# Interactive Proofs For Distribution Testing With Conditional Oracles

Ari Biswas
University Of Warwick, United Kingdom

Mark Bun[1]
Boston University, USA

Clément L. Canonne[2]
University of Sydney, Australia

Satchit Sivakumar[3]
Boston University, USA

## Abstract

We revisit the framework of interactive proofs for distribution testing, first introduced by Chiesa and Gur (ITCS 2018), which has recently experienced a surge in interest, accompanied by notable progress (e.g., Herman and Rothblum, STOC 2022, FOCS 2023; Herman, RANDOM 2024). In this model, a data-poor verifier determines whether a probability distribution has a property of interest by interacting with an all-powerful, data-rich but untrusted prover bent on convincing them that it has the property. While prior work gave sample-, time-, and communication-efficient protocols for testing and estimating a range of distribution properties, they all suffer from an inherent issue: for most interesting properties of distributions over a domain of size $N$, the verifier must draw at least $\Omega(\sqrt{N})$ samples of its own. While sublinear in $N$, this is still prohibitive for large domains encountered in practice.

In this work, we circumvent this limitation by augmenting the verifier with the ability to perform an exponentially smaller number of more powerful (but reasonable) *pairwise conditional* queries, effectively enabling them to perform "local comparison checks" of the prover's claims. We systematically investigate the landscape of interactive proofs in this new setting, giving polylogarithmic query and sample protocols for (tolerantly) testing all *label-invariant* properties, thus demonstrating exponential savings without compromising on communication, for this large and fundamental class of testing tasks.

**2012 ACM Subject Classification** Theory of computation → Interactive computation

**Keywords and phrases** Distribution Testing, Interactive Proofs

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2026.8

**Related Version** Full Version
*Full Version*: `opt. text shown instead of the URL`

**Funding** *Mark Bun*: [funding]
*Clément L. Canonne*: Supported by an ARC DECRA (DE230101329).
*Satchit Sivakumar*: [funding]

**Acknowledgements** This work was started while AB and CC were visiting the Simons Institute for the Theory of Computing as part of the Sublinear Algorithms program.

[1] Optional footnote, e.g. to mark corresponding author
[2] Optional footnote, e.g. to mark corresponding author
[3] Optional footnote, e.g. to mark corresponding author

17th Innovations in Theoretical Computer Science Conference (ITCS 2026).
Editor: Shubhangi Saraf; Article No. 8; pp. 8:1–8:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

# 1 Introduction

Distribution testing, as introduced by Batu et al. [2000], is a mature subfield of property testing [Goldreich et al., 1998, Rubinfeld and Sudan, 1996] aimed at investigating statistical properties of an unknown distribution given sample access to it. Given a property (a set of distributions) and a proximity parameter $\tau \in (0, 0.1]$, distribution testing algorithms output Accept if the distribution is in the property (or close to it), or Reject if the distribution is $\tau$-far from the property, both with high probability. Closeness and farness are quantified with respect to a prespecified notion of distance, typically total variation distance. The primary motivation behind distribution testing is to design testing algorithms for deciding properties with sample complexity sub-linear in the domain size $N$ (which is demonstrably more efficient than learning the distribution, which requires drawing $\Theta(N)$ samples). Accordingly, over the last two decades, researchers have extensively studied the sample complexity of numerous distribution properties, such as simple uniformity testing [Goldreich and Ron, 2011] (testing whether a distribution is uniform over its entire domain), support size decision problem [Raskhodnikova et al., 2009, Valiant and Valiant, 2011, Wu and Yang, 2019, Ferreira Pinto Jr. and Harms, 2025] (testing whether a distribution's support is within some pre-specified range), and many more: see, e.g., [Goldreich, 2017, Chapter 11] and Rubinfeld [2012], Canonne [2020, 2022] for a more thorough introduction to distribution testing. Unfortunately, although distribution testing is often more efficient than learning the distribution, it is still prohibitively expensive for practical use. For example, it is known that generalized uniformity testing (testing whether a distribution is uniform over its support) over a domain of size $N$ requires $\Omega\left(N^{2/3}\right)$ samples [Batu and Canonne, 2017, Diakonikolas et al., 2018], which can be impractical for large domain sizes. Even simple uniformity testing requires $\Omega(\sqrt{N})$ samples [Paninski, 2008], and its *tolerant* testing version (which asks to distinguish distributions *close* to uniform from those which are far) needs $\Omega(N/\log N)$ samples [Valiant and Valiant, 2017].

