this setting, $D_{\mathrm{H}}(\mu; \tau) \leq \varepsilon'$ for $\varepsilon' = d_{\mathrm{TV}}(\mu_{f_\mu}, \tau_{f_\tau}) + 5\varepsilon + 16\varepsilon^2$.*

Proof. For every $i \in \{1, \ldots, t; \infty\}$, let $B_i^\mu = \{x \in \Omega : f_\mu(x) = i\}$ and $B_i^\tau = \{x \in \Omega : f_\tau(x) = i\}$. Also, let (L_i^μ, R_i^μ) (resp. (L_i^τ, R_i^τ)) be a partition of B_i^μ (resp. B_i^τ) for which $|L_i^\mu| = \min\{|B_i^\mu|, |B_i^\tau|\}$ (resp. $|L_i^\tau| = \min\{|B_i^\mu|, |B_i^\tau|\}$). Let $L^\mu = L_\infty^\mu \cup \bigcup_{i=1}^t L_i^\mu$, and analogously define R^μ, L^τ, R^τ as the corresponding unions.

Let π be a permutation over Ω such that for every $i \in \{1, \ldots, t; \infty\}$, π maps L_i^μ onto L_i^τ, and also maps R^μ onto R^τ. Such a permutation exists since $|L_i^\mu| = |L_i^\tau|$ for every $i \in \{1, \ldots, t; \infty\}$ and $|R^\mu| = N - |L^\mu| = N - |L^\tau| = |R^\tau|$.

Clearly, elements in $L^\mu \setminus L_\infty^\mu$ are mapped to elements in the same bucket, hence $\frac{\mu(x)}{\tau(\pi(x))} \in e^{\pm 4\varepsilon} = 1 \pm (4\varepsilon + 16\varepsilon^2)$ for such elements. The mass of the other elements is bounded by:

$$\mu(B_\infty^\mu) + \sum_{i=1}^t \mu(R_i^\mu) \leq \mu(B_\infty^\mu) + \sum_{i=1}^t e^{-\varepsilon(i-2)} \max\{0, |B_i^\mu| - |B_i^\tau|\}$$

For the first part: every element with $\mu(x) \leq e^{-\varepsilon(t-2)}$ has $\mathrm{CDF}_\mu(x) \leq N \cdot e^{-\varepsilon(\ln(N/\varepsilon^2)/\varepsilon)} = \varepsilon^2 < \varepsilon$, hence for B_∞^μ it suffices to only consider elements with $\mathrm{CDF}_\mu(x) \leq \varepsilon$. Their mass is bounded by ε due to the definition of CDF_μ. For the second part:

$$\begin{aligned}
\sum_{i=1}^t e^{-\varepsilon(i-2)} \max\{0, |B_i^\mu| - |B_i^\tau|\} &\leq \sum_{i=1}^t e^{-\varepsilon(i-2)} \max\left\{0, e^{\varepsilon(i+2)}\mu(B_i^\mu) - e^{\varepsilon(i-2)}\tau(B_i^\tau)\right\} \\
&= \sum_{i=1}^t \max\left\{0, e^{4\varepsilon}\mu(B_i^\mu) - \tau(B_i^\tau)\right\} \\
&= \sum_{i=1}^t \max\left\{0, \mu(B_i^\mu) - \tau(B_i^\tau) + \left(e^{4\varepsilon} - 1\right)\mu(B_i^\mu)\right\} \\
&\leq \sum_{i=1}^t \max\{0, \mu(B_i^\mu) - \tau(B_i^\tau)\} + \left(e^{4\varepsilon} - 1\right)\sum_{i=1}^t \tau(B_i^\tau) \\
&\leq d_{\mathrm{TV}}(\mu_{f_\mu}, \tau_{f_\tau}) + (4\varepsilon + 16\varepsilon^2)
\end{aligned}$$

Overall, $\Pr_\mu\left[\mu(x) \notin (1 \pm (4\varepsilon + 16\varepsilon^2))\tau(x)\right] \leq d_{\mathrm{TV}}(\mu_{f_\mu}, \tau_{f_\tau}) + 5\varepsilon + 16\varepsilon^2$. □

Lemma 9.18. *Let $N \geq 1$, $\varepsilon > 0$ and $t \geq 2 + \ln(N/4\varepsilon^2)/2\varepsilon$. Let μ be a distribution over Ω and let f_μ be a 2ε-bucket function with respect to μ and t. Let ν be a distribution over $\{1, \ldots, t; \infty\}$ for which $d_{\mathrm{TV}}(\mu_{f_\mu}, \nu) \leq 6\varepsilon$. In this setting, the following constraint problem is algorithmically solvable given full access to t, ε, N and ν:*

- $\sum_{i=1}^{t} N_i \leq N$.
- $\sum_{i=1}^{t} N_i p_i \leq 1$.
- $\sum_{i=1}^{t} |N_i p_i - \nu(i)| \leq 12\varepsilon$.
- *For every $1 \leq i \leq t$: $N_i \geq 0$ is an integer.*
- *For every $1 \leq i \leq t$: $e^{-2\varepsilon(i+2)} \leq p_i \leq e^{-2\varepsilon(i-2)}$.*

Proof. Solvability: recall Definition 9.14, and observe that the following is a feasible solution: for every $1 \leq i \leq t$, $N_i = |\{x : f_\mu(x) = i\}|$ and $p_i = \frac{\mu_{f_\mu}(i)}{N_i}$.

Computability: by simple arithmetic $N_i \leq (12\varepsilon + \nu(i))e^{2\varepsilon(i+2)}$ for every $1 \leq i \leq t$, hence the search range for $(N_1, \ldots, N_t)$ is finite and can be exhausted algorithmically. The algorithm considers every assignment of $(N_1, \ldots, N_t)$ within their bounds, solves the corresponding LP problem (a sum of t absolute values can be converted to to 2^t linear constraints) and tests the feasibility of the result assignment. □

Note that the time complexity relating to the above lemma is large. If we allow the algorithm to find a solution relaxing the third condition to $\sum_{i=1}^{t} |N_i p_i - \nu(i)| \leq 24\varepsilon$ for ν satisfying the assertions of the lemma, then we can greatly reduce the time complexity, and this is still sufficient for the histogram learning task. We do not prove this here.

Algorithm 15 solves the histogram learning task by drawing $O(t/\hat{\varepsilon}^2)$ sample elements, and estimating their mass to associate them with their buckets, possibly with a ± 2-additive shift.

Lemma 9.19 (Learn-histogram-buckets)**.** *Algorithm 15 makes $O(\log(N/\hat{\varepsilon})/\hat{\varepsilon}^3)$ calls to the $\hat{\varepsilon}$-error $(\hat{\varepsilon}, \hat{\varepsilon})$-explicit sampling oracle, and with probability at least $2/3$ returns a distribution τ_B that is $6\hat{\varepsilon}$-close to a bucket distribution of the form μ_{f_μ} for some $2\hat{\varepsilon}$-bucket function f_μ of μ.*

Proof. By Observation 9.6, we can see the $\hat{\varepsilon}$-error $(\hat{\varepsilon}, \hat{\varepsilon})$-explicit sampling oracle as a propagation of the following:

- An $(\hat{\varepsilon}, \hat{\varepsilon})$-approximation function g_{truth} of μ.
- An arbitrary error function $g_{\mathrm{err}} : \Omega \to [0, 1]$.
- A correctness vector $u \in \{0, 1\}^\Omega$ whose entries are drawn independently, where $\Pr[u_x = 1] \geq 1 - r$ for every $x \in \Omega$.
- The estimation outcome of the oracle for x is $h(x) = g_{\mathrm{truth}}(x)$ if $u_x = 1$ and $h(x) = g_{\mathrm{err}}(x)$ if $u_x = 0$.

This way, the analysis can use $h(y)$ instead of $\hat{p}_y$ for the samples, noting that $h(y)$ is also defined for non-sampled $y \in \Omega$.

Algorithm 15: Procedure Learn-histogram-buckets($\hat{\varepsilon}; \mu$)
Oracle: The $\hat{\varepsilon}$-error $(\hat{\varepsilon}, \hat{\varepsilon})$-explicit sampling oracle, $\hat{\varepsilon} < 1/27$.
Output: A distribution that is $6\hat{\varepsilon}$-close to some $2\hat{\varepsilon}$-bucket distribution of μ.
Success probability: 2/3.

1. Let $t \leftarrow \left\lceil \ln(N/\hat{\varepsilon}^2)/2\hat{\varepsilon} \right\rceil + 2$.
2. Let $q \leftarrow \left\lceil (t+1)/\hat{\varepsilon}^2 \right\rceil$.
3. Set $X_1, \ldots, X_t; X_\infty \leftarrow 0$.
4. For q times:
   (a) Explicitly draw $y \sim \mu$ and obtain $\hat{p}_y$.
   (b) Set $f(y) \leftarrow \left\lceil -\frac{\ln \hat{p}_y}{2\hat{\varepsilon}} \right\rceil$.
   (c) If $f(y) = 0$:
      i. Set $f(y) \leftarrow 1$.
   (d) If $f(y) > t$:
      i. Set $f(y) \leftarrow \infty$.
   (e) Set $X_{f(y)} \leftarrow X_{f(y)} + 1$.
5. Let τ_B the distribution defined as $\tau_B(i) = X_i/q$ for every $i \in \{1, \ldots, t; \infty\}$.
6. Return τ_B.

Let f and f' be the functions that map every $x \in \Omega$ to its $2\hat{\varepsilon}$-bucket according to $g_{\text{truth}}(x)$ and $h(x)$ respectively. More precisely, the bucket associated with the mass p (which can be obtained from g_{truth} or from h) is $\lceil -\ln p/(2\hat{\varepsilon}) \rceil$ (or ∞ if larger than t). By Lemma 9.13, f is indeed a bucket function of μ.

Since μ_f and $\mu_{f'}$ are mappings of μ with respect to f and f', $d_{\text{TV}}(\mu_f, \mu_{f'}) \leq \sum_{x \in \Omega: f(x) \neq f'(x)} \mu(x)$. For every x, the probability that $g_{\text{truth}}(x) \neq h(x)$ is bounded by $\hat{\varepsilon}$, and hence $\mathrm{E}_u[d_{\text{TV}}(\mu_f, \mu_{f'})] \leq \hat{\varepsilon}$. By Markov's inequality, with probability at least 4/5 over the choice of the correctness vector u, $d_{\text{TV}}(\mu_f, \mu_{f'}) \leq 5\hat{\varepsilon}$.

By Lemma 9.11, with probability at least 8/9, the distribution τ_B constructed by the algorithm is $\hat{\varepsilon}$-close to $\mu_{f'}$. By the triangle inequality and the union bound, with probability at least $1 - 1/9 - 1/5 > 2/3$, $d_{\text{TV}}(\tau_B, \mu_f) \leq d_{\text{TV}}(\tau_B, \mu_f) + d_{\text{TV}}(\mu_{f'}, \mu_f) \leq \hat{\varepsilon} + 5\hat{\varepsilon} = 6\hat{\varepsilon}$.

The sample complexity is trivial. □

The histogram learning algorithm works by converting the (approximated) distribution over buckets back to a distribution over Ω.

Lemma 9.20. *Let μ be a distribution over $\Omega = \{1, \ldots, N\}$. Algorithm 16 solves the ε-histogram learning task at the cost of $O(\log(N/\varepsilon)/\varepsilon^3)$ calls to the $\frac{1}{300}\varepsilon$-explicit sampling oracle.*

Proof. By Lemma 9.19, with probability 2/3, the call to Learn-histogram-buckets returns a distribution τ_B that is $6\hat{\varepsilon}$-close to some $2\hat{\varepsilon}$-bucket distribution of μ. Let f_μ be a $2\hat{\varepsilon}$-bucket function of μ for which τ_B is $6\hat{\varepsilon}$-close to μ_{f_μ}. Lemma 9.18 implies that the constraint problem defined by the algorithm in Step 2 is solvable.

Observe that $\tau_B(\infty) \leq \mu_{f_\mu}(\infty) + 6\hat{\varepsilon} \leq \frac{4\hat{\varepsilon}^2}{N} \cdot N + 6\hat{\varepsilon} = 6\hat{\varepsilon} + 4\hat{\varepsilon}^2$.

Algorithm 16: Procedure Learn-histogram$(\varepsilon;\mu)$
Oracle: The $\hat{\varepsilon}$-error $(\hat{\varepsilon},\hat{\varepsilon})$-explicit sampling oracle for $\hat{\varepsilon}=\frac{1}{300}\varepsilon$.

1. Let $\tau_B \leftarrow$ Learn-histogram-buckets$(\hat{\varepsilon},\mu)$.
2. Solve the following constraints problem: (Solvable by Lemma 9.18)
   - $\sum_{i=1}^{t} N_i \leq N$.
   - $\sum_{i=1}^{t} N_i p_i \leq 1$.
   - $\sum_{i=1}^{t} |N_i p_i - \tau_B(i)| \leq 12\hat{\varepsilon}$.
   - For every $1 \leq i \leq t$: $N_i \geq 0$ is an integer.
   - For every $1 \leq i \leq p$: $e^{-2\hat{\varepsilon}(i+2)} \leq p_i \leq e^{-2\hat{\varepsilon}(i-2)}$.
3. Let $s \leftarrow \sum_{i=1}^{t} N_i p_i$.
4. Construct a function $f' : \Omega \to \{1,\ldots,t;\infty\}$ such that for every $1 \leq i \leq t$ there are exactly N_i elements for which $f(x)=i$. The other elements are mapped to ∞.
5. Construct a distribution τ' over Ω where for every $1 \leq i \leq t$ there are exactly N_i elements whose probability mass is exactly p_i/s. The other $N - \sum_{i=1}^{t} N_i$ elements have zero mass.
6. Return τ'.

Observe that $\tau'(\infty)=0$, and note that $s=\sum_{i=1}^{t} N_i p_i \geq \sum_{i=1}^{t} \tau_B(i) - 12\hat{\varepsilon} = (1-\tau_B(\infty)) - 12\hat{\varepsilon} \geq 1-(18\hat{\varepsilon}+4\hat{\varepsilon}^2)$. Also, $s \leq 1$ by the constraints of the construction.

Since $s = 1 \pm (18\hat{\varepsilon}+4\hat{\varepsilon}^2)$, $s^{-1} = 1 \pm 20\hat{\varepsilon}$ and hence:

$$\begin{aligned}
\sum_{i=1}^{t} |N_i p_i/s - \tau_B(i)| &= \sum_{i=1}^{t} |(1\pm 20\hat{\varepsilon}) N_i p_i - \tau_B(i)| \\
&\leq 20\hat{\varepsilon} \sum_{i=1}^{t} N_i p_i + \sum_{i=1}^{t} |N_i p_i - \tau_B(i)| \\
&\leq 20\hat{\varepsilon}\cdot 1 + 12\hat{\varepsilon} = 32\hat{\varepsilon}
\end{aligned}$$

Considering Definition 9.14 and Lemma 9.16 with respect to $\hat{\varepsilon}$ and $\varepsilon' = 16\cdot 2\hat{\varepsilon}$, let:

- $t' = 2 + \lfloor (t-2)/16 \rfloor$.
- ν_B be the map of τ_B according to $T_{2\hat{\varepsilon},32\hat{\varepsilon}}$, that is, $\nu_B(j) = \sum_{i:T_{2\hat{\varepsilon},32\hat{\varepsilon}}(i)=j} \tau_B(i)$.
- f' be the $32\hat{\varepsilon}$-bucket function of τ' with respect to t', defined such that every element with mass p_i/s is mapped to the $T_{2\hat{\varepsilon},32\hat{\varepsilon}}(i)$th bucket.
- f'_μ be the $32\hat{\varepsilon}$-bucket function of μ with respect to t' that is constructed from f_μ by Lemma 9.16.

Observe that ν_B is $6\hat{\varepsilon}$-close to $\mu_{f'_\mu}$ since τ_B is $6\hat{\varepsilon}$-close to μ_{f_μ}, and that $\mu_{f'_\mu}$ is a $32\hat{\varepsilon}$-bucket distribution of μ.

By the construction, $\tau'_{f'}(\infty)=0$ and $\tau'_{f'}(i) = N_i p_i/s$ for $1 \leq i \leq t$. We can use the bound for $\sum_{i=1}^{t} |N_i p_i/s - \tau_B(i)|$ to obtain:

$$\begin{aligned} d_{\mathrm{TV}}(\tau'_{f'}, \nu_B) &\leq d_{\mathrm{TV}}(\tau'_f, \tau_B) \\ &= \frac{1}{2}\tau_B(\infty) + \frac{1}{2}\sum_{i=1}^{t} |N_i p_i/s - \tau_B(i)| \leq \frac{1}{2}\cdot(6\hat{\varepsilon} + 4\hat{\varepsilon}^2) + \frac{1}{2}\cdot 32\hat{\varepsilon} = 19\hat{\varepsilon} + 2\hat{\varepsilon}^2 \end{aligned}$$

By the triangle inequality, $d_{\mathrm{TV}}(\mu_{f'_\mu}, \tau'_{f'}) \leq d_{\mathrm{TV}}(\mu_{f'_\mu}, \nu_B) + d_{\mathrm{TV}}(\nu_B, \tau'_{f'}) \leq 6\hat{\varepsilon} + (19\hat{\varepsilon} + 2\hat{\varepsilon}^2) \leq 25\hat{\varepsilon} + 4\hat{\varepsilon}^2$. By Lemma 9.17, $D_{\mathrm{H}}(\mu; \tau') \leq \varepsilon'$ for

$$\varepsilon' = d_{\mathrm{TV}}(\mu_{f'_\mu}, \tau'_{f'}) + (5(32\hat{\varepsilon}) + 16(32\hat{\varepsilon})^2) \leq (25\hat{\varepsilon} + 4\hat{\varepsilon}^2) + (160\hat{\varepsilon} + 16384\hat{\varepsilon}^2) = 185\hat{\varepsilon} + 16388\hat{\varepsilon}^2 \leq 300\hat{\varepsilon} = \varepsilon$$

□

At this point we recall Theorem 9.21 and prove it.

Theorem 9.21 (Learning histograms)**.** *We can use* $O\left(\frac{\log N \log\log N}{\varepsilon^7} \cdot \mathrm{poly}(\log \varepsilon^{-1})\right)$ *conditional samples to solve the* ε*-histogram learning.*

Proof. This is an application of Lemma 9.10 over the $O(\log(N/\varepsilon)/\varepsilon^3)$-sample algorithm for learning histograms stated in Lemma 9.20. □

Since label-invariant properties are determined by histograms, we can obtain a universal tester for label-invariant properties.

Corollary 9.22. *There exists a universal tester for* ε*-testing every label-invariant property* $\mathcal{P}$ *using* $O(\log N/\varepsilon^7 \cdot \mathrm{poly}(\log \varepsilon^{-1}))$ *conditional samples.*

Proof. We learn a distribution τ for which $D_{\mathrm{H}}(\mu; \tau) \leq \frac{1}{4}\varepsilon$ and accept if there exists any distribution $\mu' \in \mathcal{P}$ for which $D_{\mathrm{H}}(\mu'; \tau) \leq \frac{1}{4}\varepsilon$. By two applications of Lemma 3.9, if we have accepted due to some τ then there are two permutations π and π' such that $d_{\mathrm{TV}}(\mu, \pi\tau) \leq \frac{1}{2}\varepsilon$ and $d_{\mathrm{TV}}(\mu', \pi'\tau) \leq \frac{1}{2}\varepsilon$. By the triangle inequality (and invariance under permutations) we obtain $d_{\mathrm{TV}}(\mu, (\pi \circ (\pi')^{-1})\mu') \leq \varepsilon$ as required. □

9.3 Total-variation distance estimation

We first prove a generic lemma about a reduction from the (c, ε)-peek model to the fully conditional model for inputs consisting of multiple distributions. The analysis here has a penalty of $O(1/\varepsilon)$ in comparison to Lemma 9.10, since we no longer assume that a queried element x is drawn from the distribution it is queried from, which requires the use of Corollary 4.2 (worst case cost) rather than Corollary 4.3 (expected cost).

Lemma 9.23. *Consider an algorithm* $\mathcal{A}$ *whose input is a* k*-tuple* $\vec{\mu} = (\mu_1, \ldots, \mu_k)$ *of distributions over* $\Omega_1, \ldots, \Omega_k$ *(respectively), and its output is an element of a discrete set* R*. Assume that* $\mathcal{A}$ *draws at most* q *samples and makes at most* q *calls to the* (c, ε)*-peek oracle. Let* $N = \max\{|\Omega_1|, \ldots, |\Omega_k|\}$*.*

Assume that for every input $\vec{\mu}$ *there exists a set* $R_{\vec{\mu}} \subseteq R$ *for which* $\Pr\left[\mathcal{A}(\vec{\mu}) \in R_{\vec{\mu}}\right] > \frac{2}{3}$ *for every possible valid outcome sequence of the* (c, ε)*-oracle (i.e., one that comes from an* (ε, c)*-approximation function corresponding to the oracle).*

In this setting, there exists an algorithm $\mathcal{A}'$ in the fully conditional model whose sample complexity is $O(q \cdot (\log\log N + 1/\varepsilon^2 c + 1/\varepsilon^5) \cdot \mathrm{poly}(\log q, \log c^{-1}, \log \varepsilon^{-1}))$, such that for every input $\vec{\mu}$, $\Pr\left[\mathcal{A}'(\vec{\mu}) \in R_{\vec{\mu}}\right] > \frac{5}{8}$.

Proof. We run $\mathcal{A}$ and simulate the outcome of the ε-peek oracle. In each call to the (c,ε)-peek oracle with $x \in \Omega_i$ and μ_i, we call Estimate-element with parameters (μ_i, c, ε) on x_i (Theorem 4.1). We amplify the success probability to $1 - \frac{1}{24q}$ using the median of $\lceil 30\ln(12q) \rceil$ such calls (Observation 3.32(d)). Each time we estimate the probability mass of an element, we record it in a "history". If the same element is queried again later, we use the history record rather than calling the Estimate-element procedure again. This guarantees the consistency of the oracle (required by Definition 9.7).