In the face of these limitations, a nascent line of work [Chiesa and Gur, 2018, Herman and Rothblum, 2022, 2023, Herman, 2024] has asked a related question: *with testing being hard by itself, what is the complexity of* verifying *the properties of a distribution given sample access to it?* Here, in addition to drawing samples from the distribution, the tester is allowed to interactively communicate with an omniscient but *untrusted* prover that knows the distribution in its entirety. The idea here is to leverage the provers extra knowledge about the distribution, with the hope that checking the provers' claims is easier than naively testing the property. While this model of verifiable computation has only recently been explored in the context of distribution testing, it has been an active area of research in other areas of theoretical computer science for over 40 years (see for e.g. [Goldwasser et al., 1985, Micali, 2000, Rothblum et al., 2013, Goldwasser et al., 2015, Berman et al., 2018, Arun et al., 2024]). It models settings where a centralized organization (for example, a company turning billions of dollars of profit) has the ability to collect large amounts of data and learn distributions to high precision, while end-users may not have the same ability. At the same time, the company might have incentives to lie, and so verifying whether the company is being truthful is important in this setting. The work of Chiesa and Gur [2018] shows that the verification of *any* distribution property over domain $[N]$ can be reduced to identity testing[4], with communication *superlinear* in the domain size. Follow up work [Herman and

[4] Identity testing refers to the task of testing distinguishing between distributions that are exactly equal to a pre-specified reference distribution from distributions $\tau$-far from in total variation distance.

Rothblum, 2022, 2023] recovers this result for the broad class of *label-invariant properties*, while only requiring communication *sub-linear* in the domain size. More specifically, the work of Herman and Rothblum [2022, 2023] show that for label-invariant properties, verification requires only $O\left(\sqrt{N}\right)$ samples and $\widetilde{O}\left(\sqrt{N}\right)$ communication, even though, as mentioned earlier, testing some properties in this class could require $\Theta(N/\log N)$ samples. Here, a property is *label-invariant* (also known as *symmetric*) if the names of the elements themselves are not significant to the decision outcome. Testing if a distribution is uniform over its support (also known as generalized uniformity testing) is an example of a label-invariant property.

Unfortunately, while a significant improvement over unaided testing, requiring $O\left(\sqrt{N}\right)$ samples from the verifier can still be prohibitive when considering massive domains. Further, there is a matching sample complexity lower bound – verification of even basic label-invariant properties such as checking if a distribution is uniform over its entire domain requires $\Omega(\sqrt{N})$ samples[5]. To summarise: For most properties, with access to *only* samples from a distribution, it is impossible for any tester to do better than drawing $\Omega\left(\sqrt{N}\right)$ samples, with or without the help of a prover. To bypass these limitations and develop more practical algorithms, in this work we study verifiers that can make a very small number of calls to a more powerful *conditional sampling* oracle. These oracles were introduced in the context of distribution testing [Chakraborty et al., 2013, Canonne et al., 2014]; allowing the tester to condition that samples from the oracle come from a subset $S$ of the domain, of their choosing. The oracle responds with a sample with probability re-normalised over $S$. If no element in $S$ is supported, the oracle responds with FAIL. Since specifying an arbitrary set may considered be unrealistic for practical purposes, a commonly studied restriction is the pairwise conditional sampling model (`PCond`), where the specified sets are restricted to be of size exactly 2 or the entire domain (thus, just a regular sample from the distribution). These oracles can be thought of as allowing for local comparisons between the probabilities of two points. While access to a `PCond` oracle can be significantly helpful for problems like simple uniformity testing, it is unclear from prior work whether it results in more efficient testing for the general class of label-invariant properties. Verification with access to a `PCond` oracle (or any type of conditional sampling oracle) has also, to the best of our knowledge, not been explored. In our quest to find practical algorithms that work for large domains, we thus ask the following question.

*Can label-invariant properties be verified in a (query, sample and communication)-efficient way when the tester has access to a `PCond` oracle?*

## 1.1 Our Results

Our main result is an *exponential* query complexity separation between testing and verification for testing label-invariant properties with access to a `PCond` oracle. A detailed accounting of our results and comparison to existing work can be found in Table 1. A description follows.

One might have initially hoped that the power of a `PCond` oracle allows us to test label-invariant properties efficiently, even without the help of a prover. Indeed, with access to the full power of the `Cond` model (where arbitrary subsets $S$ can be queried and a sample

[5] This lower bound applies to any property that is a singleton set.

| | Query Complexity Without Prover | Query Complexity With Prover | Communication | Rounds |
|---|---|---|---|---|
| Samp | $\widetilde{\Omega}\left(\frac{N}{\log N}\right)$ Valiant and Valiant [2017] | $\widetilde{O}\left(\sqrt{N}\right)$ [Herman and Rothblum, 2022, 2023] | $\widetilde{O}\left(\sqrt{N}\right)$ [Herman and Rothblum, 2022, 2023] | 2 [Herman and Rothblum, 2022, 2023] |
| PCond | $\boldsymbol{\Omega\left(N^{1/3}\right)}$ | $\mathrm{poly}(\boldsymbol{\log N, \frac{1}{\tau}})$ | $\boldsymbol{\widetilde{O}\left(\sqrt{N}\right)}$ | $\mathrm{poly}(\boldsymbol{\log N, \frac{1}{\tau}})$ |

**Table 1** Results on testing and verifying label-invariant properties under different types of access to the distribution. We state the best known lower bounds for label invariant properties. For Samp the lower bound is for entropy estimation, whereas for PCond it is the support size decision problem described in Section 3 of the full version. The upper bounds apply for all label-invariant properties. Our results are highlighted in bold.

conditional on $S$ is obtained), Chakraborty et al. [2013] show that this class of properties over a domain of size $N$ can be tested with $O(\mathrm{poly}\log N)$ queries to the Cond oracle. Our first result dashes this hope – we show a lower bound on the number of PCond queries required to test label-invariant properties with constant proximity parameter $\tau$, demonstrating that the PCond oracle is not much better in the worst case than the sampling oracle for this class of properties. Specifically, we show that a simple variant of the support size distinguishing problem for distributions over a domain of size $N$ requires $\Omega(N^{1/3})$ queries to a PCond oracle (the exact same as with access to only a sampling oracle). Thus, unaided, there exist (label-invariant) properties for which the PCond oracle is not much better than just sample access.

▶ **Theorem 1.1** (Informal Version). *There exists a label-invariant property $\Pi$ such that every tester with access to a PCond oracle for $\Pi$ with proximity parameter $\tau \leq 1/2$ and failure probability $0.01$ must make $\Omega\left(N^{1/3}\right)$ queries.*

The above lower bound motivates the investigation of verification with access to a PCond oracle. As mentioned earlier, [Chiesa and Gur, 2018, Proposition 3.4] showed that with super-linear communication complexity, there exists a reduction from verification to identity testing. Instantiating this reduction with an identity tester using PCond oracles [Narayanan, 2021, Theorem 1.5], we get that there exists an interactive proof system for every property with super-linear communication complexity that makes only $O(\sqrt{\log N}/\tau^2)$ queries to the PCond oracle. However, super-linear communication is also prohibitive for practical algorithms; the proof systems by Herman and Rothblum [2022, 2023] require the prover to only communicate $\widetilde{O}(\sqrt{N})$ domain elements, but still achieve the sample complexity of identity testing (for the class of label-invariant properties). Could we also hope to achieve such communication while maintaining similar query complexity as that of identity testing? The main result of this paper is an affirmative answer to this question. Specifically, we give an interactive proof system for tolerantly verifying *any* label-invariant property that has communication complexity $\widetilde{O}(\sqrt{N})$ and query complexity $\mathrm{poly}(\log N)$ (suppressing the dependence on the proximity parameter).