The probability to have a wrong estimation is bounded by $q \cdot \frac{1}{24q} = \frac{1}{24}$. Hence, the probability to correctly simulate the (c,ε)-peek oracle is at least $23/24$. If the simulation is correct, then the output of the simulated $\mathcal{A}$ belongs to $R_{\vec{\mu}}$ with probability at least $2/3$. Overall, the probability of the simulation to output an element in $R_{\vec{\mu}}$ is at least $2/3 - 1/24 = 5/8$.

By Corollary 4.2, the worst-case complexity of a single estimation of x is $O(\mathrm{poly}(\log c^{-1}, \log \varepsilon^{-1})) \cdot O(\log\log N + \frac{1}{\varepsilon^2 c} + \frac{\log^6 \varepsilon^{-1}}{\varepsilon^5})$. We repeat this $O(\log q)$ times for amplification for q requests. Overall, the expected sample complexity is at most $O(q \cdot (\log\log N + 1/\varepsilon^2 c + 1/\varepsilon^5) \cdot \mathrm{poly}(\log q, \log c^{-1}, \log \varepsilon^{-1}))$. □

Lemma 9.24. *Consider a testing algorithm $\mathcal{A}$ for some property $\mathcal{P}$ with success probability $2/3$ whose input is a k-tuple $\vec{\mu} = (\mu_1, \ldots, \mu_k)$ of distributions over $\Omega_1, \ldots, \Omega_k$ (respectively). Assume that $\mathcal{A}$ draws at most q samples and makes at most q calls to the (c,ε)-peek oracle. Let $N = \max\{|\Omega_1|, \ldots, |\Omega_k|\}$.*

In this setting, there exists a testing algorithm $\mathcal{A}'$ for $\mathcal{P}$ with success probability $2/3$ in the fully conditional model whose sample complexity is $O(q \cdot (\log\log N + 1/\varepsilon^2 c + 1/\varepsilon^5) \cdot \mathrm{poly}(\log q, \log c^{-1}, \log \varepsilon^{-1}))$.

Proof. Let $R = \{\textsc{accept}, \textsc{reject}\}$. By Lemma 9.23, there exists a testing algorithm $\mathcal{A}''$ in the conditional model with the guaranteed sample complexity and success probability at least $5/8$. We define $\mathcal{A}'$ as a majority-of-3 amplification of $\mathcal{A}''$. The success probability of $\mathcal{A}'$ is at least $2/3$ by Observation 9.9. □

Most of this subsection is dedicated to an algorithm for estimating $d_{\mathrm{TV}}(\mu,\tau)$ within $\pm\varepsilon$-additive error at the cost of $O(1/\varepsilon^2)$ samples and $O(1/\varepsilon^2)$ calls to the $\frac{1}{6}\varepsilon$-peek oracle.

Lemma 9.25. *For every pair of c-truncated functions $f_\mu, f_\tau : \Omega \to [0,1]$ with respect to μ and τ,*

$$d_{\mathrm{TV}}(\mu,\tau) = \frac{1}{2}\left(\mathop{\mathrm{E}}_{x\sim\mu}\left[\max\left\{0, 1 - \frac{f_\tau(x)}{\mu(x)}\right\}\right] + \mathop{\mathrm{E}}_{x\sim\tau}\left[\max\left\{0, 1 - \frac{f_\mu(x)}{\tau(x)}\right\}\right]\right) \pm 2c$$

We defer the proof of Lemma 9.25 to Appendix F.

Lemma 9.26 (Estimate-bounded-ratio)**.** *Let μ be a distribution over Ω and let $f, g : \Omega \to [0,1]$ be two inaccessible functions. Assume that we have oracle access to functions $\hat{f}, \hat{g} : \Omega \to [0,1]$*

Algorithm 17: Procedure Estimate-bounded-ratio$(\mu, \varepsilon, f, g; \hat{f}, \hat{g})$
Input: f and g, inaccessible to the algorithm.
Input: Oracle access to $\hat{f}(x) \in (1 \pm \varepsilon) f(x)$ for every x.
Input: Oracle access to $\hat{g}(x) \in (1 \pm \varepsilon) g(x)$ for every x.
Output: $Y = \mathrm{E}_{x \sim \mu}\left[\max\left\{0, 1 - \frac{f(x)}{g(x)}\right\}\right] \pm 4\varepsilon$.
Success probability: 5/6.

1. $M \leftarrow \lceil 6/\varepsilon^2 \rceil$.
2. For i from 1 to M:
   (a) $x_i \sim \mu$.
   (b) $\hat{p}_i \leftarrow \hat{f}(x_i)$.
   (c) $\hat{q}_i \leftarrow \hat{g}(x_i)$.
   (d) $X_i \leftarrow \min\left\{1, \frac{\hat{p}_i}{\hat{q}_i}\right\}$.
3. Let $\bar{X} = \frac{1}{M}\sum_{i=1}^{M} X_i$.
4. Return $Y = 1 - \bar{X}$.

such that for every $x \in \Omega$, $\hat{f}(x) \in (1 \pm \varepsilon) f(x)$ and $\hat{g}(x) \in (1 \pm \varepsilon) g(x)$. Algorithm 17 estimates $\mathrm{E}_{\mu}\left[\max\left\{0, 1 - \frac{f(x)}{g(x)}\right\}\right]$ within $\pm 4\varepsilon$ and success probability 5/6, at the cost of $O(1/\varepsilon^2)$ oracle calls.

Proof. It suffices to show that $\bar{X}$ estimates $\mathrm{E}_{\mu}\left[\min\left\{1, \frac{f(x)}{g(x)}\right\}\right]$ within $\pm 4\varepsilon$-error with probability at least 5/6.

We explicitly bound the additive error in a single trial. If $\frac{f(x)}{g(x)} \geq \frac{1+\varepsilon}{1-\varepsilon}$, then $\frac{\hat{f}(x)}{\hat{g}(x)} \geq 1$, and hence $\min\left\{1, \frac{\hat{f}(x)}{\hat{g}(x)}\right\} = \min\left\{1, \frac{f(x)}{g(x)}\right\} = 1$. If $\frac{f(x)}{g(x)} \leq \frac{1+\varepsilon}{1-\varepsilon}$, then the error $(\frac{1\pm\varepsilon}{1\pm\varepsilon} - 1)\frac{f(x)}{g(x)}$ is bounded by $\pm 3\varepsilon$.

For $q = \lceil 6/\varepsilon^2 \rceil$, let $X_1, \ldots, X_q$ be independent samples of $\min\left\{1, \frac{\hat{f}(x)}{\hat{g}(x)}\right\}$, each costing two oracle calls. Let $\bar{X} = \frac{1}{q}\sum_{i=1}^{q} X_i$. Clearly, all X_is are bounded between 0 and 1, hence their variance is bounded by 1 as well. An average over $\lceil 6/\varepsilon^2 \rceil$ trials has variance $\mathrm{Var}[\bar{X}] \leq \frac{1}{6}\varepsilon^2$, and hence by Chebyshev inequality, the probability to deviate by more than ε is bounded by 1/6.

Overall, with probability at least 5/6,

$$\bar{X} = \mathop{\mathrm{E}}_{\mu}\left[\min\left\{1, \frac{\hat{f}(x)}{\hat{g}(x)}\right\}\right] \pm \varepsilon = \left(\mathop{\mathrm{E}}_{\mu}\left[\min\left\{1, \frac{f(x)}{g(x)}\right\}\right] \pm 3\varepsilon\right) \pm \varepsilon = \mathop{\mathrm{E}}_{\mu}\left[\min\left\{1, \frac{f(x)}{g(x)}\right\}\right] \pm 4\varepsilon \quad \square$$

Lemma 9.27. *Let μ and τ be two distributions over $\Omega = \{1, \ldots, N\}$. Algorithm 18 estimates $d_{\mathrm{TV}}(\mu, \tau)$ within $\pm\varepsilon$-additive error at the cost of $O(1/\varepsilon^2)$ samples and $O(1/\varepsilon^2)$ calls to the $\frac{1}{6}\varepsilon$-peek oracle.*

Proof. By Observation 9.8, the $\hat{\varepsilon}$-peek oracles (for μ and τ) can be seen as query oracles of $(\hat{\varepsilon}, \hat{\varepsilon})$-approximation functions h_μ and h_τ. By Observation 9.4, these h_μ and h_τ can be seen as $(1 \pm \hat{\varepsilon})$-approximations of $\hat{\varepsilon}$-truncated functions f_μ, f_τ

Algorithm 18: Procedure Estimate-$d_{\mathrm{TV}}(\varepsilon;\mu,\tau)$
Oracle: The $(\hat{\varepsilon},\hat{\varepsilon})$-peek oracles of μ and τ for $\hat{\varepsilon}=\frac{1}{6}\varepsilon$.

1. Let f_μ be a non-accessible, arbitrary $\hat{\varepsilon}$-truncated function of μ, implicitly defined by the output of the peek oracle.
2. Let f_τ be a non-accessible, arbitrary $\hat{\varepsilon}$-truncated function of τ, implicitly defined by the output of the peek oracle.
3. Consider the following functions:
   - $f(x)$: f_μ (not accessible).
   - $g(x)$: f_τ (not accessible).
   - $\hat{f}(x)$ is the oracle call to the $(\hat{\varepsilon},\hat{\varepsilon})$-peek oracle in μ.
   - $\hat{g}(x)$ is the oracle call to the $(\hat{\varepsilon},\hat{\varepsilon})$-peek oracle in τ.
4. Let $X^\mu \leftarrow$ Estimate-bounded-ratio$(\mu,\hat{\varepsilon},f,g;\hat{f},\hat{g})$.
5. Let $X^\tau \leftarrow$ Estimate-bounded-ratio$(\tau,\hat{\varepsilon},g,f;\hat{g},\hat{f})$.
6. Return $\frac{1}{2}X^\mu+\frac{1}{2}X^\tau$.

By Lemma 9.26, with probability at least 2/3 (a union bound over two 5/6-success events):

$$\begin{aligned} X^\mu &= \mathop{\mathrm{E}}_{\mu}\left[\max\left\{0,\frac{f_\tau(x)}{f_\mu(x)}\right\}\right]\pm 4\hat{\varepsilon} \\ X^\tau &= \mathop{\mathrm{E}}_{\tau}\left[\max\left\{0,\frac{f_\mu(x)}{f_\tau(x)}\right\}\right]\pm 4\hat{\varepsilon} \end{aligned}$$

If this happens, then:

$$\begin{aligned} \frac{1}{2}X^\mu+\frac{1}{2}X^\tau &= \frac{1}{2}\mathrm{E}[X^\mu]+\frac{1}{2}\mathrm{E}[X^\tau]\pm 4\hat{\varepsilon} \\ [\text{Lemma 9.25}] &= (d_{\mathrm{TV}}(\mu,\tau)\pm 2\hat{\varepsilon})\pm 4\hat{\varepsilon} = d_{\mathrm{TV}}(\mu,\tau)\pm 6\hat{\varepsilon} = d_{\mathrm{TV}}(\mu,\tau)\pm\varepsilon \end{aligned}$$

□

We now recall Theorem 1.4 and prove it.

Theorem 1.4 (Almost-tight upper bound for distance estimation)**.** *Let μ, τ be two distributions over $\Omega=\{1,\dots,N\}$ and $\varepsilon>0$. There exists an algorithm for estimating $d_{\mathrm{TV}}(\mu,\tau)$ within ε-additive error using $O((\log\log N/\varepsilon^2+1/\varepsilon^7)\cdot\mathrm{poly}(\log\varepsilon^{-1}))$ conditional samples.*

Proof. This is an application of Lemma 9.10 over the $O(1/\varepsilon^2)$-sample algorithm for estimating $d_{\mathrm{TV}}(\mu,\tau)\pm\varepsilon$ stated in Lemma 9.27. □

9.4 Non-tolerant testing of equivalence

In this subsection we present a non-tolerant ε-test for equivalence of two distributions μ and τ. Our result reduces the polynomial degree of $1/\varepsilon$ in comparison to the trivial reduction from estimating the distance between μ and τ within $\pm\frac{1}{2}\varepsilon$-additive error.

Algorithm 19: Procedure Equivalence-Test-core$(\varepsilon; \mu, \tau)$
Oracle: The $(\frac{1}{16}\varepsilon, \frac{1}{16}\varepsilon)$-peek oracles for μ and τ.
1. For $\lceil 3/\varepsilon \rceil$ times:
   (a) Draw $x \sim \mu$.
   (b) Call the $(\frac{1}{16}\varepsilon, \frac{1}{16}\varepsilon)$-peek oracle for μ, x to obtain $\hat{p}$.
   (c) Call the $(\frac{1}{16}\varepsilon, \frac{1}{16}\varepsilon)$-peek oracle for τ, x to obtain $\hat{q}$.
   (d) If $|\hat{q}/\hat{p} - 1| > \varepsilon/4$:
      i. Return REJECT.
2. Return ACCEPT.

Lemma 9.28 (Based on [RS09])**.** *Let μ and τ be two distributions over $\Omega = \{1, \ldots, N\}$. Algorithm 19 distinguishes between $\tau = \mu$ and $d_{\mathrm{TV}}(\tau, \mu) > \varepsilon$ using $O(1/\varepsilon)$ independent samples from μ and $O(1/\varepsilon)$ calls to the $(\frac{1}{16}\varepsilon, \frac{1}{16}\varepsilon)$-peek oracle.*

Proof. Let $\hat{\varepsilon} = \frac{1}{16}\varepsilon$. By Observation 9.8, the $(\hat{\varepsilon}, \hat{\varepsilon})$-peek oracles for μ and τ can be seen as query oracles to $(\hat{\varepsilon}, \hat{\varepsilon})$-approximation functions f_μ and f_τ (respectively).

If $\mu = \tau$ and $\mathrm{CDF}_\mu(x) \geq \hat{\varepsilon}$, then we expect that $\left| \frac{f_\tau(x)}{f_\mu(x)} - 1 \right| = \left| \frac{1 \pm \hat{\varepsilon}}{1 \pm \hat{\varepsilon}} - 1 \right| \leq 3\hat{\varepsilon} < \frac{1}{4}\varepsilon$.

If $\tau(x) < (1 - \varepsilon/2)\mu(x)$ and $\mathrm{CDF}_\mu(x) \geq \hat{\varepsilon}$, then we expect that $1 - \frac{f_\tau(x)}{f_\mu(x)} \geq 1 - \frac{1 \pm \hat{\varepsilon}}{1 \pm \hat{\varepsilon}} \cdot \left(1 - \frac{1}{2}\varepsilon\right) > \frac{1}{4}\varepsilon$.

If $\mu = \tau$, then in every iteration, the probability to draw x for which $\mathrm{CDF}_\mu(x) \geq \hat{\varepsilon}$ is at least $1 - \hat{\varepsilon}$. By the union bound, the probability to reject is at most $\lceil 3/\varepsilon \rceil \cdot \hat{\varepsilon} \leq \frac{4}{\varepsilon} \cdot \frac{1}{16}\varepsilon = \frac{1}{4}$.

For $d_{\mathrm{TV}}(\mu, \tau) > \varepsilon$, let $A = \{x : \tau(x) < (1 - \varepsilon/2)\mu(x)\}$. By definition of the total variation distance,

$$\begin{aligned} d_{\mathrm{TV}}(\tau, \mu) = \sum_{\mu(x) > \tau(x)} (\mu(x) - \tau(x)) &= \sum_{\mu(x) > \tau(x)} \mu(x) \left(1 - \frac{\tau(x)}{\mu(x)}\right) \\ &= \mathop{\mathrm{E}}_{x \sim \mu} \left[\max \left\{ 0, 1 - \frac{\tau(x)}{\mu(x)} \right\} \right] \\ &\leq \mu(A) \cdot 1 + \mu(\neg A) \cdot \frac{1}{2}\varepsilon \leq \mu(A) + \frac{1}{2}\varepsilon \end{aligned}$$

Since $d_{\mathrm{TV}}(\tau, \mu) > \varepsilon$, we obtain that $\mu(A) > \frac{1}{2}\varepsilon$. The probability to draw an A-sample from μ which has $\mathrm{CDF}_\mu(x) \geq \hat{\varepsilon}$, and hence reject, is at least $1 - \left(1 - (\frac{1}{2}\varepsilon - \hat{\varepsilon})\right)^{\lceil 3/\varepsilon \rceil} = 1 - \left(1 - \frac{7}{16}\varepsilon\right)^{\lceil 3/\varepsilon \rceil} \geq 1 - e^{-21/16} > \frac{2}{3}$. □

Lemma 9.29. *Let μ, τ be two distributions over $\Omega = \{1, \ldots, N\}$. There exists an algorithm for distinguishing, with probability at least $5/8$, between the case where $\mu = \tau$ and the case where $d_{\mathrm{TV}}(\mu, \tau) > \varepsilon$ using:*

- $O((\log \log N/\varepsilon + 1/\varepsilon^6) \cdot \mathrm{poly}(\log \varepsilon^{-1}))$ *conditional samples at worst-case.*
- $O((\log \log N/\varepsilon + 1/\varepsilon^5) \cdot \mathrm{poly}(\log \varepsilon^{-1}))$ *conditional samples in expectation if $\mu = \tau$.*

Proof. We simulate every peek call of Algorithm 19 using Estimate-element and amplify its confidence by taking the median of $O(\log \varepsilon^{-1})$ independent calls. Lemma 9.23 implies the correctness and the

worst-case complexity of this reduction. Since the success probability of the core algorithm is at least 2/3 and the reduction error is at most 1/24, the success probability of the result algorithm is at least 5/8.

Observe that if $\mu = \tau$ then every call to Estimate-element (for answering a peek call) is performed on a value x that was sampled from a distribution that is identical to the one for which it is queried. This allows the use of Corollary 4.3 to obtain the expected-case complexity of the reduction when $\mu = \tau$. □

We recall Theorem 1.3 and prove it.

Theorem 1.3 (Almost-tight upper bound for equivalence testing)**.** *Let μ, τ be two distributions over $\Omega = \{1, \ldots, N\}$ and $\varepsilon > 0$. There exists an algorithm for distinguishing between the case where $\mu = \tau$ and the case where $d_{\mathrm{TV}}(\mu, \tau) > \varepsilon$, using $O((\log\log N/\varepsilon + 1/\varepsilon^5) \cdot \mathrm{poly}(\log \varepsilon^{-1}))$ conditional samples.*

Proof. Let $Q = O((\log\log N/\varepsilon + 1/\varepsilon^5) \cdot \mathrm{poly}(\log \varepsilon^{-1}))$ be the expected number of samples of the algorithm that is guaranteed by Lemma 9.29 in the case where $\mu = \tau$ (for any choice of μ). Consider the following algorithm: we run 45 independent instances of the algorithm of Lemma 9.29 and take the majority answer, with the exception that if we make our $(540Q + 1)$st sample, then we immediately reject and terminate the run.

If $\mu = \tau$, then by Markov's inequality, the probability to draw the $(540Q + 1)$st sample is smaller than 1/12. Hence, with probability 11/12 the core algorithm runs successfully, and by Observation 9.9, it accepts with probability at least 3/4. By the union bound, we accept with probability at least 2/3.

If $d_{\mathrm{TV}}(\mu, \tau) > \varepsilon$, then either we terminate after $540Q + 1$ queries and reject, or the core algorithm runs successfully and rejects with probability at least 3/4. The probability to reject is at least $3/4 > 2/3$. □

10 Lower bounds

10.1 Tight lower bound for the (c, ε)-estimation task

In this subsection we show a tight lower bound of $\Omega(\log\log N)$ for estimating the probability mass of individual elements using conditional samples. Since the demonstrating distributions are uniform over their support, the expected case and the worst case are identical.

For some integer $1 \le k \le \log N$, we define D_k as the following distribution over inputs (in themselves distributions over $\{1, \ldots, N\}$): we draw a set $K \subseteq \{1, \ldots, N\}$ such that every element belongs to K with probability 2^{-k} independently, and then return the uniform distribution over K.

Lemma 10.1. *Let $1 \le k \le \log N - \log\log N - 8$. With probability $1 - o(1/N)$ over the drawing of μ from D_k, every element in the support of μ has mass in the range $\left(1 \pm \frac{1}{9}\right) \cdot \frac{2^k}{N}$.*

Proof. By Chernoff's bound, $\Pr\left[|K| \in \left(1 \pm \frac{1}{10}\right) 2^{-k} N\right] \ge 1 - 2e^{-\frac{1}{300} \cdot 2^{-k} N} = 1 - o(1/N)$. If this happens, then every element in the support of μ has probability mass $\frac{1}{|K|} = \frac{1}{1 \pm 1/10} \cdot \frac{2^k}{N} = \left(1 \pm \frac{1}{9}\right) \frac{2^k}{N}$.

□

Let $k_{\min} = \lfloor \frac{1}{3} \log N \rfloor$ and $k_{\max} = \lceil \frac{2}{3} \log N \rceil$. We use $k_{\min}$ and $k_{\max}$ to define the "composed" distribution over inputs: D draws k uniformly in the range $\{k_{\min}, \ldots, k_{\max}\}$ and then returns the pair (k, μ) where μ is an input distribution drawn from D_k.

Observation 10.2. *Let $(k, \mu) \sim D$. A conditional-sampling algorithm that draws $x \sim \mu$ and estimates $\mu(x)$ within $1 \pm \frac{1}{9}$-factor with probability at least p can correctly obtain k with probability at least $p - o(1)$ for sufficiently large N.*

Proof. If N is sufficiently large, then $k_{\max} \leq \log N - \log\log N - 8$. With probability $1 - o(1/N) = 1 - o(1)$, the mass of individual elements in μ is in the range $\left(1 \pm \frac{1}{9}\right) \frac{2^k}{N}$. Hence, with probability $1 - p - o(1)$, the algorithm obtains an estimation $\hat{p} \in \left(1 \pm \frac{1}{9}\right)\left(1 \pm \frac{1}{9}\right) \frac{2^k}{N} = \left(1 \pm \frac{1}{4}\right) \frac{2^k}{N}$.

In this case, the algorithm can retrieve k using $\hat{k} = \text{round}(\log(N/\hat{p}))$ since $\log(N/\hat{p}) = \log 2^k + \log\left(1 \pm \frac{1}{4}\right) = k \pm 0.42$. Hence, the rounding of $\log(N/\hat{p})$ to the nearest integer results in k. □

By Yao's principle [Yao77], every probabilistic algorithm can be seen as a distribution over deterministic algorithms, and a lower bound against all deterministic algorithms using a single distribution over inputs translates to a lower bound against all probabilistic algorithms. A deterministic querying algorithm can be characterized as a decision tree, where every internal node (including the root) holds a query, and every edge corresponds to a possible outcome.

Our interim models and additional notations

For our lower bound we investigate the relationship of three models. The first is not related to distributions at all, and is just a model for the plain binary search task for a value k that is drawn uniformly from the set $I = \{k_{\min}, \ldots, k_{\max}\}$. The second model, a uniform "conditional" sampling model, uses the responses to the comparison queries with k to provide additional simulated responses to a conditional sampling oracle, although at this point no actual distribution is used.

The third model, a "leaking" conditional sampling model, draws a distribution μ over $\Omega = \{1, \ldots, N\}$ (whose size is $2^{\Theta(|I|)}$) using D_k, and complements the comparison queries with actual conditional samples. In particular, the expressiveness of algorithms under this last model is at least as strong as the expressiveness of algorithms that only take conditional samples from μ. By Observation 10.2, an estimation of an element drawn from μ with high probability reveals the value of k. To finalize, we show that the behavior of an algorithm under the leaking model is very close to its behavior under the uniform model (which is fully simulated from just the comparison queries), and hence a working estimation algorithm provides an algorithm that with high probability solves the binary search problem for k. This implies the lower bound of $\Omega(\log |I|) = \Omega(\log\log N)$.

Definition 10.3 (The n-range binary search model)**.** For a parameter n and a fixed well-ordered set I of size n, the input of the algorithm is some $k \in I$, which is inaccessible. In every step, the algorithm chooses some s and queries the predicate "$s \leq k$". In the end, the algorithm chooses $k' \in I$. The algorithm *succeeds* if $k' = k$.

The following observation is well known, and easy to prove by considering the possible number of leaves of a bounded depth binary tree.

Observation 10.4. *Every algorithm in the n-range binary search model whose success probability is strictly greater than* $1/2$*, over a uniformly random choice of* $k \in I$*, must make* $q > \log n - 1$ *queries.*

We define a common framework for the two conditional sampling models that we define shortly: the uniform conditional model and the leaking conditional model.

Definition 10.5 (Common framework for conditional sampling models)**.** For a parameter N, the input of the algorithm is $k \in \{k_{\min}, \ldots, k_{\max}\}$, which is inaccessible, and a distribution μ over $\{1, \ldots, N\}$. In every step, the algorithm chooses a non-empty subset $B \subseteq \{1, \ldots, N\}$ and receives a pair (b, y) where $b \in \{\text{“}\leq\text{”}, \text{“}>\text{”}\}$ and $y \in B \cup \{\text{“err”}\}$. The behavior of (b, y) given k, μ, B and the execution path so-far is determined by the specific model.

Note that the upper-bound algorithm for the estimation task in this paper interfaces with a model that has no b component in the answers to its queries. The leaking model that we define below provides conditional query access to a specific drawn distribution along with some additional information given through the additional component. Also, the leaking model does not require a logical guarantee that B has strictly positive probability mass in the input distribution μ (a guarantee that our upper-bound algorithm satisfies). The option for an “err” answer for the y component is used by the leaking model to also handle zero probability condition sets.

Every algorithm in the common framework defined above can be described as a decision tree whose internal nodes (including the root) hold the condition set B and whose edges are labeled with the possible outcomes (b, y).

Definition 10.6 (Characterization of a decision node)**.** A decision node u of a decision tree A is characterized by:

- ℓ_u (short form: ℓ), the node-distance of u from the root. ($\ell = 1$ for the root).
- A sequence $(b_{u,1}, y_{u,1}), \ldots, (b_{u,\ell_u-1}, y_{u,\ell_u-1})$ (short form: $(b_1, y_1), \ldots, (b_{\ell-1}, y_{\ell-1})$) describing the path from the root to u.
- A non-empty condition set B_u (short form: B).

Definition 10.7 (Set of already-seen elements, Y_{old})**.** Let u be a decision node characterized by $(\ell_u, (b_{u,i}, y_{u,i})_{1 \leq i \leq \ell_u - 1}, B_u)$. The *set of already-seen elements* is $Y_{\text{old}}(u) \stackrel{\text{def}}{=} \{y_{u,1}, \ldots, y_{u,\ell_u-1}\} \setminus \{\text{“err”}\}$ (short form: Y_{old}).

Definition 10.8 (Set of ruled-out elements, Y_{out})**.** Let u be a decision node characterized by $(\ell_u, (b_{u,i}, y_{u,i})_{1 \leq i \leq \ell_u - 1}, B_u)$. The *set of ruled-out elements* is $Y_{\text{out}}(u) \stackrel{\text{def}}{=} \bigcup_{1 \leq j \leq \ell_u - 1 : y_{u,j} = \text{“err”}} B_{u_j}$ (short form: Y_{out}).

The following defines the set of elements for which a node query can provide new information.

Definition 10.9 (Net condition set, net condition size)**.** Let u be a decision node characterized by $(\ell_u, (b_{u,i}, y_{u,i})_{1 \leq i \leq \ell_u - 1}, B_u)$. The *net condition set* of u is $B'_u \stackrel{\text{def}}{=} B \setminus (Y_{\text{old}}(u) \cup Y_{\text{out}}(u))$ (short form: B'). The *net condition size* of u is $s_u = |B'_u|$ (short form: s).

Based on the above notations we define the uniform conditional model and the leaking conditional sampling model.

Definition 10.10 (The uniform conditional model)**.** This model is based on the framework for conditional sampling models. Let u be a decision node characterized by $(\ell, (b_i, y_i)_{1 \le i \le \ell-1}, B)$. The behavior of (b, y), which is the outcome of the query to be made by u, is defined as follows:

- If $s_u \le 2^k$, then the outcome of the query is ("$\le$", y) for y uniformly drawn from $B_u \cap Y_{\text{old}}(u)$ if it is not empty, and otherwise it is ("$\le$","err").
- If $s_u > 2^k$, then the outcome of the query is (">", y) for y uniformly drawn from $B_u \setminus Y_{\text{out}}(u)$ if it is not empty, and otherwise it is (">","err").

In the end, the algorithm chooses $k' \in \{1, \ldots, n\}$. The algorithm *succeeds* if $k' = k$.

Note that this is essentially a simulation model, as it gives its query answers without taking any samples from μ. The following lemma indeed connects it to the "pure binary search" model.

Lemma 10.11. *Every q-query algorithm in the uniform conditional model is behaviorally identical to a q-query probabilistic algorithm in the n-range binary search model, where $n = k_{\max} - k_{\min} + 1$. Specifically, such an algorithm is equivalent to a distribution over (deterministic) binary decision trees that only use queries on whether $s \le k$ for some s (i.e., use only the b components of the answers provided by the uniform conditional model).*

Proof. Consider a decision tree in the common conditional framework, in which every edge is labeled by a pair (b, y) for some $b \in \{\text{"}\le\text{"}, \text{">"}\}$ and $y \in B_u$. For every node u in the decision tree, consider the possible distributions over its children under the uniform conditional model conditioned on the value of b and on the algorithm reaching this node.

If $b =$ "$\le$", this means in particular that $s_u \le 2^k$. If $B_u \cap Y_{\text{old}}(u) = \emptyset$ then the edge labeled by ("$\le$","err") is taken with probability 1, and otherwise the outgoing edge is chosen uniformly from the set of edges whose labels are in the set $\{(\text{"}\le\text{"}, y) : y \in B_u \cap Y_{\text{old}}(u)\}$.

If $b =$ ">", this means in particular that $s_u > 2^k$. If $B_u \setminus Y_{\text{out}}(u) = \emptyset$ then the edge labeled by (">","err") is taken with probability 1, and otherwise the outgoing edge is chosen uniformly from the set of edges whose labels are in the set $\{(\text{">"}, y) : y \in B_u \setminus Y_{\text{out}}(u)\}$.

The common theme here is that the identity of u and the value of b by themselves determine a set of outgoing edges, from which one is uniformly picked, without any dependency on the other parameters of the input. This means that a run of this q-query decision tree can be alternatively described by the following process:

- For every node u of the tree, one edge is picked uniformly from the set of the relevant outgoing edges with $b =$ "$\le$", and one edge is picked uniformly from the set of the relevant outgoing edges with $b =$ ">".
- Then, all edges in the tree that were not picked in the previous step are removed, after which all nodes with no remaining path to the root are removed as well. In the remaining tree, the edge labels are trimmed to include only the b component, which refers to a comparison of some $k' = \lceil \log s \rceil$ with k. The result of this process is a deterministic binary decision tree that can be run under the binary search model.
- Finally, the resulting tree is run with respect to the input k. □

Definition 10.12 (The leaking conditional sampling model)**.** This model is based on the framework for conditional sampling models. Let u be a decision node characterized by $(\ell, (b_i, y_i)_{1 \le i \le \ell-1}, B)$. The behavior of (b, y), which is the outcome of the query to be made by u, is defined as follows:

- b is "$\le$" if $s_u \le 2^k$ and "$>$" if $s_u > 2^k$.
- If $\mu(B) > 0$ then y is drawn from μ when conditioned on $y \in B$, and otherwise $y =$ "err".

Clearly, the leaking conditional sampling model is not weaker than any reasonable variant of the classic conditional sampling model, and hence it is suitable for lower bound statements. All such models behave the same when $\mu(B) > 0$, but the fallback behavior when $\mu(B) = 0$ is explicitly defined by every model. The return of an error message in the case where $\mu(B) = 0$ provides the most information among common fallback behaviors (uniform choice, minimum, etc.), which makes it the best choice for lower bound statements.

In the following, A is a decision tree of height q corresponding to a deterministic algorithm in the common framework of conditional sampling models. Our random variables are:

- $u_1, \ldots, u_{q+1}$ – the nodes on the execution path.
- u – an alias for the leaf u_{q+1}.
- $(b_1, y_1), \ldots, (b_q, y_q)$ – the outcomes of the queries. In other words, for every $1 \le i \le q$, (b_i, y_i) is the label on the edge from u_i to u_{i+1}. Note that $y_1, \ldots, y_q$ are generally random even that the analyzed algorithm is deterministic.
- K – the support of the input distribution that is drawn according to D_k (following the random choice of k). It plays a role only in the analysis of the leaking model.

In the following, we refer to the set Λ that refers to the combination of the choice of k, the support K (relevant for the leaking model), and the outcome (the leaf reached) of a run of the given deterministic algorithm. The two distributions that we analyze over Λ are $\mathcal{U}$, the one resulting from the uniform model, and $\mathcal{L}$, the one resulting from the leaking model.

In particular, note the following well-known common bound.

Lemma 10.13. *Let $\mathcal{U}$ and $\mathcal{L}$ be two distributions over Λ. If there exists an event $E \subseteq \Lambda$ for which $\mathcal{L}(x) > (1-\varepsilon)\mathcal{U}(x)$ for every $x \in E$, then $d_{\mathrm{TV}}(\mathcal{U}, \mathcal{L}) \le \varepsilon + \Pr_{\mathcal{U}}[\neg E]$.*

We also use some shorthand. In particular, a set of leaves (or more generally, nodes) of the analyzed algorithm (given as a decision tree) is identified with the event of reaching a node from this set. Also, the notation $\Pr_{\mathcal{U}}[E|k]$ refers to the probability of an event E (usually given by a set of leaves) when conditioned on the event of the specific k being drawn from the range $\{k_{\min}, \ldots, k_{\max}\}$.

Analysis for the uniform model

Definition 10.14 (The set of improbable elements, A_{small})**.** Let u be a decision node characterized by $(\ell, (b_i, y_i)_{1 \le i \le \ell-1}, B)$. Let $u_1, \ldots, u_\ell$ be its path from the root (where u_1 is the root and $u_\ell = u$). The *set of small elements* with respect to u and some k is $A_{\mathrm{small}}(u, k) = \bigcup_{1 \le i \le \ell : s_{u_i} \le 2^k / \log N} B'_{u_i}$ (short form: A_{small}).

Definition 10.15 (The good events, $\mathcal{G}_k$, $\mathcal{G}_k^{(1)}$, $\mathcal{G}_k^{(2)}$, $\mathcal{G}_k^{(3)}$)**.** Let u be a leaf. Let $u_1, \ldots, u_q, u_{q+1}$ be its path from the root (where u_1 is the root and $u_{q+1} = u$). We define the following good events about u:

- $\mathcal{G}_k^{(1)}$: for every $1 \leq i \leq q$, $s_{u_i} \notin \left(\frac{1}{\log N}, 8\log^3 N\right) \cdot 2^k$.
- $\mathcal{G}_k^{(2)}$: for every $1 \leq i \leq q$, $y_i \notin A_{\text{small}}(u_i, k)$.
- $\mathcal{G}_k^{(3)}$: for every $1 \leq i \leq q$, if $s_{u_i} \geq 2^k \cdot 8\log^3 N$, then $y_i \in B_{u_i} \setminus Y_{\text{old}}(u_i)$.
- $\mathcal{G}_k$: the intersection $\mathcal{G}_k^{(1)} \wedge \mathcal{G}_k^{(2)} \wedge \mathcal{G}_k^{(3)}$.

Lemma 10.16. *Let A be a decision tree representing a deterministic algorithm in the common framework for conditional sampling algorithms that makes $q \leq \log\log N - 2\log\log\log N$ queries. There exists a set $G \subseteq \{k_{\min}, \ldots, k_{\max}\}$ of size at least $\left(1 - \frac{1}{\log\log\log N}\right) n$, for $n = k_{\max} - k_{\min} + 1$, such that for every $k \in G$, considering the (random) leaf u that the execution path reaches in the uniform model, $\Pr_{\mathcal{U}}\left[u \in \mathcal{G}_k^{(1)} \middle| k\right] \geq 1 - \frac{\log\log\log N}{15\log\log N}$.*

Proof. As observed in Lemma 10.11, a decision tree in the uniform model behaves as a distribution over deterministic binary search trees, where every node u in such a tree corresponds to a comparison of k with some s_u, receiving an answer $b \in \{\text{“}\leq\text{”}, \text{“}>\text{”}\}$.

A binary decision tree of edge-height $q \leq \log\log N - 2\log\log\log N$ has exactly $2^q - 1 < \frac{\log N}{(\log\log N)^2}$ decision nodes. For every decision node u_i, there are at most $\lceil \log\log N + \log 8\log^3 N \rceil \leq 5\log\log N$ “bad” choices of k for which $s_{u_i} \in \left(\frac{1}{\log N}, 8\log^3 N\right) \cdot 2^k$. Considering the whole tree, there are at most $\frac{5\log N}{\log\log N}$ such bad choices.

For every k, let p_k be the probability to choose a binary tree that k is bad with respect to it. By linearity of expectation, $\sum_{k=k_{\min}}^{k_{\max}} p_k$ is the expected number of ks that are bad with respect to the drawn binary tree, which is bounded by $\frac{5\log N}{\log\log N}$.

For a uniform drawing of k between $k_{\min}$ and $k_{\max}$, $\mathrm{E}_k[p_k] \leq \frac{5\log N/\log\log N}{n} \leq \frac{15\log N/\log\log N}{\log N} = \frac{1}{15\log\log N}$. The last transition is correct since $n = k_{\max} - k_{\min} + 1 \geq \frac{1}{3}\log N$.

By Markov's inequality, there are at most $\frac{n}{\log\log\log N}$ choices of k for which $p_k \geq \frac{\log\log\log N}{15\log\log N}$. □

Lemma 10.17. *Let A be a decision tree representing a deterministic algorithm in the common framework for conditional sampling algorithms that makes $q \leq \log\log N - 2\log\log\log N$ queries. There exists a set $G \subseteq \{k_{\min}, \ldots, k_{\max}\}$ of size at least $\left(1 - \frac{1}{\log\log\log N}\right) n$, for $n = k_{\max} - k_{\min} + 1$, such that for every $k \in G$, considering the (random) leaf u that the execution path reaches in the uniform model, $\Pr_{\mathcal{U}}[u \in \mathcal{G}_k | k] \geq 1 - \frac{\log\log\log N}{10\log\log N}$.*

Proof. We use the set G provided by Lemma 10.16. For $k \in G$, with probability at least $1 - \frac{\log\log\log N}{15\log\log N}$, we reach a leaf u (on the $q+1$st level) for which, for every $1 \leq i \leq q$, $s_{u_i} \notin \left(\frac{1}{\log N}, 8\log^3 N\right) \cdot 2^k$. Also, for every $1 \leq i \leq q$, the probability that $y_i \in A_{\text{small}}(u_i, k)$ is bounded by:

- Zero if $s_{u_i} \leq 2^k/\log N$, by definition of the model.
- $\frac{|A_{\text{small}}(u_i,k)|}{\left|B_{u_i} \cap Y_{\text{old}}(i)\right|+s_{u_i}} \leq \frac{q}{8\log^4 N}$ if $s_{u_i} \geq 2^k \cdot 8\log^3 N$.

Also, if $s_{u_i} \geq 2^k \cdot 8\log^3 N$, then the probability to obtain an already-seen y_i is bounded by $\frac{\left|B_{u_i} \cap Y_{\text{old}}(i)\right|}{\left|B_{u_i} \cap Y_{\text{old}}(i)\right|+s_{u_i}} \leq \frac{q}{q+2^k \cdot 8\log^3 N} \leq \frac{1}{\log N}$.

By a union bound, the probability of u to be good is at least $1 - \left(\frac{\log\log\log N}{15\log\log N} + \frac{q^2}{8\log^4 N} + \frac{q}{\log N}\right) \geq 1 - \frac{\log\log\log N}{10\log\log N}$ for N large enough. □

Analysis for the leaking model

Lemma 10.18. *Let k be fixed and let $u \in \mathcal{G}_k$ be a leaf (node of depth $q+1$) whose path from the root is $u_1, \ldots, u_q, u_{q+1}$. For every $1 \leq i \leq q$, $A_{\text{small}}(u_i, k)$ is disjoint from $Y_{\text{old}}(u)$.*

Proof. Let $(b_1, y_1), \ldots, (b_q, y_q)$ be the outcome sequence. By definition, $Y_{\text{old}} = \{y_1, \ldots, y_q\} \setminus \{\text{“err”}\}$.

For $i \leq j$, the definition of $\mathcal{G}_k$ eliminates the possibility that $y_i \in A_{\text{small}}(u_j, k)$.

For $i > j$, if $y_i \in A_{\text{small}}(u_j, k)$, then due to monotonicity, $y_i \in A_{\text{small}}(u_i, k)$ as well, a contradiction to the definition of $\mathcal{G}_k$. □

Observation 10.19. *For a given k and a leaf $u \in \mathcal{G}_k$ whose path from the root is $u_1, \ldots, u_q, u_{q+1}$,*

$$\Pr_{\mathcal{U}}[u|k] = \prod_{i=1}^{q} \begin{cases} 1 & s_{u_i} \leq 2^k/\log N, \quad B_{u_i} \cap Y_{\text{old}} = \emptyset \\ \frac{1}{|B_{u_i} \cap Y_{\text{old}}|} & s_{u_i} \leq 2^k/\log N, \quad B_{u_i} \cap Y_{\text{old}} \neq \emptyset \\ \frac{1}{|B_{u_i} \setminus Y_{\text{out}}|} & s_{u_i} \geq 2^k \cdot 8\log^3 N \end{cases}$$

Proof. If $s_{u_i} \leq 2^k/\log N$, then by definition, y_i is uniformly drawn from $B_{u_i} \cap Y_{\text{old}}$, unless this intersection is empty, and in this case y_i is “err” with probability 1.

If $s_{u_i} \geq 2^k \cdot 8\log^3 N$, then since $Y_{\text{out}}(u_i) \subseteq A_{\text{small}}(u_i, k)$, $|B_{u_i} \setminus Y_{\text{out}}| \geq \left(1 - \frac{1}{8\log N}\right)|B_{u_i}| > 0$. Hence, by definition y_i is uniformly drawn from $B_{u_i} \setminus Y_{\text{out}} \neq \emptyset$. □

Lemma 10.20. *For every $k \in \{k_{\min}, \ldots, k_{\max}\}$ and leaf $u \in \mathcal{G}_k$ (which is good with respect to k), $\Pr_{\mathcal{L}}[u|k] \geq \left(1 - \frac{3q}{\log N}\right)\Pr_{\mathcal{U}}[u|k]$.*

Proof. Let $u_1, \ldots, u_q, u_{q+1}$ be the path from the root u_1 to $u_{q+1} = u$, and let $(b_1, y_1), \ldots, (b_q, y_q)$ be the sequence of answers in this path. Let t be the number of indexes for which $s_{u_i} \geq 2^k \cdot 8\log^3 N$. Recall that since $u \in \mathcal{G}_k$, t is also the number of unique non-error values in $y_1, \ldots, y_q$.

We define the following good events corresponding to an input distribution $\mu \sim D_k$ (which is fully determined by the set K):

- G_1: $\{y_1, \ldots, y_q\} \setminus \{\text{“err”}\} \subseteq K$.

- G_2: $A_{\text{small}}(u_q, k)$ is disjoint from K (in the rest of this proof, non-indexed instances of A_{small} refer to this set).
- G_3: for every $1 \le i \le q$, if $s_{u_i} \ge 2^k \cdot 8\log^3 N$, then $|B_{u_i} \cap K| \le \left(1 + \frac{1}{\log N}\right) 2^{-k} |B_{u_i} \setminus Y_{\text{out}}|$.