▶ **Theorem 1.2** (Informal Label-Invariant Tolerant Verification Theorem ). *Fix a label-invariant property $\Pi$ over a domain $[N]$ and proximity parameters $\tau_c, \tau_f \in (0, 1/2]$. There exists a polylogarithmic (in $N$) round interactive protocol $\Pi$ between an honest verifier* T*, and an* *omniscient untrusted prover* $\mathsf{P}^{\mathcal{D}}$*, where the verifier has* **PCond** *access to* $\mathcal{D}$*, such that at the end of the interaction the verifier satisfies the following conditions:*

1. ***Completeness:*** *If the prover follows the protocol as prescribed, and* $d_{TV}(D, \Pi) \leq \tau_c$*, then*

$$\Pr\left[out\left[\Pi\left(\mathsf{P}^{\mathcal{D}}, \mathsf{T}^{(\mathcal{D})}; \tau_c, N\right)\right] = \mathsf{Accept}\right] \geq 2/3$$

2. ***Soundness:*** *If* $d_{TV}(D, \Pi) \geq \tau_f$*, then for any prover* $\widetilde{\mathsf{P}^{\mathcal{D}}}$

$$\Pr\left[out\left[\Pi\left(\widetilde{\mathsf{P}}^{\mathcal{D}}, \mathsf{T}^{(\mathcal{D})}; \tau_c, N\right)\right] = \mathsf{Reject}\right] \geq 2/3$$

*The complexity of the verifier is as follows:*

1. ***Query Complexity + Sample Complexity****:* $O\left(\mathrm{poly}(\log N, 1/(\tau_f - \tau_c))\right)$
2. ***Communication Complexity:*** $\widetilde{O}\left(\sqrt{N}\,\mathrm{poly}(1/(\tau_f - \tau_c))\right)$

In the process of proving this result, we give protocols for more basic primitives that may be of independent interest. Most significantly, we give an interactive proof system that is able to calculate the approximate probability mass of any point[6] in the domain using communication complexity $\widetilde{O}(\sqrt{N})$ and query complexity $O(\mathrm{poly}(\log N, 1/\tau))$. As we will explain in the techniques section to follow, this is a key technical workhorse in our protocol for verifying label-invariant properties.

▶ **Theorem 1.3** (Informal Version). *Fix a domain* $[N]$*. For every* $\tau > 0$ *and* $\delta \in (0, 1/2)$*, there exists a* $O(\ \mathrm{poly}(\log N, \log 1/\delta, \frac{1}{\tau}))$*-round interactive proof system such that the verifier* $\mathsf{T}$ *with access to a* **PCond** *oracle satisfies the following.*

1. ***Completeness:*** *For every distribution* $\mathcal{D}$*, if the prover* $\mathsf{P}^{\mathcal{D}}$ *is honest, then*

$$\Pr\left[\mathsf{T}\ outputs\ (y^\star, \widetilde{\mathcal{D}}[y^\star])\ s.t\ \frac{\widetilde{\mathcal{D}}[y^\star]}{\mathcal{D}[y^\star]} \in \left[\frac{1}{(1+\tau)}, 1+\tau\right]\right] \geq 1 - \delta$$

2. ***Soundness:*** *For any cheating prover* $\widetilde{\mathsf{P}}^{\mathcal{D}}$*, then*

$$\Pr\left[\mathsf{T}\ outputs\ \mathsf{Reject} \vee \mathsf{T}\ outputs\ (y^\star, \widetilde{\mathcal{D}}[y^\star])\ s.t\ \frac{\widetilde{\mathcal{D}}[y^\star]}{\mathcal{D}[y^\star]} \in [1/(1+\tau)^2, (1+\tau)^2]\right] \geq 1 - \delta$$

*The complexity of the verifier is as follows:*

1. ***Query Complexity + Sample Complexity****:* $O\left(\mathrm{poly}(\log N, 1/(\tau))\right)$
2. ***Communication Complexity:*** $\widetilde{O}\left(\sqrt{N}\,\mathrm{poly}(1/(\tau))\right)$

## 1.2 Technical Overview

In this section, we give an overview of our protocol for verifying label-invariant properties.

[6] Provided it does not have prohibitively small probability mass in its neighbourhood.

### Unlabelled Bucket Histogram:

There is now a long line of work on the testing and verification of label-invariant properties [Batu et al., 2000, Valiant, 2008, Chakraborty et al., 2013, Herman and Rothblum, 2022, 2023], and a key object used in this work is the unlabelled *approximate* $\tau$-bucket histogram of a distribution. Bucketing corresponds to partitioning the interval $[0,1]$ into smaller multiplicative probability intervals. The $\tau$-bucket histogram divides the interval $[0,1]$ into $\widetilde{O}\left(\log N/\tau\right)$ buckets where the $\ell^{\text{th}}$ bucket is a set of domain elements with individual probability mass in the range $(\tau(1+\tau)^{\ell}/N, \tau(1+\tau)^{\ell+1}/N]$. The *approximate* unlabelled bucket histogram of a distribution then corresponds to a list of $\widetilde{O}(\log N/\tau)$ fractions, where the $\ell^{\text{th}}$ element of the list is the fraction of domain points whose probability lies in the range specified by the $\ell^{\text{th}}$ bucket. It is well known (see [Valiant, 2008]) that the *approximate* unlabelled $\tau$-bucket histogram of a distribution is a sufficient statistic to (tolerantly) test any label-invariant property with proximity parameter(s) $O\left(\tau\right)$. Thus, similar to prior work [Herman and Rothblum, 2022, 2023, Herman, 2024], our protocol also focuses on efficiently verifying an unlabelled $\tau$-bucket histogram given to us by the prover.

### Using Pairwise Comparisons to Learn Bucket Histogram:

Note that the unlabelled bucket histogram is a distribution over the buckets of the $\tau$-bucket histogram and is hence a distribution over a domain of size $\widetilde{O}(\log \frac{N}{\tau})$. Hence, by standard results in distribution learning, $\widetilde{O}(\log N/\tau^2)$ samples from this bucket distribution would be sufficient to learn it. However, sampling from this bucket distribution is non-trivial since a sample from the original distribution $\mathcal{D}$ does not come with information about histogram bucket index.

While sampling from the bucket distribution might be hard with `Samp` access to the distribution, one might be more optimistic about the possibility of sampling from the $\tau$-approximate histogram with `PCond` queries. In particular, one approach that we might take is the following: the verifier draws a dataset of size $\widetilde{O}(\log N/\tau^2)$ (enough to learn the histogram), and sends these samples to the prover, who responds with the bucket index of each sample. From how a $\tau$-histogram is defined, if $x$ and $y$ belong to buckets $i$ and $j$ respectively, then this implies that the ratio of the probability masses of $x$ and $y$ under $\mathcal{D}$ is guaranteed to be in the interval $\left[\frac{1}{(1+\tau)^{|j-i|}}, (1+\tau)^{|j-i|}\right]$. As `PCond` access allows us to conditionally sample from a set restricted to two domain elements, it allows us to approximately learn the ratio of their probability masses up to a multiplicative constant. Equipped with this power, for each pair $x \neq y$ from the set of drawn samples, the verifier uses the `PCond` oracle to check if the learned ratios align with the provers claims. If the prover were to lie significantly, then for at least one pair of samples, the claimed ratio would significantly different from the learned ratio. Unfortunately, this simplistic strategy comes with two pitfalls. Firstly, assuming the above strategy was sound, naively comparing elements with arbitrary bucket indices could require $\Omega\left(N\right)$ `PCond` queries if the elements being compared had significantly different probability masses. This would be as bad as learning the distribution itself. Secondly, and more importantly, the above strategy is *not* sound. It does not catch a prover that "slides" all samples into different buckets *in the same way*, i.e., it lies about *every* bucket index by the same offset. As an example, consider two distributions: $\mathcal{D}_1$ is the uniform distribution over $N^{1/4}$ domain elements and $\mathcal{D}_2$ is the uniform distribution over $\sqrt{N}$ domain elements. The bucket histograms of both involve a unit mass on a single (but different) bucket. However, pairwise comparisons between samples taken from either distribution will always reveal a ratio that is approximately 1, since the probabilities of all elements in the support of both distributions are identical. Hence, given distribution $\mathcal{D}_1$, the prover can output the bucket histogram of distribution $\mathcal{D}_2$, and it is impossible for a verifier to catch it purely by using the test described above.

A remedy to these obstacles lies in the following observation. If we had a good estimate of the probability mass of *a single point* $y$ in the domain, then we could resolve the soundness issue discussed above. We simply use the `PCond` oracle to learn the ratio of probabilities between each of our samples and $y$. Using this ratio, and knowledge of the approximate mass of $y$, we can compute estimates of the probabilities of all samples. This would squash the sliding attack described above. To deal with the first issue (that of the probability mass of $y$ being very far from that of a sample), we would need to do more than learn just one value of $y$. Instead, we could learn the mass of a point $y_j$, for every bucket $j$ that has large enough mass. This way, for any sample $x$ in the tester's set of samples (which are likely to come from buckets with sufficiently large mass), we can find some $y_j$ that is in a near-by bucket with high probability.