By the chain rule,

$$\Pr_{\mathcal{L}}[u|k] \ge \Pr_{\mathcal{L}}[u \wedge G_1 \wedge G_2 \wedge G_3|k] = \Pr_{\mathcal{L}}[G_1|k] \cdot \Pr_{\mathcal{L}}[G_2 \wedge G_3|k, G_1] \cdot \Pr_{\mathcal{L}}[u|k, G_1, G_2, G_3]$$

Clearly, $\Pr_{\mathcal{L}}[G_1|k] = (2^{-k})^t$.

For bounding the probability for G_2, we note that by Markov's inequality, the probability that A_{small} is disjoint from K is at least $1 - 2^{-k} \cdot |A_{\text{small}}| \ge 1 - \frac{q}{\log N}$. This holds also when conditioned on G_1 since $A_{\text{small}} \cap \{y_1, \ldots, y_q\} = \emptyset$ (Lemma 10.18).

For nodes with $s_{u_i} \ge 2^k \cdot 8\log^3 N$, since $Y_{\text{out}}(u_i) \subseteq A_{\text{small}}(u_i, k)$, we obtain that $|B_{u_i} \setminus Y_{\text{out}}| \ge \left(1 - \frac{1}{8\log N}\right) |B_{u_i}|$ (and by definition of $\mathcal{G}_k$ we cannot get $y_i =$ "err" for such nodes).

By Chernoff's inequality and a union bound,

$$\begin{aligned}
\Pr_{\mathcal{L}}[\neg G_3|k, G_1] &\le q \cdot \Pr\left[\text{Bin}\left(|B_u| - (t + |A_{\text{small}}|), 2^{-k}\right) + t > \left(1 + \frac{1}{\log N}\right) 2^{-k} |B_u \setminus Y_{\text{out}}|\right] \\
(*) &\le q \cdot \Pr\left[\text{Bin}\left(|B_u|, 2^{-k}\right) + t > \left(1 + \frac{1}{\log N}\right) 2^{-k} |B_u \setminus Y_{\text{out}}|\right] \\
(**) &\le q \cdot \Pr\left[\text{Bin}\left(|B_u|, 2^{-k}\right) + t > \left(1 + \frac{3}{4\log N}\right) 2^{-k} |B_u|\right] \\
(***) &\le q \cdot \Pr\left[\text{Bin}\left(|B_u|, 2^{-k}\right) > \left(1 + \frac{1}{2\log N}\right) 2^{-k} |B_u|\right] \\
&\le q \cdot e^{-\frac{1}{12\log^2 N} \cdot 8\log^3 N} < \frac{q}{\log N}
\end{aligned}$$

$(*)$: since the random variable $\text{Bin}\left(|B_u|, 2^{-k}\right)$ has "more opportunities" to be bigger than a given bound $\text{Bin}\left(|B_u| - (t + |A_{\text{small}}|), 2^{-k}\right)$.
$(**)$: since $|B_u \setminus Y_{\text{out}}| \ge \left(1 - \frac{1}{8\log N}\right) |B_u|$ and $\left(1 + \frac{1}{\log N}\right)\left(1 - \frac{1}{8\log N}\right) \ge \left(1 + \frac{3}{4\log N}\right)$ for large enough N.
$(***)$: since $t \le \log\log N \le \frac{1}{8\log N} \cdot 2^{-k} |B_u|$ and $\left(1 + \frac{3}{4\log N}\right)\left(1 - \frac{1}{3\log N}\right) \ge \left(1 + \frac{1}{2\log N}\right)$ for large enough N.

By the union bound, $\Pr_{\mathcal{L}}[G_2 \wedge G_3|k, G_1] \ge 1 - \frac{2q}{\log N}$.

Finally,

$$\begin{aligned}
\Pr_{\mathcal{L}}[u|k,G_1,G_2,G_3] &= \prod_{i=1}^{q}\Pr_{\mathcal{L}}[y_i|k,G_1,G_2,G_3] \\
&\geq \prod_{i=1}^{q}\begin{cases} 1 & s_{u_i}\leq 2^k/\log N, \quad B_{u_i}\cap Y_{\text{old}}=\emptyset \\ \frac{1}{|B_{u_i}\cap Y_{\text{old}}|} & s_{u_i}\leq 2^k/\log N, \quad B_{u_i}\cap Y_{\text{old}}\neq\emptyset \\ \frac{1}{\left(1+\frac{1}{\log N}\right)2^{-k}\,|B_{u_i}\setminus Y_{\text{out}}|} & s_{u_i}\geq 2^k\cdot 8\log^3 N \end{cases} \\
\text{[Observation 10.19]} &= \frac{2^{kt}}{\left(1+\frac{1}{\log N}\right)^t}\Pr_{\mathcal{U}}[u|k] \geq 2^{kt}\left(1-\frac{t}{\log N}\right)\Pr_{\mathcal{U}}[u|k] \geq 2^{kt}\left(1-\frac{q}{\log N}\right)\Pr_{\mathcal{U}}[u|k]
\end{aligned}$$

Combined,

$$\Pr_{\mathcal{L}}[u|k] \geq 2^{-kt}\cdot\left(1-\frac{2q}{\log N}\right)\cdot 2^{kt}\left(1-\frac{q}{\log N}\right)\Pr_{\mathcal{U}}[u|k] \geq \left(1-\frac{3q}{\log N}\right)\Pr_{\mathcal{U}}[u|k] \qquad \square$$

Lemma 10.21. *Consider a deterministic algorithm making $q \leq \log\log N - 2\log\log\log N$ queries in the common framework for conditional sampling algorithms. For at least $\left(1-\frac{1}{\log\log\log N}\right)n$ choices of k, where $n = k_{\max}-k_{\min}+1$, the distance between the distributions over execution paths of the algorithm, when executed on either the leaking model or on the uniform model, is bounded by $\frac{1}{\log\log N}$ when considering $\mu \sim D_k$.*

Proof. By Lemma 10.17, $\Pr_{\mathcal{U}}[\mathcal{G}_k|k] \geq 1-\frac{\log\log\log N}{10\log\log N}$ for $\left(1-\frac{1}{\log\log\log N}\right)n$ choices of k. By Lemma 10.20, if $u\in\mathcal{G}_k$, then $\Pr_{\mathcal{L}}[u|k] \geq \left(1-\frac{3q}{\log N}\right)\Pr_{\mathcal{U}}[u|k]$. Hence, by Lemma 10.13, the total variation distance between the distribution of the respective runs is bounded by $\frac{3q}{\log N}+\frac{\log\log\log N}{10\log\log N} \leq \frac{3\log\log N}{\log N}+\frac{\log\log\log N}{10\log\log N} \leq \frac{1}{\log\log N}$ for these choices of k. $\square$

We are now ready to prove our lower bound. Note that in particular it applies to algorithms solving the $(\frac{1}{9},\frac{1}{9})$-estimation task.

Theorem 10.22. *Every conditional sampling algorithm that, with probability at least p for a fixed $p>\frac{1}{2}$, can estimate an element drawn from μ within a factor of $1\pm\frac{1}{9}$, must draw $\Omega(\log\log N)$ conditional samples.*

Proof. By Observation 10.2, such an algorithm can compute k with probability at least $p-o(1)$ when its input (k,μ) is drawn from D (that is, k is uniformly drawn from $\{k_{\min},\ldots,k_{\max}\}$ and then μ is drawn from D_k). By Lemma 10.21, unless $q > \log\log N - 2\log\log\log N$, the chosen k is with probability $1-o(1)$ such that the same algorithm, when executed in the uniform conditional sampling model, has its distribution over runs $o(1)$-close to the one produced by the leaking model. Hence the algorithm can compute k with probability $p-o(1)>\frac{1}{2}$ under the uniform conditional model as well. By Lemma 10.11 and Observation 10.4, $\log\log N - 2\log\log\log N$ queries do not suffice for computing k in this model with success probability greater than $\frac{1}{2}$, and hence the algorithm must make strictly more than $\log\log N - 2\log\log\log N = \Omega(\log\log N)$ queries. $\square$

10.2 Lower bound estimation task under weaker models

Recall Theorem 1.3:

Theorem 1.3 (Almost-tight upper bound for equivalence testing)**.** *Let μ, τ be two distributions over $\Omega = \{1, \ldots, N\}$ and $\varepsilon > 0$. There exists an algorithm for distinguishing between the case where $\mu = \tau$ and the case where $d_{\mathrm{TV}}(\mu, \tau) > \varepsilon$, using $O((\log\log N/\varepsilon + 1/\varepsilon^5) \cdot \mathrm{poly}(\log \varepsilon^{-1}))$ conditional samples.*

The proof of Theorem 1.3 assumes that the (c, ε)-peek oracle can be simulated using T conditional samples in expectation where $c = \varepsilon$, and obtains the upper bound of $T \cdot \tilde{O}(1/\varepsilon)$ conditional samples for an ε-test of equivalence.

We now review two well-investigated distribution testing models that are more restrictive than the full one in which our estimator operates. For each of them we use a known lower bound on the equivalence testing task along with the above observation to provide a corresponding lower bound for the (c, ε)-estimation task.

Definition 10.23 (The subcube conditional oracle)**.** A set $A \subseteq \{0,1\}^n$ is a *subcube* if there exist $A_1, \ldots, A_n \subseteq \{0,1\}$ for which $A = A_1 \times \cdots \times A_n$. The *subcube conditional oracle* is the restriction of the conditional oracle to answer only subcube condition sets.

For product distributions, [CDKS17, Theorem 43] shows a lower bound of $\tilde{\Omega}(n)$ samples on $(\varepsilon/2, \varepsilon)$-tolerant equivalence testing of product distributions over $\{0,1\}^n$ (the size of the sample set is $N = 2^n$). [JHW18] improves the ε-dependency of the lower bound. As observed in [AFL24a], subcube conditional sampling has no additional power over unconditional sampling when the input distributions are guaranteed to be product distributions. This implies the following bound.

Corollary 10.24. *Every algorithm that solves the (c, ε)-estimation task using subcube sampling must make at least $\tilde{\Omega}(\log N)$ subcube queries in expectation for every sufficiently small $\varepsilon > 0$ and $c > 0$.*

Definition 10.25 (The interval conditional oracle)**.** A set $A \subseteq \{1, \ldots, N\}$ is an *interval* if there exist $1 \le a \le b \le N$ for which $A = \{i : a \le i \le b\}$. The *interval conditional oracle* is the restriction of the conditional oracle to answer only interval condition sets.

For interval conditions, [CRS15] show a lower bound of $\tilde{\Omega}(\log N)$ interval queries for uniformity testing, which is a special case of equivalence testing. This implies the following bound.

Corollary 10.26. *Every algorithm that solves the (c, ε)-estimation task using interval conditions must make at least $\tilde{\Omega}(\log N)$ subcube queries in expectation for every sufficiently small $\varepsilon > 0$ and $c > 0$.*

Note that the polylogarithmic algorithm from [CFGM16] in particular applies to both the subcube conditional model and the interval conditional model. The above corollaries in particular imply a limit on the possibility for its improvement.

10.3 Lower bound for testing label-invariant properties

We show that there exist a label-invariant property that has an $\Omega(\log N/\varepsilon)$ lower bound for ε-testing using the conditional model for every sufficiently small $\varepsilon > 0$. We show that some k-bit

string property is linearly hard to test in an ad-hoc testing model, and encode string instances related to this property in the histogram of distributions over a domain of size $N = 2^{\Omega(\varepsilon k)}$.

Definition 10.27 (Notations)**.**

- Let X be a set. We use 2^X to denote the set of all subsets of X.
- Let I be a set of integers. For an integer k, we use $k - I$ to denote the set $\{k - i : i \in I\}$.
- Let I be a set of integers. For an integer k, we use $\neg_k I$ to denote the set $\{1, \ldots, k\} \setminus I$.

Definition 10.28 (q-uniform family)**.** A family $\mathcal{I} \subseteq 2^{\{1,\ldots,k\}}$ is *q-uniform* if, for every subset $J \subseteq \{1, \ldots, k\}$ of size q, the intersection of J with a uniformly drawn set $I \sim \mathcal{I}$ is uniformly distributed over 2^J.

Definition 10.29 (k-paired set)**.** A set $I \subseteq \{1, \ldots, k\}$ is *k-paired* if $(k + 1) - I = \neg_k I$.

Definition 10.30 (paired q-uniform family)**.** For an even k, a family $\mathcal{I} \subseteq 2^{\{1,\ldots,k\}}$ is *paired q-uniform* if every $I \in \mathcal{I}$ is a k-paired set, and for every subset $J \subseteq \{1, \ldots, k\}$ of size q which is disjoint from $(k + 1) - J$, the intersection of J with a uniformly drawn set $I \sim \mathcal{I}$ is uniformly distributed over 2^J.

Observation 10.31. *Let $\mathcal{I} \subseteq \{1, \ldots, k\}$ be a q-uniform family. The family $\mathcal{I}' = \{I \cup ((2k + 1) - (\neg_k I)) : I \in \mathcal{I}\} \subseteq 2^{\{1,\ldots,2k\}}$ is a paired q-uniform family.*

Observation 10.32. *For an even k, let $\mathcal{I} \subseteq 2^{\{1,\ldots,k\}}$ be a paired q-uniform family. For every $J \subseteq \{1, \ldots, k\}$ of size less than q, $I' \subseteq J$ for which $\Pr_{I\sim\mathcal{I}}[I \cap J = I'] > 0$ and j for which $\{j, k + 1 - j\} \cap J = \emptyset$, if we uniformly draw $I \sim \mathcal{I}$, then $\Pr_{I\sim\mathcal{I}}[j \in I | J \cap I = I'] = \frac{1}{2}$.*

Proof. Set $J' = J \setminus (\{1, \ldots, k/2\} \cap ((k+1) - J))$. In words, J' is the result of taking J and removing every $j \leq k/2$ for which $\{j, k+1-j\} \subseteq J$. Note that for a random choice over family of paired sets, the events $I \cap J = I'$ and $I \cap J' = I' \cap J'$ are identical. Also note that $J' \cup \{j\}$ and $(k+1) - (J' \cup \{j\})$ are disjoint by the assertion on j. Hence,

$$\begin{aligned}
\Pr_{I\sim\mathcal{I}}[j \in I | J \cap I = I'] &= \Pr_{I\sim\mathcal{I}}[j \in I | J' \cap I = J' \cap I'] \\
\text{[Chain rule]} &= \frac{\Pr_{I\sim\mathcal{I}}[(j \in I) \wedge (J' \cap I = J' \cap I')]}{\Pr_{I\sim\mathcal{I}}[J' \cap I = J' \cap I']} \\
&= \frac{\Pr_{I\sim\mathcal{I}}[(J' \cup \{j\}) \cap (I \cup \{j\}) = (J' \cap I') \cup \{j\}]}{\Pr_{I\sim\mathcal{I}}[J' \cap I = J' \cap I']} \\
\text{[Definition 10.30]} &= \frac{2^{-|J'\cup\{j\}|}}{2^{-|J'|}} = \frac{1}{2}
\end{aligned}$$

□

Definition 10.33 (ε-pairwise far families)**.** Two families $\mathcal{I}_1, \mathcal{I}_2 \subseteq 2^{\{1,\ldots,k\}}$ are *ε-pairwise far* if for every $I_1 \in \mathcal{I}_1$ and $I_2 \in \mathcal{I}_2$, $|I_1 \Delta I_2| > \varepsilon k$.

Lemma 10.34. *Let $I_1, I_2 \subseteq \{1, \ldots, k\}$ be ε-far subsets. Let $J_1 = I_1 \cup ((2k + 1) - \neg_k I_1)$ and $J_2 = I_2 \cup ((2k + 1) - \neg_k I_2)$. In this setting, J_1 and J_2 are ε-far as well.*

Proof.

$$\begin{aligned}
|J_1 \Delta J_2| &= |(I_1 \cup ((2k+1) - (\neg_k I_1))) \,\Delta\, (I_2 \cup ((2k+1) - (\neg_k I_2)))| \\
&= \underbrace{|I_1 \Delta I_2|}_{\text{in } \{1,\ldots,k\}} + \underbrace{|((2k+1) - (\neg_k I_1)) \Delta ((2k+1) - (\neg_k I_2))|}_{\text{in } \{k+1,\ldots,2k\}} \\
&= |I_1 \Delta I_2| + |(\neg_k I_1) \Delta (\neg_k I_2)| \\
&= 2\,|I_1 \Delta I_2| > 2 \cdot \varepsilon k = \varepsilon \cdot (2k)
\end{aligned}$$

□

Definition 10.35 (Weighted sampling oracle)**.** Let $I \subseteq \{1,\ldots,k\}$ be a subset. The *weighted sampling* oracle for I gets a weight function $w : \{1,\ldots,k\} \to [0,1]$ as its input, and its output is an index $i \in \{1,\ldots,k\}$ and a bit $b \in \{0,1\}$ distributed as follows:

- If $\sum_{i=1}^{k} w(i) > 0$, then the probability to draw the index i is $\frac{w(i)}{\sum_{i=1}^{k} w(i)}$. The oracle returns $(i,1)$ if $i \in I$ and $(i,0)$ if $i \notin I$.
- If $\sum_{i=1}^{k} w(i) = 0$, then the oracle indicates an error.

Lemma 10.36. *Let $\mathcal{I}$ be a paired q-uniform family of subsets of $\{1,\ldots,k\}$. A sequence of q weighted sampling oracle calls with inputs $w_1,\ldots,w_q$ to a uniformly chosen $I \sim \mathcal{I}$ results in a sequence of pairs $(j_1,b_1),\ldots,(j_q,b_q)$ where the j_is are indexes and the b_is are bits. In this setting, for every $1 \le i \le q$ for which $\{j_i, (k+1)-j_i\} \cap \{j_1,\ldots,j_{i-1}\} = \emptyset$, the bit b_i is uniformly distributed, even when conditioned on the values of $(b_1,\ldots,b_{i-1})$. Additionally, for every other i, the bit b_i is a function of $j_1,\ldots,j_{i-1}$ and $b_1,\ldots,b_{i-1}$ (which does not depend on $\mathcal{I}$ or I at all). This holds even if the input w_i can be chosen based on the result of the previous $i-1$ calls.*

Proof. Note that $\mathcal{I}$ is non-empty since any paired q-uniform family must consist of at least 2^q sets.

Consider the ith call ($1 \le i \le q$) to the weighting sampling oracle. It uses internal randomness and adaptivity to choose an index j_i and query its belonging to the input set I. Let $I' = \{j_1,\ldots,j_{i-1}\} \cap I$ be the knowledge about past queried indexes. If $j_i, (k+1)-j_i \notin \{j_1,\ldots,j_{i-1}\}$ then, due to $\mathcal{I}$ being paired q-uniform and Observation 10.32, $\Pr_{I\sim\mathcal{I}}[j_i \in I | I \cap \{j_1,\ldots,j_{i-1}\} = I'] = \frac{1}{2}$.

On the other hand, if $j_i = j_{i'}$ for some $i' < i$ then $b_i = b_{i'}$ deterministically, and if $k+1-j_i = j_{i'}$ for some $i' < i$ then $b_i = 1 - b_{i'}$ deterministically, irrespective of I or $\mathcal{I}$. □

Lemma 10.37. *Let $k \ge 2$, $q \ge 2$. Let $\mathcal{I} \subseteq 2^{\{1,\ldots,k\}}$ be a paired q-uniform family. If there exists another paired q-uniform family $\mathcal{I}' \subseteq 2^{\{1,\ldots,k\}}$ which is ε-pairwise far from $\mathcal{I}$, then every ε-testing algorithm for distinguishing between $\mathcal{I}$ and being ε-far from $\mathcal{I}$ must make more than q calls to the weighted sampling oracle.*

Proof. Let $\mathcal{U} = \frac{1}{2} \cdot (\mathrm{uni}(\mathcal{I}) \times \{1\}) + \frac{1}{2} \cdot (\mathrm{uni}(\mathcal{I}') \times \{0\})$ be the distribution that uniformly chooses $b \in \{0,1\}$, and then uniformly draws a set I from $\mathcal{I}$ if $b = 1$ and from $\mathcal{I}'$ if $b = 0$. If $\mathcal{A}$ is an ε-test for the property $\mathcal{I}$, then $\Pr_{(b,I)\sim\mathcal{U}}[(\mathcal{A}(I) = \textsc{accept}) \leftrightarrow b] > \frac{1}{2}$, since it should accept every $I \in \mathcal{I}$ with probability strictly greater than $\frac{1}{2}$ and reject every $I \in \mathcal{I}'$ with probability strictly greater than $\frac{1}{2}$.

If $\mathcal{A}$ makes at most q calls to the weighted sampling oracle, then by Lemma 10.36, it receives an identical distribution of outputs regardless of whether I is drawn from $\mathcal{I}$ or from $\mathcal{I}'$. This implies that b and $\mathcal{A}(I)$ are independent, and thus $\Pr_{(b,I)\sim\mathcal{U}}[(\mathcal{A}(I) = \text{ACCEPT}) \leftrightarrow b] = \frac{1}{2}$. This is a contradiction, and hence $\mathcal{A}$ must make more than q oracle calls. □

Lemma 10.38 (Lemma 22 in [BEFLR20])**.** *A set $\{v_1 \ldots, v_{3r}\}$ of random vectors in $\{0,1\}^{4r}$ satisfies with probability $1 - o(1)$ the following two conditions:* $\text{Span}\{v_1, \ldots, v_{3r}\}$ *is a $\frac{1}{30}$-distance code, and* $\text{Span}\{v_{r+1}, \ldots, v_{3r}\}$ *is a $\frac{1}{10}$-dual distance code.*

Lemma 10.39 (Direct application of Lemma 10.38)**.** *For every sufficiently large r, there exist two families $\mathcal{J}_1$ and $\mathcal{J}_2$ of subsets of $\{1, \ldots, 4r\}$, each of them having size 2^{2r}, such that both of them are q-uniform for $q = \lceil 2r/5 \rceil$, which are $\frac{1}{30}$-pairwise far from each other. Additionally, the two families contain no members with fewer than $\lceil 2r/15 \rceil$ elements.*

Lemma 10.40. *For every sufficiently large r, there exist two families $\mathcal{I}_1$ and $\mathcal{I}_2$ of subsets of $\{1, \ldots, 8r\}$, each of them having size 2^{2r}, such that both of them are paired q-uniform for $q = \lceil 2r/5 \rceil$, which are $\frac{1}{30}$-pairwise far from each other. Additionally, the two families contain only members with exactly $4r$ elements.*

Proof. Let $\mathcal{J}_1$ and $\mathcal{J}_2$ be two q-uniform families of subsets of $\{1, \ldots, 4r\}$ that are $\frac{1}{30}$-pairwise far, whose existence is guaranteed by Lemma 10.39.

Let $\mathcal{I}_1 = \{J \cup ((8r+1) - (\neg_{4r} J)) : J \in \mathcal{J}_1\}$ and $\mathcal{I}_2 = \{J \cup ((8r+1) - (\neg_{4r} J)) : J \in \mathcal{J}_2\}$. By Observation 10.31, $\mathcal{I}_1$ and $\mathcal{I}_2$ are paired q-uniform families. Note that they are $\frac{1}{30}$-far by Lemma 10.34.

Every $I \in \mathcal{I}_1 \cup \mathcal{I}_2$ has size exactly $4r$ since there exists some $J \in \mathcal{J}_1 \cup \mathcal{J}_2$ for which $I = J \cup ((8r+1) - (\neg_{4r} J))$ and hence $|I| = |J| + |\neg_{4r} J| = |J| + (4r - |J|) = 4r$. □

We now define the distributions whose histograms can encode subsets of $\{1, \ldots, 8r\}$ as above, and for which we can perform a reduction from the conditional testing model.

Definition 10.41 (Non-empty k-partition)**.** A k-tuple $\mathcal{S} = (S_1, \ldots, S_k)$ is a *non-empty k-partition* if the sets $S_1, \ldots, S_k$ are non-empty and mutually disjoint.

If $\Omega = \bigcup_{i=1}^{k} S_i$, then we say that $\mathcal{S}$ is a non-empty k-partition *of* Ω.

Definition 10.42 (Chunk distribution)**.** Let $\mathcal{S} = (S_1, \ldots, S_k)$ be a non-empty k-partition. Let $I \subseteq \{1, \ldots, k\}$ be a non-empty set of indexes. The *chunk distribution* $\mu_{\mathcal{S},I}$ over $\bigcup_{i=1}^{k} S_i$ is defined such that for every $i \in I$, $\mu_{\mathcal{S},I}(S_i) = \frac{1}{|I|}$ and the restriction of $\mu_{\mathcal{S},I}$ to S_i is the uniform distribution over S_i. More precisely, for every $i \in I$ and $j \in S_i$, $\mu_{\mathcal{S},I}(j) = \frac{1}{|I| \cdot |S_i|}$, and $\mu_{\mathcal{S},I}(j) = 0$ for every $j \notin \bigcup_{i \in I} S_i$.

Definition 10.43 (Set of chunk distributions)**.** Let $\mathcal{S}$ be a non-empty k-partition. Let $\mathcal{I}$ be a family of non-empty subsets of $\{1, \ldots, k\}$. The *set of chunk distributions* with respect to $\mathcal{S}$ and $\mathcal{I}$ is the set $\mathcal{H}_{\mathcal{S},\mathcal{I}} = \{\mu_{\mathcal{S},I} : I \in \mathcal{I}\}$, where $\mu_{\mathcal{S},I}$ is the chunk distribution corresponding to $\mathcal{S}$ and I.

Definition 10.44 (The histogram property $\mathcal{P}_{\mathcal{S},\mathcal{I}}$)**.** Let $\mathcal{S}$ be a non-empty k-partition of a subset of a domain set Ω and $\mathcal{I}$ be a family of non-empty subsets of $\{1, \ldots, k\}$. The histogram property

with parameters $\mathcal{S}$ and $\mathcal{I}$ is the property $\mathcal{P}_{\mathcal{S},\mathcal{I}}$ of all distributions μ over Ω that are a permutation of a distribution in $\mathcal{H}_{\mathcal{S},\mathcal{I}}$.

Observation 10.45. *$\mathcal{P}_{\mathcal{S},\mathcal{I}}$ is label-invariant.*

Definition 10.46 (ρ-increasing partition)**.** A non-empty k-partition $\mathcal{S} = (S_1, \ldots, S_k)$ is ρ-increasing if for every $2 \leq i \leq k$, $|S_i| \geq \rho\,|S_{i-1}|$.

Lemma 10.47. *Let $\mathcal{S}$ be a $(1+\varepsilon)$-increasing non-empty k-partition and let $I_1, I_2 \subseteq \{1, \ldots, k\}$ be two $\frac{1}{30}$-pairwise far subsets of size exactly $\frac{1}{2}k$. In this setting, $\mu_{\mathcal{S},I_1}$ is $\frac{1}{120}\varepsilon$-far from any permutation of $\mu_{\mathcal{S},I_2}$.*

Proof. Let $\mathcal{S} = (S_1, \ldots, S_k)$, $\Omega = \bigcup_{i=1}^{k} S_i$, some $j \in I_1 \setminus I_2$, $x \in A_j$, $y \in \bigcup_{i \in I_2} S_i$ and $j' \neq j$ for which $y \in A_{j'}$. Note that:

$$\mu_{\mathcal{S},I_2}(y) = \frac{1}{|I_2|\,|S_{j'}|} = \frac{|S_j|}{|S_{j'}|} \cdot \frac{1}{|I_1|\,|S_j|} = \frac{|S_j|}{|S_{j'}|} \cdot \mu_{\mathcal{S},I_1}(x)$$

We have two cases with respect to the order of j' and j.

$$j' > j: \qquad \frac{|S_j|}{|S_{j'}|} \leq (1+\varepsilon)^{j-j'} \leq (1+\varepsilon)^{-1} \leq 1 - \frac{1}{2}\varepsilon, \qquad \mu_{\mathcal{S},I_2}(y) \leq \left(1 - \frac{1}{2}\varepsilon\right)\mu_{\mathcal{S},I_1}(x)$$

$$j' < j: \qquad \frac{|S_j|}{|S_{j'}|} \geq (1+\varepsilon)^{j-j'} \geq (1+\varepsilon)^{+1} \geq 1 + \frac{1}{2}\varepsilon, \qquad \mu_{\mathcal{S},I_2}(y) \geq \left(1 + \frac{1}{2}\varepsilon\right)\mu_{\mathcal{S},I_1}(x)$$

In both cases, $|\mu_{\mathcal{S},I_1}(x) - \mu_{\mathcal{S},I_2}(y)| \geq \frac{1}{2}\varepsilon\mu_{\mathcal{S},I_1}(x)$. This bound holds for every x in the support of $\mu_{\mathcal{S},I_1}$ and hence, for every permutation π over Ω,

$$\begin{aligned}
d_{\mathrm{TV}}(\mu_{\mathcal{S},I_1}, \pi\mu_{\mathcal{S},I_2}) &= \frac{1}{2}\sum_{i=1}^{k}\sum_{x \in S_i} |\mu_{\mathcal{S},I_1}(x) - \mu_{\mathcal{S},I_2}(\pi(x))| \\
&\geq \frac{1}{2}\sum_{i \in I_1 \setminus I_2}\sum_{x \in S_i} |\mu_{\mathcal{S},I_1}(x) - \mu_{\mathcal{S},I_2}(\pi(x))| \\
&\geq \frac{1}{2}\sum_{i \in I_1 \setminus I_2}\sum_{x \in S_i} \frac{1}{2}\varepsilon\mu_{\mathcal{S},I_1}(x) = \frac{1}{4}\varepsilon \sum_{i \in I_1 \setminus I_2} |S_i| \cdot \frac{1}{|I_1|\,|S_i|} = \frac{1}{2k}\varepsilon \cdot |I_1 \setminus I_2|
\end{aligned}$$

By a symmetric analysis we can obtain that $d_{\mathrm{TV}}(\mu_{\mathcal{S},I_2}, \pi\mu_{\mathcal{S},I_1}) \geq \frac{1}{2k}\varepsilon \cdot |I_2 \setminus I_1|$.

Since d_{TV} is invariant under both-sides permutation ($d_{\mathrm{TV}}(\pi\mu_1, \pi\mu_2) = d_{\mathrm{TV}}(\mu_1, \mu_2)$) we obtain that:

$$\begin{aligned}
\min_{\pi} d_{\mathrm{TV}}(\mu_{\mathcal{S},I_1}, \pi\mu_{\mathcal{S},I_2}) &= \min_{\pi} d_{\mathrm{TV}}(\mu_{\mathcal{S},I_2}, \pi\mu_{\mathcal{S},I_1}) \\
&\geq \frac{1}{2k}\varepsilon \max\{|I_1 \setminus I_2|, |I_2 \setminus I_1|\} \\
&\geq \frac{1}{2k}\varepsilon \cdot \frac{1}{2}\,|I_1 \Delta I_2| > \frac{1}{4k}\varepsilon \cdot \frac{1}{30}k = \frac{1}{120}\varepsilon
\end{aligned}$$

□

Algorithm 20: Procedure Initialize-COND-simulator$(k, \mathcal{S}, I)$
Input: $I \subseteq \{1, \ldots, k\}$, accessible only through the weighted sampling oracle.
1. Let $I' \leftarrow \emptyset$ be the (initially empty) partial knowledge about elements in I.
2. Let $J' \leftarrow \emptyset$ be the (initially empty) partial knowledge about elements outside I.
3. Return $(k, \mathcal{S}, I, I', J')$.

Algorithm 21: Procedure Sample-COND-simulator(obj, C)
Input: An object obj created by Initialize-COND-simulator.
Input: A condition set C.
Side effects: The algorithm may alter the I', J' components of obj.
Output: A sample $x \sim \mu_{\mathcal{S},I}$ conditioned on C, or $x \sim C$ uniformly if $\mu_{\mathcal{S},I}(C) = 0$.
1. Let $k, \mathcal{S}, I, I', J'$ be the components of obj as a 5-tuple.
2. Let $S_1, \ldots, S_k$ be the components of $\mathcal{S}$ as a k-tuple.
3. While not explicitly terminated:
   (a) Let $\hat{C} \leftarrow \{1 \leq i \leq k : (S_i \cap C \neq \emptyset) \wedge (i \notin J') \wedge ((k+1) - i \notin I')\}$.
   (b) If $\hat{C} = \emptyset$: ($\mu_{\mathcal{S},I}(C) = 0$)
      i. Draw $x \sim C$ uniformly.
      ii. Return x.
   (c) Else:
      i. Let w be the weight function for which:
         - If $i \in \hat{C}$, then $w(i) = \frac{|S_i \cap C|}{|S_i|}$.
         - If $i \notin \hat{C}$, then $w(i) = 0$.
      ii. Call the weight sampling oracle for I with w to obtain (i, b).
      iii. If $b = 1$: ($i \in I$)
         A. Add i to I'.
         B. Draw $x \sim S_i \cap C$ uniformly.
         C. Return x.
      iv. Else: ($i \notin I$)
         A. Add i to J'.

Next, we show that the conditional oracle for chunk distributions can be simulated using the weighted oracle (Algorithm 20 to initialize, 21 to simulate).

Lemma 10.48. *For every $I \subseteq \{1, \ldots, k\}$ that is a member of a paired family, the distribution of output of a sequence starting with a single call to* Initialize-COND-simulator *(Algorithm 20) with k, $\mathcal{S}$, I followed by q calls to* Sample-COND-simulator *(Algorithm 21) over the produced object with the condition sets $C_1, \ldots, C_q$ is identical to the distribution of output of a sequence that, for each $1 \leq i \leq q$, draws x_i from $\mu_{\mathcal{S},I}$ conditioned on C_i with the fallback of uniformly drawing x_i from C_i if $\mu_{\mathcal{S},I}(C_i) = 0$. This bound holds also for an adaptive choice of each C_i based on $x_1, \ldots, x_{i-1}$.*

Proof. Observe that, if $\mu_{\mathcal{S},I}(C) > 0$, then using the sets I' and J' only affects the query complexity, and not the distribution of the output, since we ignore zeroes. Hence, Algorithm 21 is identical to a rejection sampler of the distribution over $\{1, \ldots, k\}$ defined by w with respect to the event I.

If $\mu_{\mathcal{S},I}(C) = 0$, then in every iteration of the while loop in which $\hat{C} \neq \emptyset$, the call to the weighted oracle results in an output of the form $(i, 0)$ for some $i \in \hat{C}$. Since the algorithm keeps history, every such i is then excluded from further iterations, and hence after at most k steps the set $\hat{C}$ becomes empty. When this happens, the algorithm exits the loop and uniformly draws $x \sim C$. □

Lemma 10.49. *Let $q \geq 5$ and $\mathcal{I}$ be a paired $4q$-uniform family of subsets of $\{1, \ldots, k\}$. If I is drawn uniformly from $\mathcal{I}$, then with probability at least $\frac{9}{10}$, the simulator uses at most $4q$ weighted sampling oracle calls to simulate a sequence of q conditional samples from $\mu_{\mathcal{S},I}$ (according to the scheme of Lemma 10.48).*

Proof. Consider the ith sampling oracle call for $1 \leq i \leq 4q$ (inside the jth call of Algorithm 21 for some $1 \leq j \leq q$). The probability to terminate is at least $\frac{1}{2}$: if we query an already-queried index (or its paired index) then we always terminate (because we take care to never query an index that is already known to be zero), and if we query a new bit, then the probability that its value is 1 is exactly $\frac{1}{2}$, even if conditioned on past queries, due to the $4q$-uniformness and Observation 10.32.

The probability to make $4q$ oracle calls before the qth termination is bounded by (Chernoff):

$$\Pr\left[\mathrm{Bin}(4q, 1/2) < q\right] \leq e^{-2(2q-q)^2/(4q)} = e^{-q/2} < \frac{1}{10}$$ □

Lemma 10.50. *Consider the sequence where $N_1 = 1$ and for every $i \geq 2$, $N_i = \lceil (1 + 120\varepsilon) N_{i-1} \rceil$. For every $N \geq 1$, $\varepsilon < \frac{1}{120}$ and $k \leq \ln N/(240\varepsilon)$, $\sum_{i=1}^{k} N_i < \frac{\sqrt{N} \log_2 N}{\varepsilon^2}$.*

We prove Lemma 10.50 in Appendix F.

Theorem 10.51 (Lower bound for label-invariant testing)**.** *For every sufficiently small $\varepsilon > 0$ and every sufficiently large N, there exists a label-invariant property of distributions over $\{1, \ldots, N\}$ for which every ε-test must draw $\Omega(\log N/\varepsilon)$ conditional samples.*

Proof. If $N \leq 1/\varepsilon^5$, then we can use the trivial lower-bound $\Omega(1/\varepsilon^2) = \omega(\log N/\varepsilon)$ of distinguishing between the uniform distribution over $\{1, 2\}$ and the distribution that draws 1 with probability $\frac{1}{2} + \frac{1}{2}\varepsilon$ and 2 with probability $\frac{1}{2} - \frac{1}{2}\varepsilon$. In the following we assume that $N > 1/\varepsilon^5$. For sufficiently small ε, this implies that $\sqrt{N} \ln N/\varepsilon^2 \leq N$.

Let $\Omega = \{1, \ldots, N\}$, $q = 2\left\lfloor \frac{1}{2} \ln N/(4800 \cdot \varepsilon) \right\rfloor$, $r = \frac{5}{2}q$, $k' = 4r = 10q$, $k = 2k' = 20q$. If $q = 0$ then the lower bound of one query is trivial. Hence, in the following we assume that N is sufficiently large to have $q \geq 1$.

Observe that $k \leq \ln N/(240\varepsilon)$. Let $\mathcal{S} = (S_1, \ldots, S_k)$ be the following $1 + 120\varepsilon$-increasing non-empty k-partition: let $N_1 = 1$ and $N_i = \lceil (1 + 120\varepsilon) N_{i-1} \rceil$ for every $2 \leq i \leq k$. The size of S_i is N_i for $1 \leq i \leq k - 1$ and at least N_k for $i = k$. Such a partition exists since Lemma 10.50 guarantees that $\sum_{i=1}^{k} N_i \leq \frac{\sqrt{N} \ln N}{\varepsilon^2} \leq N$.

By Lemma 10.40, there exists two paired q-uniform ($q = \frac{2}{5}r$) properties $\mathcal{I}_1$ and $\mathcal{I}_2$ of subsets of $\{1, \ldots, k\}$ ($k = 8r$) that are $\frac{1}{30}$-pairwise far and that only consist of subsets of size $\frac{1}{2}k = k'$.

By Lemma 10.37, every algorithm that distinguishes between $\mathcal{I}_1$ and $\mathcal{I}_2$ with success probability greater than $1/2$ must make at least q calls to the weighted sampling oracle.

By Lemma 10.47, for every $\mu_1 \in \mathcal{P}_{\mathcal{S},\mathcal{I}_1}$ and $\mu_2 \in \mathcal{P}_{\mathcal{S},\mathcal{I}_2}$, μ_2 is $\frac{1}{120}\varepsilon$-far from every relabeling of μ_1.

Consider a $\frac{1}{120}\varepsilon$-testing algorithm $\mathcal{A}$ for $\mathcal{P}_{\mathcal{S},\mathcal{I}_1}$. In particular, $\mathcal{A}$ distinguishes between $\mathcal{P}_{\mathcal{S},\mathcal{I}_1}$ and $\mathcal{P}_{\mathcal{S},\mathcal{I}_2}$ with success probability at least $2/3$.

By Lemma 10.49, we can construct an algorithm $\mathcal{A}'$ that distinguishes between $\mathcal{I}_1$ and $\mathcal{I}_2$ by simulating the conditional sampling calls of $\mathcal{A}$ using weighted sampling. With probability at least $9/10$, the number of weighted samples used by the simulation is at most four times the number of conditional samples drawn by $\mathcal{A}$. If we terminate the simulation after reaching this bound, it can still distinguish between $\mathcal{I}_1$ and $\mathcal{I}_2$ with probability at least $2/3 - 1/10 > 1/2$.

Since $\mathcal{I}_1$ and $\mathcal{I}_2$ are indistinguishable using q or fewer weighted samples with any success probability greater than $1/2$, $\mathcal{A}$ must draw strictly more than $q/4 = \Omega(\log N/\varepsilon)$ conditional samples. □

Corollary 10.52. *For every sufficiently small $\varepsilon > 0$ and every sufficiently large N, every algorithm that solves that ε-histogram learning task must draw $\Omega(\log N/\varepsilon)$ conditional samples.*

Proof. This holds since we can ε-test any label-invariant property by $\varepsilon/4$-histogram learning the input distribution (see Corollary 9.22). □

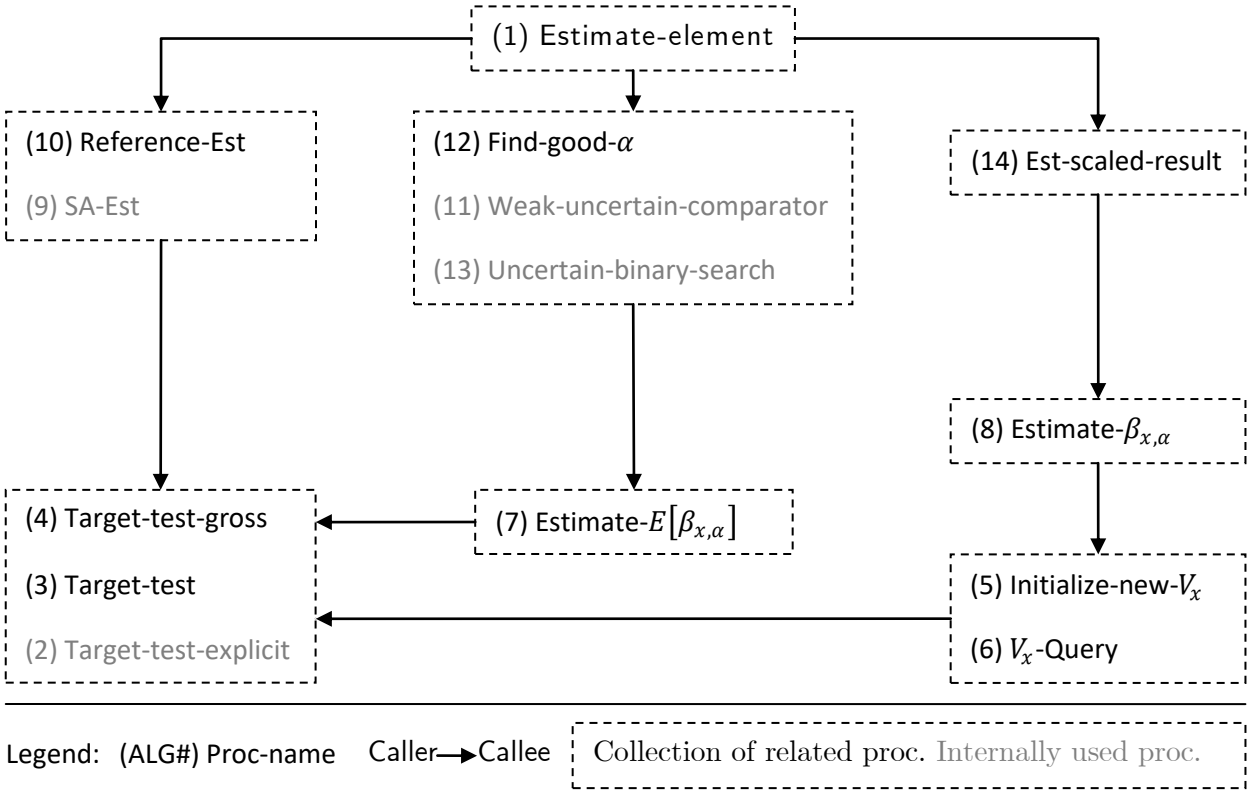
A Summary of paper notations

The following table summarizes the specific notation (mostly set in Subsection 3.2) that is used throughout this paper.