### Verifying the probability of points:

Given that we simply need to identify the probability of a few points in the domain, one might expect that this could be done even without access to a prover — this would give a query-efficient tester for label-invariant properties. However, this ends up being a surprisingly challenging task. Indeed, our lower bound in Theorem 1.1 shows that this is impossible (if we wanted to bypass the $\text{poly}(N)$ lower bound). This indicates a power of an untrusted prover; it is able to certify the probability of a few points in the distribution support. Indeed, the proof system with super-linear communication Chiesa and Gur [2018] achieves something stronger- it certifies the entire distribution. Since we only need to certify the mass of at most $O(\log N)$ points, we ask if this be done in a more communication efficient way?

### Support Size Verification:

Inspired by the "sliding" cheating prover from earlier, we consider the orthogonal but related problem of verifying the support size of a flat distribution.[7] We will subsequently show that a protocol for this problem can be combined with the ability of a `PCond` oracle to learn neighbourhoods around points, enabling us to solve the probability approximation problem for "relevant" domain elements.

Given a support size claim represented by four numbers $A', A, B, B'$, corresponding to the claim that $A' < A \leq \text{Supp}(\mathcal{D}) \leq B < B'$, our hope is to accept if the claimed support size range is accurate, and reject if the true support is larger than $B'$ or smaller than $A'$. Given different values of $A$ and $B$, we develop a number of tests to verify with *sub-linear* communication, the support size assuming the distribution is uniform over its support[8]. We summarize the ideas below.

If the claimed support size upper bound $B$ is small (that is, $O\left(\sqrt{N}\right)$), we could ask the prover to send us the claimed support of the distribution. If the prover lies, and the true support is actually much larger, then taking a few samples from the distribution would give us a domain element outside the claimed support, thus catching the lie of the prover. If the true support is much smaller than $A$, then taking a number of uniform samples from

[7] A distribution is *flat* if it is uniform over its support.

[8] We relax this condition in the main protocol, but assuming uniformity makes the description more intuitive.

the claimed support sent by the prover would result in a sample outside the support of the distribution, which could be easily detected with a few PCond queries. This gives us a protocol with a constant number of queries, and communication complexity roughly $O(B)$.

On the other hand, if the claimed support size lower bound $A$ is large (that is, $\omega\left(\sqrt{N}\right)$), then asking the prover to send the support is not communication-efficient. Our approach instead involves using uniform samples from the domain. The first test is to catch provers that lie that the support is much larger than it really is. It involves drawing $O(N/A)$ uniform samples $S_1$ from the domain, and sending them to the prover. We ask the prover to send back a sample in this subset that is in the support of the distribution. If the true support is much smaller than $A$, there are (with high probability) no samples in the support of the distribution in $S_1$, and we can ensure the prover does not cheat by checking whether any element it sends back is in the support using a constant number of PCond queries. The second test catches provers that lie that the support is much smaller than it really is. It involves drawing $O(N/B')$ samples $z_1, \ldots, z_m$ uniformly from the domain and permuting them with one sample $x$ taken from the distribution. We ask the prover to identify the index of $x$ in the permuted set. If the true support is smaller than $B$, then w.h.p, we expect that none of $z_1, \ldots, z_s$ are in the support of the distribution and hence, the honest prover can identify $x$ exactly. On the other hand, if the support is larger than $B'$, we expect that at least one of $z_1, \ldots, z_s$ is in the support of the distribution, and the prover is unable to tell what the sample inserted by the prover was (since it could be $x$ or $z_i$). This gives us a protocol with a constant number of queries, and communication complexity roughly $O(N/A)$. Balancing parameters in these tests to optimize the communication complexity, we get an overall protocol for support size verification with communication complexity roughly $\sqrt{N}$.

We emphasize that the above outline is a simplification of the truth, and sweeps important details under the rug. Recall that our main goal is to certify the histogram of a distribution. In an attempt to do so, we will use the support size protocol above repeatedly as a sub-routine. This requires bounding the soundness and completeness errors in a meaningful way. As described above, these protocols do not have low enough soundness and completeness error to be used as sub-routines. Additionally, the above description assumes that distributions are exactly flat. In practice this will not be the case, and we need to be able to handle distributions where the probability ratio between any two elements in the support is upper bounded by some constant $\alpha$ (nearly flat). We show how to analyse the protocols above to facilitate amplification of soundness and completeness errors, and make the protocol work for the more general class of $\alpha$-flat distributions.