Notation	Name	Short description	Definition
L_x	The x-light set	Set of $\mu(y) \leq \mu(x)$	3.11
M_x	The x-medium set	Set of $\mu(x) < \mu(y) < 1.2\mu(x)$	3.11
H_x	The x-heavy set	Set of $\mu(y) \geq 1.2\mu(x)$	3.11
$\eta_{c,\varepsilon}$	The target error	$\min\left\{\frac{1}{4}c\varepsilon, \frac{1}{10^9}, \frac{\varepsilon^5}{10^{20}(\ln \varepsilon^{-1})^5}\right\}$	3.12
f_x, $f_{x,c,e}$	The target function	The acceptance probability of a canonical procedure (Target-test) that distinguishes between L_x and H_x	3.16
V_x, $V_{x,c,\varepsilon}$	The target set	Contains all of L_x, a random subset of M_x (using f_x), and disjoint from H_x	3.17
s_x, $s_{x,c,\varepsilon}$	The scale mass	$\mathrm{E}[\mu(V_x)]$	3.18
w_x, $w_{x,c,\varepsilon}$	The weight of x	$\mu(x) + s_x$, "the CDF of x and possibly a bit more"	3.19
α	The filter density	A parameter for the filter set	3.20
A_α	The filter set	Every element except x belongs to A_α with probability α, iid	3.20
$V_{x,\alpha}$, $V_{x,\alpha,c,\varepsilon}$	The filtered target set	$V_x \cap A_\alpha$	3.21
$\beta_{x,\alpha}$, $\beta_{x,\alpha,c,\varepsilon}$	The filtered density	$\Pr_\mu[\neg x \mid V_{x,\alpha} \cup \{x\}]$, equals $\mu(V_{x,\alpha}) / \mu(V_{x,\alpha} \cup \{x\})$	3.23
γ_x	The goal magnitude	$\mu(x)/s_x$, a good α is $\Theta(\gamma_x)$	3.22
κ	Used in Target-test-explicit	Hard-coded to $10^{-9}/45$	Section 5
$h(\beta)$	Truncated assessment function	$h(\beta) = \min\{\beta/(1-\beta), T\}$, $T = 8\ln\varepsilon^{-1} + 100$	Section 8
$\mathrm{Ct}[X\|B]$	Contribution of X over B	$\mathrm{E}[X\|B] \cdot \Pr[B]$	3.34
$D_{\mathrm{H}}(\mu;\tau)$	Histogram divergence	Minimum ε for which there exists a permutation π over the domain such that $\Pr[\mu(x) \notin (1 \pm \varepsilon)\tau(\pi(x))] \leq \varepsilon$	3.8

B Procedural dependency chart

The following diagram describes the calling dependencies of the various procedures defined in Sections 4 through 8. These are grouped by function. Procedures in gray are not called from any procedure outside their group, but they may themselves call an outside procedure (following the outgoing arrows) at the behest of the procedure that called them from their own group.



C Reducing the extreme constants in the estimator at a cost

The target-test (Algorithm 2) presented in Section 5 requires impractical constant factors, which carry over to the estimator. In this appendix we show that, at the cost of an additional $O(\log \frac{1}{\varepsilon c})$-penalty applied to the $\log \log N$-factor of Estimate-element, we can significantly reduce them.

Most of the algorithmic parts require that, for every $y \in \Omega$:

$$|\Pr[y \in V_x] - \Pr[\textsf{Target-test}(x, y) = \text{ACCEPT}]| \leq \eta_{c,\varepsilon}$$

Apart from that, Procedure Estimate-E$[\beta_{x,\alpha}]$ requires that, for every $y \in \Omega$:

$$|\Pr[y \in V_x] - \Pr[\textsf{Target-test-gross}(x, y) = \text{ACCEPT}]| \leq \frac{1}{10^8}$$

We observe that, if Target-test-gross would be identical to Target-test, then it would still satisfy the second constraint, since for $y \in L_x \cup H_x$ this follows from the first constraint (and from the bound $\eta_{c,\varepsilon} \leq 10^8$), and for $y \in M_x$, $\Pr[y \in V_x]$ and $\Pr[\textsf{Target-test}(x, y) = \text{ACCEPT}]$ are identical by definition and hence their difference is zero. Hence, it suffices to implement Target-test more efficiently and use it instead of Target-test-gross. Note that this is where the $O(\log \frac{1}{c\varepsilon})$-penalty comes from: the binary search makes $O(\log \log N)$ calls to Estimate-E$[\beta_{x,\alpha}]$, and every such a call under this scheme now uses Target-test, whose implementation requires $O(\log \frac{1}{c\varepsilon})$ conditional samples rather than $O(1)$ conditional samples.

At this point we present Target-test-lightweight, which is a cheaper implementation of Target-test (but does not have a compatible "gross" version), and show that it satisfies the required constraints.

Algorithm 22: Procedure Target-test-lightweight$(\mu, c, \varepsilon; x, y)$
Input: $y \in \Omega$.
Output: ACCEPT or REJECT.

1. If $y = x$: Technical guarantee
   (a) REJECT
2. Let $\ell \leftarrow \left\lceil 968 \ln \eta_{c,\varepsilon}^{-1} \right\rceil$.
3. Draw $z_1, \ldots, z_\ell$ independent samples from μ conditioned on $\{x, y\}$.
4. Let $Y = |\{i : z_i = y\}|$.
5. If $Y < \frac{23}{44}\ell$:
   (a) ACCEPT.
6. Else:
   (a) REJECT.

Lemma C.1 (Target-test-lightweight)**.** *Algorithm 22 uses $O(\log \frac{1}{\varepsilon c})$ conditional samples, accepts with probability at least $1 - \eta_{c,\varepsilon}$ if the input belongs to L_x and rejects with probability at least $1 - \eta_{c,\varepsilon}$ if the input belongs to H_x.*

Proof. Observe that $Y \sim \mathrm{Bin}\left(\ell, \frac{\mu(y)}{\mu(x)+\mu(y)}\right)$, for $\ell \geq 968 \ln(1/\eta_{c,\varepsilon})$.

If $y \in L_x$, then $\mu(y) \leq \mu(x)$ and $\mathrm{E}[Y] \leq \frac{1}{2}\ell$. The probability to reject is bounded by $e^{-2\left(\frac{23}{44}-\frac{1}{2}\right)^2 \ell} \leq e^{-\frac{1}{968}\ell} \leq e^{-\frac{968}{968}\ln(1/\eta_{c,\varepsilon})} = \eta_{c,\varepsilon}$.

If $y \in H_x$, then $\mu(y) \geq 1.2\mu(x)$ and $\mathrm{E}[Y] \geq \frac{6}{11}\ell$. The probability to accept is bounded by $e^{-2\left(\frac{23}{44}-\frac{6}{11}\right)^2 \ell} \leq e^{-\frac{1}{968}\ell} \leq e^{-\frac{968}{968}\ln(1/\eta_{c,\varepsilon})} = \eta_{c,\varepsilon}$.

Since $\eta_{c,\varepsilon}$ is the minimum of three expressions that are all polynomial in c and ε, the complexity is $O\left(\log \eta_{c,\varepsilon}^{-1}\right) = O\left(\log \frac{1}{\varepsilon c}\right)$. □

D Another generic application lemma

We provide here a variant of Lemmas 9.10, 9.23 and 9.24. While not used in the application examples that we provided Section 9, we believe that it has potential for future similar applications.

Definition D.1 (ε-explicit sampling oracle)**.** Let μ be an input distribution over a set Ω. The *ε-explicit sampling oracle* for μ has no additional input, and outputs a pair (x, p), where $x \in \Omega$ distributes like μ and with probability 1, $p \in (1 \pm \varepsilon)\mu(x)$.

The oracle guarantees *consistency*, which means that if some element y is drawn more than once, then all pairs of the form $(y, \cdot)$ have the same second entry.

Note that the ε-explicit sampling oracle is a restricted case of the r-lying (c, ε)-explicit sampling oracle (Definition 9.5) when using $r = c = 0$. In particular, following Observation 9.6, this oracle can be thought of as the result of sampling and receiving the corresponding values of an arbitrary (possibly probabilistic) ε-approximation function $g_{\mathrm{truth}} : \Omega \to [0, 1]$ along with the samples.

Lemma D.2. *Consider an algorithm $\mathcal{A}$ whose input is a k-tuple $\vec{\mu} = (\mu_1, \ldots, \mu_k)$ of distributions over $\Omega_1, \ldots, \Omega_k$ (respectively), and its output is an element of a discrete set R. Assume that $\mathcal{A}$ makes at most q calls to the ε-explicit sampling oracle. Let $N = \max\{|\Omega_1|, \ldots, |\Omega_k|\}$.*

Assume that for every input $\vec{\mu}$ there exists a set $R_{\vec{\mu}} \subseteq R$ for which $\Pr\left[\mathcal{A}(\vec{\mu}) \in R_{\vec{\mu}}\right] > \frac{2}{3}$ whenever the algorithm is supplied with ε-explicit sampling oracles for the input distributions.

In this setting, there exists an algorithm $\mathcal{A}'$ in the fully conditional model whose sample complexity is $O(q \log q \log\log N + \frac{q}{\varepsilon^4}\mathrm{poly}(\log q, \log \varepsilon^{-1}))$, such that for every input $\vec{\mu}$, $\Pr\left[\mathcal{A}'(\vec{\mu}) \in R_{\vec{\mu}}\right] > \frac{13}{24}$.

Proof. We run $\mathcal{A}$ and simulate the outcome of the ε-sampling oracle. In each call to the ε-sampling oracle for μ_i, we unconditionally draw $x_i \sim \mu_i$ and call Estimate-element with parameters $(\mu_i, 1/24q, \varepsilon)$ on x_i (Theorem 4.1). We amplify the success probability to $1 - \frac{1}{24q}$ using the median of $\lceil 30 \ln(12q) \rceil$ such calls (Observation 3.32(d)). Each time we estimate the probability mass of an element, we record it in a "history". If the same element is sampled again later, we use the history record rather than calling the Estimate-element procedure again. This guarantees the consistency of the oracle (required by Definition D.1).

The probability to draw an element x_i for which $\mathrm{CDF}_{\mu_i}(x_i) < 1/24q$ is clearly bounded by $1/24q$. Therefore, by the union bound, the probability to draw such a rare element within q samples is bounded by $1/24$.

The probability to have a wrong estimation for any x_i, assuming that $\mathrm{CDF}_{\mu_i}(x_i) \geq 1/24q$ for all $1 \leq i \leq q$, is bounded by $q \cdot \frac{1}{24q} = \frac{1}{24}$. Hence, the probability to correctly simulate the ε-explicit sampling oracle is at least $11/12$. This is by the union bound over two bad events: a $1/24$ bound for the event of drawing a hard-to-estimate element, and another $1/24$ bound for the event of an incorrect estimation for an estimable element. If the simulation is correct, then the output of the simulated $\mathcal{A}$ belongs to $R_{\vec{\mu}}$ with probability at least $2/3$. Overall, the probability of the simulation to output an element in $R_{\vec{\mu}}$ is at least $2/3 - 1/12 = 7/12$.

By Corollary 4.3 (using $c = \frac{1}{24q}$), the expected complexity of a single estimation of x is $O(\log\log N) + O\left(\frac{\mathrm{poly}(\log q, \log \varepsilon^{-1})}{\varepsilon^4}\right)$. We repeat this $O(\log q)$ times for amplification of q oracle calls. Therefore, the expected sample complexity is at most $O(q \log q \log\log N + \frac{q}{\varepsilon^4}\mathrm{poly}(\log q, \log \varepsilon^{-1}))$.

By Markov's inequality, with probability at least $23/24$, the actual sample complexity is at most 24 times the expected complexity, which is asymptotically equivalent. Overall, with probability at least $7/12 - 1/24 = 13/24$, the algorithm terminates after $O(q \log q \log\log N + \frac{q}{\varepsilon^4}\mathrm{poly}(\log q, \log \varepsilon^{-1}))$ samples and outputs an element in $R_{\vec{\mu}}$. □

E Technical analysis of the filtered target set

E.1 Concentration inequalities for the filtered target set $V_{x,\alpha}$

We prove here some Chernoff-like inequalities for the mass of $V_{x,\alpha}$, derived (unsurprisingly) by first proving a bound on the expectation of its Moment Generating Function.

Lemma E.1 (Moment Generating Function of $\mu(V_{x,\alpha})$)**.** *For every $r \leq 1$ (possibly negative) and $0 < \alpha \leq 1$,* $\mathrm{E}\left[e^{\frac{r}{1.2\mu(x)}(V_{x,\alpha} - \mathrm{E}[\mu(V_{x,\alpha})])}\right] \leq e^{\frac{5r^2}{8\mu(x)}\mathrm{E}[\mu(V_{x,\alpha})]}$.

Proof. For every $y \in M_x$, let X_y be a random variable that gets $\mu(y)$ with probability $f_x(y)\alpha$ and 0 otherwise, and let $X = \sum_{y \in M_x} X_y$. Clearly, $\mathrm{E}[X] = \alpha \sum_{y \in M_x} f_x(y)\mu(y) = \mathrm{E}[\mu(V_{x,\alpha})] - \mu(L_x)$.

Recall that $e^x \leq 1 + x + \frac{3}{4}x^2$ for every $x \leq 1$, and let $\lambda \leq \frac{1}{1.2\mu(x)}$ (possibly negative).

$$\begin{aligned}
\mathrm{E}\left[e^{\lambda(X_y - \mathrm{E}[X_y])}\right] &= e^{-\lambda\alpha f_x(y)\mu(y)} \cdot (\alpha f_x(y) e^{\lambda\mu(y)} + (1 - \alpha f_x(y))) \\
&= e^{-\lambda\alpha f_x(y)\mu(y)} \cdot (\alpha f_x(y)(e^{\lambda\mu(y)} - 1) + 1) \\
[\text{Since } \lambda\mu(y) \leq 1] &\leq e^{-\lambda\alpha f_x(y)\mu(y)} \cdot \left(\alpha f_x(y)(1 + (\lambda\mu(y)) + \frac{3}{4}(\lambda\mu(y))^2 - 1) + 1\right) \\
&= e^{-\lambda\alpha f_x(y)\mu(y)} \cdot \left(\alpha f_x(y)((\lambda\mu(y)) + \frac{3}{4}(\lambda\mu(y))^2) + 1\right)
\end{aligned}$$

We use the well-known bound $t + 1 \leq e^t$ to obtain:

$$\begin{aligned}
\mathrm{E}\left[e^{\lambda(X_y - \mathrm{E}[X_y])}\right] &\leq e^{-\lambda\alpha f_x(y)\mu(y)} \cdot e^{\alpha f_x(y)((\lambda\mu(y)) + \frac{3}{4}(\lambda\mu(y))^2)} \\
&= e^{\frac{3}{4}\alpha f_x(y)(\lambda\mu(y))^2} \\
&= e^{\frac{3}{4}\lambda^2\mu(y)\cdot(\alpha f_x(y)\mu(y))} \leq e^{\frac{3}{4}\cdot 1.2\lambda^2\cdot\mu(x)\cdot\mathrm{E}[X_y]} = e^{\frac{9}{10}\lambda^2\cdot\mu(x)\cdot\mathrm{E}[X_y]}
\end{aligned}$$

Since the X_ys are independent, this implies that $e^{\lambda(X - \mathrm{E}[X])} \leq e^{\frac{9}{10}\lambda^2\mu(x)\cdot\mathrm{E}[X]}$. We choose $\lambda = \frac{r}{1.2\cdot\mu(x)}$ to obtain the desired bound. □

Lemma E.2. *For every* $0 \leq \delta \leq 1$, $\Pr[\mu(V_{x,\alpha}) \geq (1+\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]] \leq e^{-\frac{1}{4\mu(x)}\delta^2\,\mathrm{E}[\mu(V_{x,\alpha})]}$.

Proof. We use Lemma E.1 using $r = \frac{3}{4}\delta$ to obtain $\mathrm{E}\left[e^{\frac{5\delta}{8\mu(x)}(V_{x,\alpha} - \mathrm{E}[\mu(V_{x,\alpha})])}\right] \leq e^{\frac{45\delta^2}{128\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}$.

We now use Chernoff-Markov bound:

$$\begin{aligned}
\Pr\left[\mu(V_{x,\alpha}) \geq (1+\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]\right] &= \Pr\left[\mu(V_{x,\alpha}) - \mathrm{E}[\mu(V_{x,\alpha})] \geq \delta\,\mathrm{E}[\mu(V_{x,\alpha})]\right] \\
&= \Pr\left[e^{\frac{5\delta}{8\mu(x)}(\mu(V_{x,\alpha}) - \mathrm{E}[\mu(V_{x,\alpha})])} \geq e^{\frac{5\delta^2}{8\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}\right] \\
&\leq e^{-\frac{5\delta^2}{8\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}\,\mathrm{E}\left[e^{\frac{5\delta}{8\mu(x)}(\mu(V_{x,\alpha}) - \mu(\mathrm{E}[V_{x,\alpha}])])}\right] \\
&\leq e^{-\frac{5\delta^2}{8\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]} e^{\frac{45\delta^2}{128\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]} \\
&= e^{-\frac{35}{128\mu(x)}\delta^2\,\mathrm{E}[\mu(V_{x,\alpha})]} \leq e^{-\frac{\delta^2}{4\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}
\end{aligned}$$

□

Lemma E.3. *For every* $0 \leq \delta \leq 1$, $\Pr[\mu(V_{x,\alpha}) \leq (1-\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]] \leq e^{-\frac{1}{4\mu(x)}\delta^2\,\mathrm{E}[\mu(V_{x,\alpha})]}$.

Proof. We use Lemma E.1 using $r = -\frac{3}{4}\delta$ to obtain $\mathrm{E}\left[e^{-\frac{5\delta}{8\mu(x)}(V_{x,\alpha} - \mathrm{E}[\mu(V_{x,\alpha})])}\right] \leq e^{\frac{45\delta^2}{128\mu(x)}\,\mathrm{E}[X_y]}$.

We now use Chernoff-Markov bound:

$$\begin{aligned}
\Pr\left[\mu(V_{x,\alpha}) \le (1-\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]\right] &= \Pr\left[\mu(V_{x,\alpha}) - \mathrm{E}[\mu(V_{x,\alpha})] \le -\delta\,\mathrm{E}[\mu(V_{x,\alpha})]\right] \\
&= \Pr\left[e^{-\frac{5\delta}{8\mu(x)}(\mu(V_{x,\alpha})-\mathrm{E}[\mu(V_{x,\alpha})])} \ge e^{\frac{5\delta^2}{8\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}\right] \\
&\le e^{-\frac{5\delta^2}{8\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}\,\mathrm{E}\left[e^{-\frac{5\delta}{8\mu(x)}(\mu(V_{x,\alpha})-\mu(\mathrm{E}[\mu(V_{x,\alpha})])}\right] \\
&\le e^{-\frac{5\delta^2}{8\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]} e^{\frac{45\delta^2}{128\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]} \\
&= e^{-\frac{35}{128\mu(x)}\delta^2\,\mathrm{E}[\mu(V_{x,\alpha})]} \le e^{-\frac{\delta^2}{4\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}
\end{aligned}$$

□

Lemma E.4. *For every $\delta > 0$, $\Pr[\mu(V_{x,\alpha}) \ge (2+\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]] \le e^{-\frac{1}{2\mu(x)}\delta\,\mathrm{E}[\mu(V_{x,\alpha})]}$.*

Proof. We use Lemma E.1 using $r = 1$ to obtain $\mathrm{E}\left[e^{\frac{1}{1.2\mu(x)}(\mu(V_{x,\alpha})-\mathrm{E}[\mu(V_{x,\alpha})])}\right] \le e^{\frac{5}{8\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}$.

We now use Chernoff-Markov bound:

$$\begin{aligned}
\Pr\left[\mu(V_{x,\alpha}) \ge (2+\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]\right] &= \Pr\left[\mu(V_{x,\alpha}) - \mathrm{E}[\mu(V_{x,\alpha})] \ge (1+\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]\right] \\
&= \Pr\left[e^{\frac{1}{1.2\mu(x)}(\mu(V_{x,\alpha})-\mathrm{E}[\mu(V_{x,\alpha})])} \ge e^{\frac{1}{1.2\mu(x)}(1+\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]}\right] \\
&\le e^{-\frac{1}{1.2\mu(x)}(1+\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]}\,\mathrm{E}\left[e^{\frac{1}{1.2\mu(x)}(\mu(V_{x,\alpha})-\mathrm{E}[\mu(V_{x,\alpha})])}\right] \\
&\le e^{-\frac{1}{1.2\mu(x)}(1+\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]} e^{\frac{5}{8\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]} \\
&\le e^{-\frac{\delta}{1.2\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]} \le e^{-\frac{\delta}{2\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]}
\end{aligned}$$

□

E.2 Expectation inequalities

Lemma E.5. *For every $0 < \alpha \le 1$, $\mathrm{E}\left[\frac{1}{\mu(x)+\mu(V_{x,\alpha})}\right] \le \frac{1}{\mu(x)} \cdot \left(e^{-\frac{1}{16}a} + \frac{1}{1+\frac{1}{2}a}\right)$, where $a = \alpha/\gamma_x$.*

Proof.

$$\begin{aligned}
&\mathrm{E}\left[\frac{1}{\mu(x)+\mu(V_x \cap A_\alpha)}\right] \\
&\le \Pr\left[\mu(V_{x,\alpha}) < \frac{1}{2}\,\mathrm{E}[\mu(V_{x,\alpha})]\right] \cdot \frac{1}{\mu(x)} + \Pr\left[\mu(V_{x,\alpha}) \ge \frac{1}{2}\,\mathrm{E}[\mu(V_{x,\alpha})]\right] \cdot \frac{1}{\mu(x)+\frac{1}{2}\,\mathrm{E}[\mu(V_{x,\alpha})]} \\
(*) \quad &\le e^{-\frac{1}{16\mu(x)}\,\mathrm{E}[\mu(V_{x,\alpha})]} \cdot \frac{1}{\mu(x)} + 1 \cdot \frac{1}{\mu(x)+\frac{1}{2}\,\mathrm{E}[\mu(V_{x,\alpha})]} \\
(**) \quad &= \frac{e^{-\frac{1}{16}a}}{\mu(x)} + \frac{1}{\mu(x)+\frac{1}{2}a\mu(x)} = \frac{1}{\mu(x)} \cdot \left(e^{-\frac{1}{16}a} + \frac{1}{1+\frac{1}{2}a}\right)
\end{aligned}$$

(∗): By Lemma E.3 (Chernoff $\Pr[\mu(V_{x,\alpha}) < (1-\delta)\,\mathrm{E}[\mu(V_{x,\alpha})]]$ with $\delta = \frac{1}{2}$),
(∗∗): Since $\mathrm{E}[\mu(V_{x,\alpha})] = a\mu(x)$. □

At this point we recall and prove Lemma 5.8.

Lemma 5.8. $\mathrm{E}\left[\frac{\mu(A_\alpha \cup \{x\})}{\mu((V_x \cap A_\alpha) \cup \{x\})}\right] \leq \frac{20}{w_x}$.

Proof. Recall that $w_x = \mu(x) + s_x$.

Case I: $\mu(x) \geq \frac{1}{2} w_x$:

$$\mathrm{E}\left[\frac{\mu(A_\alpha \cup \{x\})}{\mu\left((V_x \cap A_\alpha) \cup \{x\}\right)}\right] \leq \mathrm{E}\left[\frac{\mu(A_\alpha \cup \{x\})}{\mu(x)}\right] \leq \mathrm{E}\left[\frac{1}{\mu(x)}\right] = \frac{1}{\mu(x)} \leq \frac{2}{w_x}$$

Case II: $\mu(x) < \frac{1}{2} w_x$ and hence $s_x \geq \frac{1}{2} w_x$. Let $a = \alpha/\gamma_x$ (hence $\mathrm{E}[\mu(V_{x,\alpha})] = a\mu(x)$).

$$\begin{aligned}
\mathrm{E}\left[\frac{\mu(A_\alpha \cup \{x\})}{\mu((V_x \cap A_\alpha) \cup \{x\})}\right] &= \mathrm{E}\left[\frac{\mu((V_x \cap A_\alpha) \cup \{x\}) + \mu(A_\alpha \setminus V_x)}{\mu((V_x \cap A_\alpha) \cup \{x\})}\right] \\
&= 1 + \mathrm{E}\left[\frac{\mu(A_\alpha \setminus V_x)}{\mu((V_x \cap A_\alpha) \cup \{x\})}\right] \\
&= 1 + \sum_U \Pr[V_x = U] \, \mathrm{E}\left[\frac{\mu(A_\alpha \setminus U)}{\mu((U \cap A_\alpha) \cup \{x\})}\right]
\end{aligned}$$

Since U is hard-coded, the random set $A_\alpha \setminus U$, which contains every element in $\Omega \setminus (U \cup \{x\})$ with probability α independently, is independent of the random set $A_\alpha \cap U$, which contains every element in U with probability α independently. Hence,

$$\begin{aligned}
\mathrm{E}\left[\frac{\mu(A_\alpha \cup \{x\})}{\mu((V_x \cap A_\alpha) \cup \{x\})}\right] &= 1 + \sum_U \Pr[V_x = U] \, \mathrm{E}[\mu(A_\alpha \setminus U)] \, \mathrm{E}\left[\frac{1}{\mu((U \cap A_\alpha) \cup \{x\})}\right] \\
&\leq 1 + \sum_U \Pr[V_x = U] \cdot \alpha \, \mathrm{E}\left[\frac{1}{\mu((U \cap A_\alpha) \cup \{x\})}\right] \\
&= 1 + \alpha \, \mathrm{E}\left[\frac{1}{(\mu(V_x) \cap A_\alpha) \cup \{x\})}\right] \\
&= 1 + \alpha \, \mathrm{E}\left[\frac{1}{\mu(x) + \mu(V_{x,\alpha})}\right]
\end{aligned}$$

We can use Lemma E.5 to obtain that:

$$\begin{aligned}
\mathrm{E}\left[\frac{\mu(A_\alpha \cup \{x\})}{\mu((V_x \cap A_\alpha) \cup \{x\})}\right] &\leq 1 + \alpha \cdot \frac{e^{-\frac{1}{16}a} + \frac{1}{1+\frac{1}{2}a}}{\mu(x)} \\
[\alpha = a\mu(x)/s_x] &= 1 + \frac{a\mu(x)}{s_x} \cdot \frac{e^{-\frac{1}{16}a} + \frac{1}{1+\frac{1}{2}a}}{\mu(x)} \\
&= 1 + \frac{1}{s_x} \cdot a \left(e^{-\frac{1}{16}a} + \frac{1}{1+\frac{1}{2}a}\right) \\
(*) &\leq 1 + \frac{16e^{-1} + 2}{s_x} \leq 1 + \frac{8}{s_x} \leq \frac{9}{s_x} \leq \frac{20}{w_x}
\end{aligned}$$

$(*)$: since $ae^{-a/16} = 16(a/16)e^{-(a/16)} \leq 16(\sup_{t \geq 0} te^{-t}) = 16e^{-1}$ and $\frac{a}{1+a/2} \leq 2$ for $a > 0$. □

Lemma E.6. *For every $0 < \alpha \le 1$, $\mathrm{E}[\beta_{x,\alpha}] \ge \left(1 - e^{-a/9}\right)\left(1 - \frac{3}{3+a}\right)$, where $a = \alpha/\gamma_x$.*

Proof.

$$\begin{aligned}
\mathrm{E}\left[\frac{\mu(V_{x,\alpha})}{\mu(x)+\mu(V_{x,\alpha})}\right] &\ge \Pr\left[\mu(V_{x,\alpha}) \ge \left(1-\frac{2}{3}\right)\mathrm{E}[\mu(V_{x,\alpha})]\right] \cdot \frac{\left(1-\frac{2}{3}\right)\mathrm{E}[\mu(V_{x,\alpha})]}{\mu(x)+\left(1-\frac{2}{3}\right)\mathrm{E}[\mu(V_{x,\alpha})]} \\
\text{[Lemma E.2]} &\ge \left(1 - e^{-\frac{(2/3)^2}{4\mu(x)}\mathrm{E}[\mu(V_{x,\alpha})]}\right) \cdot \frac{\frac{1}{3}\mathrm{E}[\mu(V_{x,\alpha})]}{\mu(x)+\frac{1}{3}\mathrm{E}[\mu(V_{x,\alpha})]} \\
&= \left(1 - e^{-\frac{1}{9\mu(x)}\cdot a\mu(x)}\right) \cdot \frac{\frac{1}{3}\cdot a\mu(x)}{\mu(x)+\frac{1}{3}a\mu(x)} \\
&= \left(1 - e^{-\frac{1}{9}a}\right) \cdot \frac{a}{3+a} = \left(1 - e^{-\frac{1}{9}a}\right) \cdot \left(1 - \frac{3}{3+a}\right)
\end{aligned}$$

□

Lemma E.7. *For every $0 < \alpha \le 1$, $\mathrm{E}[\beta_{x,\alpha}] \le \frac{2\sqrt{a^2+a}}{1+a+\sqrt{a^2+a}}$, where $a = \alpha/\gamma_x$.*

Proof. Recall that $\mathrm{E}[\mu(V_{x,\alpha})] = a\mu(x)$. Let $k_a = 1 + \sqrt{1+a^{-1}}$.

$$\begin{aligned}
\mathrm{E}[\beta_{x,\alpha}] &= \mathrm{E}\left[\frac{\mu(V_{x,\alpha})}{\mu(x)+\mu(V_{x,\alpha})}\right] \\
&\le \Pr[\mu(V_{x,\alpha}) \le k_a a\mu(x)] \cdot \frac{k_a a\mu(x)}{\mu(x)+k_a a\mu(x)} + \Pr[\mu(V_{x,\alpha}) > k_a a\mu(x)] \cdot 1 \\
&= (1 - \Pr[\mu(V_{x,\alpha}) > k_a a\mu(x)]) \cdot \frac{k_a a\mu(x)}{\mu(x)+k_a a\mu(x)} + \Pr[\mu(V_{x,\alpha}) > k_a a\mu(x)] \cdot 1 \\
&= \frac{ak_a}{1+ak_a} + \Pr[\mu(V_{x,\alpha}) > k_a a\mu(x)] \cdot \left(1 - \frac{ak_a}{1+ak_a}\right)
\end{aligned}$$

By Markov's inequality, $\Pr[\mu(V_{x,\alpha}) > k_a a\mu(x)] < \frac{1}{k_a}$, and hence:

$$\begin{aligned}
\mathrm{E}[\beta_{x,\alpha}] &\le \frac{ak_a}{1+ak_a} + \frac{1}{k_a} \cdot \left(1 - \frac{ak_a}{1+ak_a}\right) \\
&= \frac{1}{1+ak_a}\left(ak_a + \frac{1}{k_a}\right) \\
&= \frac{1}{1+a(1+\sqrt{1+a^{-1}})}\left(a(1+\sqrt{1+a^{-1}}) + \frac{1}{1+\sqrt{1+a^{-1}}}\right) \\
&= \frac{1}{1+a(1+\sqrt{1+a^{-1}})}\left(a(1+\sqrt{1+a^{-1}}) - a\left(1-\sqrt{1+a^{-1}}\right)\right) \\
&= \frac{2\sqrt{a^2+a}}{1+a+\sqrt{a^2+a}}
\end{aligned}$$

□

Lemma E.8. *The function $\alpha \to \mathrm{E}[\beta_\alpha]$ is continuous in $0 < \alpha < 1$.*

Proof. Implied from Observation 3.31 since β_α is bounded between 0 and 1 with probability 1. □

We prove now Lemma 7.7, which we recall here.

Lemma 7.7 (Effective bounds for $\mathrm{E}[\beta_{x,\alpha}]$)**.** *There exists $2.3\gamma_x \le \alpha_x \le 38\gamma_x$ for which $\mathrm{E}[\beta_{x,\alpha_x}] = 0.91$. Additionally, if $\alpha \le 2\gamma_x$ then $\mathrm{E}[\beta_{x,\alpha}] < 0.9$ and if $\alpha \ge 41\gamma_x$ then $\mathrm{E}[\beta_{x,\alpha}] > 0.92$.*

Proof. We apply Lemma E.6 and Lemma E.7 to obtain the following bounds:

- $\mathrm{E}[\beta_{x,2\cdot\gamma_x}] \le \frac{2\sqrt{2^2+2}}{1+2+\sqrt{2^2+2}} < 0.9$, and by Observation 7.6, $\mathrm{E}[\beta_{x,\alpha}] < 0.9$ for all $\alpha \le 2\gamma_x$ as well.
- $\mathrm{E}[\beta_{x,41\gamma_x}] \ge \left(1 - e^{-41/9}\right)\left(1 - \frac{3}{3+41}\right) > 0.92$, and by Observation 7.6, $\mathrm{E}[\beta_{x,\alpha}] > 0.92$ for all $\alpha \ge 41\gamma_x$ as well.

Also, we use these lemmas to obtain:

- $\mathrm{E}[\beta_{x,2.3\cdot\gamma_x}] \le \frac{2\sqrt{(2.3)^2+2.3}}{1+2.3+\sqrt{(2.3)^2+2.3}} < 0.91$.
- $\mathrm{E}[\beta_{x,38\cdot\gamma_x}] \ge \left(1 - e^{-38/9}\right)\left(1 - \frac{3}{3+38}\right) > 0.91$.

Since the mapping $\alpha \to \mathrm{E}[\beta_{x,\alpha}]$ is continuous (Lemma E.8), the Intermediate Value Theorem guarantees the existence of $2.3\gamma_x < \alpha_x < 38\gamma_x$ for which $\mathrm{E}[\beta_{x,\alpha}] = 0.91$. □

E.3 Technical analysis of the assessment function $h(\beta_{x,\alpha})$

In this appendix we prove the technical lemmas of Section 8. Recall that we use $h(\beta) = \min\left\{\frac{\beta}{1-\beta}, T\right\}$ for $T = 8\ln\varepsilon^{-1} + 100$.

Lemma E.9. *Recall that $T = 8\ln\varepsilon^{-1} + 100$ and let $a = \alpha/\gamma_x$. If $1 \le a \le 50$ and $\varepsilon < \frac{1}{10}$, then $\mathrm{Ct}\left[\mu(V_{x,\alpha}) | \mu(V_{x,\alpha}) > T\mu(x)\right] \le \frac{1}{10}\varepsilon \cdot a\mu(x)$.*

Proof. Let $\hat{T} = 8\ln\varepsilon^{-1} + 2a \le 8\ln\varepsilon^{-1} + 100 = T$.

$$\begin{aligned}
\Pr\left[\mu(V_{x,\alpha}) > 2^t T\mu(x)\right] &\le \Pr\left[\mu(V_{x,\alpha}) > 2^t\hat{T}\mu(x)\right] \\
&= \Pr\left[\mu(V_{x,\alpha}) > (2^t\hat{T}/a)\,\mathrm{E}[\mu(V_{x,\alpha})]\right] \\
&= \Pr\left[\mu(V_{x,\alpha}) > 2^t\left(\frac{8}{a}\ln\varepsilon^{-1} + 2\right)\mathrm{E}[\mu(V_{x,\alpha})]\right] \\
&\le \Pr\left[\mu(V_{x,\alpha}) > \left(2 + 2^t\left(\frac{8}{a}\ln\varepsilon^{-1}\right)\right)\mathrm{E}[\mu(V_{x,\alpha})]\right] \\
[\text{Lemma E.4 (Chernoff)}] &\le e^{-\frac{2^{t+3}\ln\varepsilon^{-1}/a}{2\cdot\mu(x)}\cdot\mathrm{E}[\mu(V_{x,\alpha})]} = e^{-\frac{8\cdot 2^t\ln\varepsilon^{-1}}{2a\mu(x)}\cdot a\mu(x)} = e^{-2^{t+2}\ln\varepsilon^{-1}}
\end{aligned}$$

We obtain that:

$$\begin{aligned}
\mathrm{Ct}\left[\mu(V_{x,\alpha})|\mu(V_{x,\alpha}) > T\mu(x)\right] &\leq \mathrm{Ct}\left[\mu(V_{x,\alpha})\middle|\mu(V_{x,\alpha}) > \hat{T}\mu(x)\right] \\
&\leq \sum_{t=0}^{\infty} \Pr\left[2^t\hat{T}\mu(x) < \mu\left(V_x \cap A_\alpha\right) \leq 2^{t+1}\hat{T}\mu(x)\right] \cdot 2^{t+1}\hat{T}\mu(x) \\
&\leq 2\hat{T}\mu(x)\sum_{t=0}^{\infty} 2^t \Pr\left[\mu\left(V_x \cap A_\alpha\right) > 2^t\hat{T}\mu(x)\right] \\
&\leq 2\hat{T}\mu(x)\sum_{t=0}^{\infty} 2^t \cdot e^{-2^{t+2}\ln\varepsilon^{-1}} \\
&\leq 2\cdot(8\ln\varepsilon^{-1}+2a)\mu(x)\sum_{t=0}^{\infty} e^{t\ln 2 - 2^{t+2}\ln\varepsilon^{-1}}
\end{aligned}$$

Since $t\ln 2 - 2^{t+2}\ln\varepsilon^{-1} \leq -(t+4)\ln\varepsilon^{-1}$ for every $\varepsilon < e^{-1}$ and $t \geq 0$, we obtain that:

$$\begin{aligned}
\mathrm{Ct}\left[\mu(V_{x,\alpha})|\mu(V_{x,\alpha}) > T\mu(x)\right] &\leq (16\ln\varepsilon^{-1}+4a)\mu(x)\sum_{t=0}^{\infty} e^{-(t+4)\ln\varepsilon^{-1}} \\
&= (16\ln\varepsilon^{-1}+4a)\mu(x)\cdot\varepsilon^4\sum_{t=0}^{\infty}\varepsilon^t \\
[\varepsilon < 1/2] &\leq (16\ln\varepsilon^{-1}+4a)\mu(x)\cdot 2\varepsilon^4 \\
&= \left(\frac{32\ln\varepsilon^{-1}}{a}+8\right)\varepsilon^3\cdot\varepsilon a\mu(x) \\
[a \geq 1] &\leq (32\ln\varepsilon^{-1}+8)\varepsilon^3\cdot\varepsilon a\mu(x) \\
[\varepsilon < 1/10] &< \frac{1}{10}\varepsilon a\mu(x)
\end{aligned}$$

□

Lemma 8.2. *If $\gamma_x \leq \alpha \leq 50\gamma_x$ then $\mathrm{E}[h(\beta_{x,\alpha})] \in \left(1 \pm \frac{1}{10}\varepsilon\right)\alpha s_x/\mu(x)$. In particular, $\mathrm{E}[h(\beta_{x,\alpha})] \geq \frac{9}{10}$ for $\varepsilon < 1$.*

Proof. Let $\beta_{\mathrm{root}} = 1 - \frac{1}{T+1}$ be the break-even point in the definition of $h(\beta) = \min\left\{\frac{\beta}{1-\beta}, T\right\}$. This is the only non-differentiable point of h in $(0,1)$.

Let $a = \alpha/\gamma_x$, so that $\mathrm{E}[V_{x,\alpha}] = a\mu(x)$ and $\mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right] = a$.

$$\begin{aligned}
0 \le \mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right] - \mathrm{E}\left[h(\beta_{x,\alpha})\right] &\le \mathrm{Ct}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\middle|\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}} > T\right] \\
&= \mathrm{Ct}\left[\frac{\mu(V_x \cap A_\alpha)}{\mu(x)}\middle|\frac{\mu(V_x \cap A_\alpha)}{\mu(x)} > T\right] \\
&= \frac{1}{\mu(x)}\mathrm{Ct}\left[\mu(V_x \cap A_\alpha)|\mu(V_x \cap A_\alpha) > T\mu(x)\right] \\
[\text{Lemma E.9}] &\le \frac{1}{\mu(x)} \cdot \frac{1}{10}\varepsilon a\mu(x) = \frac{1}{10}\varepsilon\,\mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right]
\end{aligned}$$

By Observation 3.24, $\mathrm{E}\left[\frac{\beta_{x,\alpha}}{1-\beta_{x,\alpha}}\right] = \alpha s_x/\mu(x)$, hence $\mathrm{E}[h(\beta_{x,\alpha})] = \left(1 \pm \frac{1}{10}\varepsilon\right)\alpha s_x/\mu(x)$. □

Lemma 8.4. *For $0 < \delta \le \frac{\varepsilon}{21(T+3)}$ and $\hat{\beta} = \beta \pm \delta$, $h(\hat{\beta}) = h(\beta) \pm \max\{2\delta, \frac{1}{20}\varepsilon h(\beta)\}$.*

Proof. For $\beta < \frac{3}{T+1}$, note that $h'(\beta) \le \frac{1}{(1-\beta)^2} \le 2$, hence $h(\beta \pm \delta) = h(\beta) \pm 2\delta$ for $\beta < \frac{2}{T+1}$.

Let $g(\beta) = \ln h(\beta)$. If $\frac{1}{T+1} < \beta < 1 - \frac{1}{T+1}$ then $g'(\beta) = \frac{h'(\beta)}{h(\beta)} = \frac{1}{\beta(1-\beta)} \le \frac{(T+1)^2}{T} \le T + 3$. If $1 - \frac{1}{T+1} < \beta < 1$ then $g'(\beta) = 0$ since h is fixed there. That is, g is $(T+3)$-Lipschitz in $[\frac{1}{T+1}, 1]$.

Hence, for every $\beta_1, \beta_2 \in [\frac{1}{T+1}, 1]$ for which $|\beta_1 - \beta_2| \le \delta$, $|g(\beta_1) - g(\beta_2)| \le (T+3)\delta \le \frac{1}{21}\varepsilon$. This means that $\frac{h(\beta\pm\delta)}{h(\beta)} = e^{\pm\frac{1}{21}\varepsilon} = 1 \pm \frac{1}{20}\varepsilon$ in this range.

Overall, $|h(\beta \pm \delta) - h(\beta)| \le \max\left\{2\delta, \frac{1}{20}\varepsilon h(\beta)\right\}$. □

F Long technical proofs

This paper contains some elementary statements whose proofs were omitted, such as encapsulations of long arithmetic calculations or simple inclusion-exclusions. To make it verifiable, we put here the proofs of these statements.

Lemma 3.9. *For every two distributions μ, τ over Ω there exists a permutation π over Ω for which $d_{\mathrm{TV}}(\mu, \pi\tau) \le 2D_{\mathrm{H}}(\mu;\tau)$.*

Proof. Let $\varepsilon = D_{\mathrm{H}}(\mu;\tau)$ and π be a permutation that realizes this divergence. Let $H = \{x : \mu(x) > (1+\varepsilon)\tau(\pi(x))\}$ and $M = \{x : \tau(x) < \mu(x) \le (1+\varepsilon)\tau(\pi(x))\}$.

$$\begin{aligned}
d_{\mathrm{TV}}(\mu, \pi\tau) &= \sum_{x:\mu(x)>\tau(\pi(x))} (\mu(x) - \tau(\pi(x))) \\
&= \sum_{x\in H} (\mu(x) - \tau(\pi(x))) + \sum_{x\in M} (\mu(x) - \tau(\pi(x))) \\
&\le \sum_{x\in H} \mu(x) + \sum_{x\in M} \varepsilon\tau(\pi(x)) \\
&= \mu(H) + \varepsilon\tau(M) \le \varepsilon + \varepsilon \cdot 1 = 2\varepsilon
\end{aligned}$$

□

Observation 3.32 (Median amplification)**.** *Let X be a random variable, and $[a,b]$ be a range such that $\Pr[X \in [a,b]] \geq 2/3$. We use "median-of-$M$" to denote the process of drawing M independent samples of X and taking their median value. Then:*

(a) Median-of-9 amplifies the probability of obtaining a value in $[a,b]$ to $5/6$.

(b) Median-of-13 amplifies it to $8/9$.

(c) Median-of-47 amplifies it to $99/100$.

(d) Median-of-$\lceil 30 \ln c^{-1} \rceil$ amplifies it to $1 - \frac{1}{2}c$ for $c < 1/3$.

(e) Median-of-$\lceil 30 \ln c^{-1} \rceil$ amplifies it to $1 - \frac{1}{24}c$ for $c < 1/150$.

Proof. Let X_i be the probability of the ith trial to be outside the desired range and $X = \sum_{i=1}^{M} X_i$ be the number of trials outside this range. If $X < \frac{1}{2}k$ then the median is inside the desired range. Hence, the probability that the median is wrong is bounded by $\Pr\left[Y \geq \frac{1}{2}k\right]$, where $Y \sim \mathrm{Bin}(k, 1/3)$.

For parts (a), (b) and (c), we explicitly bound the error probability:

$$\Pr\left[Y \geq \frac{1}{2}k\right] = \sum_{i=\lceil k/2 \rceil}^{k} \binom{k}{i} \left(\frac{1}{3}\right)^i \left(\frac{2}{3}\right)^{k-i} = \frac{1}{3^k} \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{i} 2^i$$

For a separation parameter $1 \leq t \leq \lfloor k/2 \rfloor$, we can bound the last expression using

$$\begin{aligned}
\frac{1}{3^k} \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{i} 2^i &= \frac{1}{3^k} \left(\sum_{i=0}^{t-1} \binom{k}{i} 2^i + 2^t \binom{k}{t} + \sum_{i=t+1}^{\lfloor k/2 \rfloor} \binom{k}{i} 2^i \right) \\
&\leq \frac{1}{3^k} \left(\sum_{i=0}^{t-1} \binom{k}{t-1} 2^i + 2^t \binom{k}{t} + \sum_{i=t+1}^{\lfloor k/2 \rfloor} \binom{k}{i} 2^i \right) \\
&\leq \frac{1}{3^k} \left(2^t \binom{k}{t-1} + 2^t \binom{k}{t} + \sum_{i=t+1}^{\lfloor k/2 \rfloor} \binom{k}{i} 2^i \right) = \frac{1}{3^k} \left(2^t \binom{k+1}{t} + \sum_{i=t+1}^{\lfloor k/2 \rfloor} \binom{k}{i} 2^i \right)
\end{aligned}$$

For part (a) we separate in $t = 3$:

$$\frac{1}{3^9} \sum_{i=0}^{4} \binom{9}{i} 2^i \leq \frac{2^3 \binom{10}{3} + 2^4 \binom{9}{4}}{3^9} = \frac{960 + 2016}{19683} < \frac{1}{6}$$

For part (b) we separate in $t = 5$:

$$\frac{1}{3^{13}} \sum_{i=0}^{6} \binom{13}{i} 2^i \leq \frac{2^5 \binom{14}{5} + 2^6 \binom{13}{6}}{3^{13}} = \frac{64064 + 109824}{1594323} < \frac{1}{9}$$

For part (c) we separate in $t = 23$:

$$\frac{1}{3^{47}} \sum_{i=0}^{23} \binom{47}{i} 2^i \leq \frac{2^{23} \binom{48}{23}}{3^{47}} \leq \frac{(8.3887 \cdot 10^6) \cdot (3.0958 \cdot 10^{13})}{2.6588 \cdot 10^{22}} < \frac{1}{100}$$

For part (d) and part (e) we use Chernoff bound:

$$\Pr[\text{error}] \le \Pr\left[Y \ge \frac{1}{2}k\right] = \Pr\left[\mathrm{Bin}(k,1/3) \ge \frac{1}{3}k + \frac{1}{6}k\right] \le e^{-2(k/6)^2/k} = e^{-k/18}$$

In both parts, $k = \lceil 30 \ln c^{-1} \rceil \ge 30 \ln c^{-1}$, hence $\Pr\left[Y \ge \frac{1}{2}k\right] \le e^{-(30/18)\ln c^{-1}} = e^{-(2/3)\ln c^{-1}} c$.

- Part (d): if $c < 1/3$ then $e^{-(2/3)\ln c^{-1}} < \frac{1}{2}$.
- Part (e): if $c < 1/150$ then $e^{-(2/3)\ln c^{-1}} < \frac{1}{24}$. □

Observation 9.9 (Amplification of testing)**.** *Assume that we have a decision test whose answer is correct with probability at least* $5/8$*. Then the majority answer of* 3 *indpendent trials is correct with probability at least* $2/3$ *and the majority answer of* 45 *independent trials is correct with probability at least* $3/4$*.*

Proof. For $k = 3$:

$$\Pr\left[\mathrm{Bin}\left(k, \frac{5}{8}\right) \le \frac{1}{2}k\right] \le \Pr\left[\mathrm{Bin}\left(3, \frac{5}{8}\right) \le 1\right] = (3/8)^3 + 3 \cdot (5/8) \cdot (3/8)^2 = \frac{162}{512} < \frac{1}{3}$$

For $k \ge 45$:

$$\Pr\left[\mathrm{Bin}\left(k, \frac{5}{8}\right) \le \frac{1}{2}k\right] \le e^{-2\left(\frac{5}{8}-\frac{1}{2}\right)^2 k} = e^{-\frac{1}{32}k} \le e^{-45/32} < \frac{1}{4}$$ □

Lemma 9.25. *For every pair of* c*-truncated functions* $f_\mu, f_\tau : \Omega \to [0,1]$ *with respect to* μ *and* τ*,*

$$d_{\mathrm{TV}}(\mu,\tau) = \frac{1}{2}\left(\mathop{\mathrm{E}}_{x\sim\mu}\left[\max\left\{0, 1 - \frac{f_\tau(x)}{\mu(x)}\right\}\right] + \mathop{\mathrm{E}}_{x\sim\tau}\left[\max\left\{0, 1 - \frac{f_\mu(x)}{\tau(x)}\right\}\right]\right) \pm 2c$$

Proof. In this proof we use the contribution notation of Definition 3.34.

Let:

- $L_\mu = \{x \in \Omega : f_\mu(x) = 0\}$.
- $L_\tau = \{x \in \Omega : f_\tau(x) = 0\}$.
- $H_\mu = \{x \in \Omega : \mu(x) > \tau(x)\} \setminus (L_\mu \cup L_\tau)$.
- $H_\tau = \{x \in \Omega : \tau(x) > \mu(x)\} \setminus (L_\mu \cup L_\tau)$.
- $M = \{x \in \Omega : \tau(x) = \mu(x)\} \setminus (L_\mu \cup L_\tau)$.

Observe that:

$$\begin{aligned}
\mathop{\mathrm{E}}_{x\sim\mu}\left[\max\left\{0, 1 - \frac{f_\tau(x)}{\mu(x)}\right\}\right] &= \mathop{\mathrm{Ct}}_{x\sim\mu}[1|f_\tau(x)=0] + \mathop{\mathrm{Ct}}_{x\sim\mu}\left[\max\left\{0, 1 - \frac{\tau(x)}{\mu(x)}\right\}\middle| f_\tau(x) \ne 0\right] \\
&= \mathop{\mathrm{Ct}}_{x\sim\mu}[1|x \in L_\tau] + \mathop{\mathrm{Ct}}_{x\sim\mu}\left[\max\left\{0, 1 - \frac{\tau(x)}{\mu(x)}\right\}\middle| x \notin L_\tau\right] \\
&= \mu(L_\tau) + \mathop{\mathrm{Ct}}_{x\sim\mu}\left[\max\left\{0, 1 - \frac{\tau(x)}{\mu(x)}\right\}\middle| x \notin L_\tau\right]
\end{aligned}$$

Analogously,

$$\mathop{\mathrm{E}}_{x\sim\tau}\left[\max\left\{0,1-\frac{f_\mu(x)}{\tau(x)}\right\}\right]=\mu(L_\mu)+\mathop{\mathrm{Ct}}_{x\sim\tau}\left[\max\left\{0,1-\frac{\mu(x)}{\tau(x)}\right\}\middle|x\notin L_\mu\right]$$

By definition, $d_{\mathrm{TV}}(\mu,\tau)=\frac{1}{2}\sum_{x\in\Omega}|\mu(x)-\tau(x)|$. We use the partition $\Omega=L_\mu\cup L_\tau\cup H_\mu\cup H_\tau\cup M$.

Set of elements that are negligible in at least one side:

$$\begin{aligned}\sum_{x\in L_\mu\cup L_\tau}|\mu(x)-\tau(x)| &= \sum_{x\in L_\mu}|\mu(x)-\tau(x)|+\sum_{x\in L_\tau}|\mu(x)-\tau(x)|-\sum_{x\in L_\mu\cap L_\tau}|\mu(x)-\tau(x)|\\ &= (\tau(L_\mu)\pm\mu(L_\mu))+(\mu(L_\tau)\pm\tau(L_\tau))\pm(\mu(L_\mu\cap L_\tau)+\tau(L_\mu\cap L_\tau))\\ &= (\tau(L_\mu)\pm c)+(\mu(L_\tau)\pm c)\pm 2c\\ &= \mu(L_\tau)+\tau(L_\mu)\pm 4c\end{aligned}$$

Observe that if B is a set then $\mu(B)=\mathrm{Ct}_{x\sim\mu}[1|x\in B]$. Hence,

$$\begin{aligned}\sum_{x\in L_\mu\cup L_\tau}|\mu(x)-\tau(x)| &= \mathop{\mathrm{Ct}}_{x\sim\mu}[1|x\in L_\tau]+\mathop{\mathrm{Ct}}_{x\sim\tau}[1|x\in L_\mu]\pm 4c\\ [f_\tau(x)=0 \text{ for } x\in L_\tau] &= \mathop{\mathrm{Ct}}_{x\sim\mu}\left[\max\left\{0,1-\frac{f_\tau(x)}{\mu(x)}\right\}\middle|x\in L_\tau\right]\\ [f_\mu(x)=0 \text{ for } x\in L_\mu] &\quad +\mathop{\mathrm{Ct}}_{x\sim\tau}\left[\max\left\{0,1-\frac{f_\tau(x)}{\mu(x)}\right\}\middle|x\in L_\mu\right]\pm 4c\end{aligned}$$

Set of non-negligible elements in one side that are also heavier than in the other side:

$$\begin{aligned}\sum_{x\in H_\mu}|\mu(x)-\tau(x)| &= \sum_{x\in H_\mu}(\mu(x)-\tau(x))\\ &= \sum_{x\in H_\mu}\left(\mu(x)\left(1-\frac{\tau(x)}{\mu(x)}\right)\right)\\ &= \mathop{\mathrm{E}}_{\mu}\left[1_{H_\mu}\cdot\left(1-\frac{\tau(x)}{\mu(x)}\right)\right]\\ &= \mathop{\mathrm{Ct}}_{\mu}\left[1_{H_\mu}\cdot\left(1-\frac{\tau(x)}{\mu(x)}\right)\middle|x\notin L_\tau\right]\\ (*) &= \mathop{\mathrm{Ct}}_{\mu}\left[\max\left\{0,1-\frac{\tau(x)}{\mu(x)}\right\}\middle|x\notin L_\tau\right]\end{aligned}$$

$(*)$: holds since if $x\notin L_\tau, H_\mu$ then $\tau(x)\geq\mu(x)$ and the contribution is zero.

By an analogous analysis,

$$\sum_{x\in H_\tau}|\mu(x)-\tau(x)|=\mathop{\mathrm{Ct}}_{\tau}\left[\max\left\{0,1-\frac{\mu(x)}{\tau(x)}\right\}\middle|x\notin L_\mu\right]$$

For the equal-weights part M, clearly $\sum_{x\in M}|\mu(x)-\tau(x)|=0$.

We sum the partial bounds to obtain:

$$\begin{aligned} 2d_{\mathrm{TV}}(\mu,\tau) &= \mu(L_\tau) + \operatorname*{Ct}_{\mu}\left[\max\left\{0, 1-\frac{\tau(x)}{\mu(x)}\right\}\middle| x \notin L_\tau\right] \\ &\quad +\tau(L_\mu) + \operatorname*{Ct}_{\tau}\left[\max\left\{0, 1-\frac{\mu(x)}{\tau(x)}\right\}\middle| x \notin L_\mu\right] \pm 4c \\ &= \operatorname*{E}_{\mu}\left[\max\left\{0, 1-\frac{f_\tau(x)}{\mu(x)}\right\}\right] + \operatorname*{E}_{\tau}\left[\max\left\{0, 1-\frac{f_\mu(x)}{\tau(x)}\right\}\right] \pm 4c \end{aligned}$$

□

Lemma 10.50. *Consider the sequence where $N_1 = 1$ and for every $i \geq 2$, $N_i = \lceil (1+120\varepsilon)N_{i-1} \rceil$. For every $N \geq 1$, $\varepsilon < \frac{1}{120}$ and $k \leq \ln N/(240\varepsilon)$, $\sum_{i=1}^{k} N_i < \frac{\sqrt{N}\log_2 N}{\varepsilon^2}$.*

Proof. Observe that:

$$\sum_{i=1}^{k} N_i = N_1 + \sum_{i=1}^{k-1} \lceil (1+120\varepsilon)N_i \rceil \leq N_1 + (1+120\varepsilon)\sum_{i=1}^{k-1} N_i + (k-1)$$

Let $M_t = \sum_{i=1}^{t} N_t$. Then $M_1 = N_1$ and $M_t \leq (N_1 + t - 1) + (1+120\varepsilon)M_{t-1}$. By induction,

$$\begin{aligned} M_k &\leq \sum_{i=1}^{k-1} (1+120\varepsilon)^{i-1}(N_1 + k - i) + (1+120\varepsilon)^{k-1} M_1 \\ &\leq (N_1 + k)\sum_{i=1}^{k} (1+120\varepsilon)^{i-1} \\ &\leq (N_1 + k)\frac{(1+120\varepsilon)^k}{120\varepsilon} \leq (N_1 + k)\frac{e^{120\varepsilon k}}{120\varepsilon} \end{aligned}$$

We use $k \leq \frac{\ln N}{240\varepsilon}$ and $N_1 = 1$ to obtain:

$$M_k \leq \left(1 + \frac{\ln N}{240\varepsilon}\right)\frac{e^{\ln N/2+1}}{120\varepsilon} = \frac{e}{\varepsilon^2}\left(\frac{1}{120}\varepsilon + \frac{\ln N}{28800}\right)\sqrt{N} \underset{N\geq 2}{\leq} \frac{\sqrt{N}\ln N}{\varepsilon^2}$$

□

References

[ABDK18] Jayadev Acharya, Arnab Bhattacharyya, Constantinos Daskalakis, and Saravanan Kandasamy. Learning and testing causal models with interventions. *Advances in Neural Information Processing Systems*, 31, 2018.

[ACK15] Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. Adaptive estimation in weighted group testing. In *IEEE International Symposium on Information Theory, ISIT 2015, Hong Kong, China, June 14-19, 2015*, pages 2116–2120. IEEE, 2015.

[ACK18] Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. A chasm between identity and equivalence testing with conditional queries. *Theory Comput.*, 14(1):1–46, 2018.

[AF24] Tomer Adar and Eldar Fischer. Refining the adaptivity notion in the huge object model. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2024.

[AFL24a] Tomer Adar, Eldar Fischer, and Amit Levi. Improved bounds for high-dimensional equivalence and product testing using subcube queries. In *Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques (APPROX/RANDOM)*, 2024.

[AFL24b] Tomer Adar, Eldar Fischer, and Amit Levi. Support testing in the huge object model. In *Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques (APPROX/RANDOM)*, 2024.

[BBC+20] Ivona Bezáková, Antonio Blanca, Zongchen Chen, Daniel Štefankovič, and Eric Vigoda. Lower bounds for testing graphical models: Colorings and antiferromagnetic ising models. *Journal of Machine Learning Research*, 21(25):1–62, 2020.

[BC18] Rishiraj Bhattacharyya and Sourav Chakraborty. Property testing of joint distributions using conditional samples. *ACM Transactions on Computation Theory (TOCT)*, 10(4):1–20, 2018.

[BCG19] Eric Blais, Clément L. Canonne, and Tom Gur. Distribution testing lower bounds via reductions from communication complexity. *ACM Transactions on Computation Theory*, 12(2):1–37, 2019.

[BEFLR20] Omri Ben-Eliezer, Eldar Fischer, Amit Levi, and Ron D Rothblum. Hard properties with (very) short pcpps and their applications. In *11th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 9–1, 2020.

[BFF+01] Tugkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 442–451. IEEE, 2001.

[BFR+00] Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 259–269. IEEE, 2000.

[Can20] Clément L Canonne. A survey on distribution testing: Your data is big. but is it blue? *Theory of Computing*, pages 1–100, 2020.

[CCK+21] Clément L Canonne, Xi Chen, Gautam Kamath, Amit Levi, and Erik Waingarten. Random restrictions of high dimensional distributions and uniformity testing with subcube conditioning. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 321–336. SIAM, 2021.

[CCK24] Diptarka Chakraborty, Sourav Chakraborty, and Gunjan Kumar. Tight lower bound on equivalence testing in conditional sampling model. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4371–4394. SIAM, 2024.

[CCR+25] Deeparnab Chakrabarty, Xi Chen, Simeon Ristic, C Seshadhri, and Erik Waingarten. Monotonicity testing of high-dimensional distributions with subcube conditioning. *arXiv preprint arXiv:2502.16355*, 2025.

[CDKS17] Clément L Canonne, Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Testing bayesian networks. In *Conference on Learning Theory*, pages 370–448. PMLR, 2017.

[CDVV14] Siu-On Chan, Ilias Diakonikolas, Paul Valiant, and Gregory Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1193–1203. SIAM, 2014.

[CFG+23] Sourav Chakraborty, Eldar Fischer, Arijit Ghosh, Gopinath Mishra, and Sayantan Sen. Testing of index-invariant properties in the huge object model. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 3065–3136. PMLR, 2023.

[CFG+24] Sourav Chakraborty, Eldar Fischer, Arijit Ghosh, Amit Levi, Gopinath Mishra, and Sayantan Sen. Testing vs estimation for index-invariant properties in the huge object model. *arXiv preprint arXiv:2412.02235*, 2024.

[CFGM16] Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. *SIAM Journal on Computing*, 45(4):1261–1296, 2016.

[CJLW21] Xi Chen, Rajesh Jayaram, Amit Levi, and Erik Waingarten. Learning and testing junta distributions with sub cube conditioning. In *Conference on Learning Theory*, pages 1060–1113. PMLR, 2021.

[CRS15] Clément L Canonne, Dana Ron, and Rocco A Servedio. Testing probability distributions using conditional samples. *SIAM Journal on Computing*, 44(3):540–616, 2015.

[DDK19] Constantinos Daskalakis, Nishanth Dikkala, and Gautam Kamath. Testing ising models. *IEEE Transactions on Information Theory*, 65(11):6829–6852, 2019.

[DKP19] Ilias Diakonikolas, Daniel M Kane, and John Peebles. Testing identity of multidimensional histograms. In *Conference on Learning Theory*, pages 1107–1131. PMLR, 2019.

[DP17] Constantinos Daskalakis and Qinxuan Pan. Square hellinger subadditivity for bayesian networks and its applications to identity testing. In *Conference on Learning Theory*, pages 697–703. PMLR, 2017.

[FF06] Eldar Fischer and Lance Fortnow. Tolerant versus intolerant testing for boolean properties. *TOC*, 2(9):173–183, 2006.

[FJO+15] Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Faster algorithms for testing under conditional sampling. In *Conference on Learning Theory*, pages 607–636. PMLR, 2015.

[FLV19] Eldar Fischer, Oded Lachish, and Yadu Vasudev. Improving and extending the testing of distributions for shape-restricted properties. *Algorithmica*, 81:3765–3802, 2019.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[GLP18] Reza Gheissari, Eyal Lubetzky, and Yuval Peres. Concentration inequalities for polynomials of contracting ising models. *Electronic Communications in Probability*, 2018.

[GR11] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 68–75, 2011.

[GR23] Oded Goldreich and Dana Ron. Testing distributions of huge objects. *TheoretiCS*, 2, 2023.

[GTZ17] Themistoklis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Faster sublinear algorithms using conditional sampling. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA '2017)*, 2017.

[GTZ18] Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Certified computation from unreliable datasets. In *Conference On Learning Theory*, pages 3271–3294. PMLR, 2018.

[JHW18] Jiantao Jiao, Yanjun Han, and Tsachy Weissman. Minimax estimation of the L_1 distance. *IEEE Transactions on Information Theory*, 64(10):6672–6706, 2018.

[KT19] Guatam Kamath and Christos Tzamos. Anaconda: a non-adaptive conditional sampling algorithm for distribution testing. In *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms (SODA '2019)*, 2019.

[MKP25] Kuldeep S Meel, Gunjan Kumar, and Yash Pote. Distance estimation for high-dimensional discrete distributions. In *The 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2025.

[Nar20] Shyam Narayanan. On distribution testing in the conditional sampling model. *arXiv preprint arXiv:2007.09895*, 2020.

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

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252—-271, 1996.

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

[Val08] Paul Valiant. Testing symmetric properties of distributions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 383–392, 2008.

[VV10a] Gregory Valiant and Paul Valiant. A clt and tight lower bounds for estimating entropy. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 17, page 9, 2010.

[VV10b] Gregory Valiant and Paul Valiant. Estimating the unseen: A sublinear-sample canonical estimator of distributions. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 17, page 9, 2010.

[VV11] Gregory Valiant and Paul Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new clts. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 685–694, 2011.

[Yao77] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 222–227, 1977.