### Estimating Probability of a Point using Support Size Verification:

Finally, we explain how we can use the support size protocols to estimate the probability of a point. Prior work [Canonne et al., 2015] using the PCond oracle has shown that it can be used to estimate the mass within a multiplicative $(1+\tau)$-neighbourhood of any point[9]. We ask the prover to send us a point $y^\star$ [10] with sufficiently large mass in its neighbourhood (by an averaging argument, at least one histogram bucket needs to have $\tau/\log N$ mass, ensuring that such a point exists, and tell us the bucket the $y^\star$ belongs to. We then use the PCond oracle to estimate the mass within the multiplicative neighbourhood of $y^\star$. The learned

[9] As long as the point does not have prohibitively small probability mass under the distribution.

[10] Recall that in the final protocol, we will ask the prover to send us points from every bucket with sufficiently large mass. For simplicity, we consider a single point in this description.

mass of the neighborhood divided by the prover's claimed probability mass[11] of $y^\star$ gives us bounds on the number of elements in $y^\star$'s neighbourhood. Additionally, by definition the neighbourhood of $y^\star$ will be nearly flat. Hence, we have reduced the problem to a claim about the support size of a nearly-uniform distribution over a subset of the domain. Observe that we can sample from the distribution restricted to this bucket using PCond queries—since there is sufficient mass in the neighbourhood of the point, $O(\text{poly}\log N)$ samples from the distribution will contain at least one sample from the bucket, and we can use PCond queries between the samples and $y^\star$ to find out which sample it is. If the prover was telling the truth (or rather did not lie egregiously), then the support size claim holds and the verifier will accept, thereby giving us a point and its approximate probability mass[12]. If the prover significantly lied, then the support size claim will be false and the prover will be caught. We note that the final analysis needs to handle some additional subtleties, since the PCond oracle is itself only approximate and does not provide perfect comparisons (which can affect the sampling), and additionally the bucket boundaries and the neighbourhood of the provided point do not overlap precisely. Once we have estimated the probability of important points, we can use the ratio learning techniques discussed earlier to certify the prover's claim about the bucket histogram of the distribution.

## 1.3 Other Related Work:

### Interactive Proofs for Distribution Testing:

As mentioned earlier, interactive proofs for verifying distribution properties were first introduced in the work of Chiesa and Gur [2018]. Follow-up work [Herman and Rothblum, 2022, 2023] studied interactive proofs for verifying *label-invariant* properties, focusing on sublinear communication, and doubly efficient protocols (i.e. computationally-efficient and sample-efficient generation of a proof). Herman [2024] studied *public-coin* interactive proofs for testing label-invariant properties, where the verifier has no private randomness. Herman and Rothblum [2024] gives communication-efficient and sample-efficient interactive proofs for more general distribution properties that can be decided by uniform polynomial-size circuits with bounded depth. Herman and Rothblum [2025] introduces *computationally sound* interactive proofs and shows communication-, time-, and sample-efficient protocols for any distribution property that can be decided in polynomial time given explicit access to the full list of distributiom probabilities. All of the above works assume that the verifier only has sample access to the distribution, and the verifier sample complexity in all of them is $\Omega(\sqrt{N})$ (where $N$ is the size of the domain).

### Interactive Proofs for Learning:

A related but orthogonal line of work [Goldwasser et al., 2021, Mutreja and Shafer, 2023, Gur et al., 2024, Caro et al., 2024b,a] focuses on interactive proofs for verifying learning problems- for a specific hypothesis class $H$, given access to an untrusted prover, the goal is for a verifier to output an accurate hypothesis from $H$ for an underlying unknown distribution $D$ if the resource-rich prover is honest, and to reject if the prover lies egregiously. Different types of resource asymmetry between the prover and the verifier are explored in these papers –

[11] The bucket index gives a lower and upper bound on the true probability mass of $y^\star$. This is sufficient to catch a cheating prover.

[12] The claimed mass of $y^\star$ is derived from the bucket sent by the prover.

including differing number of samples, different computational complexities, different types of access to the underlying function (sample vs query), and differing access to computational resources (classical vs quantum computation and communication).

### Distribution testing under conditional oracles:

Our work focuses on interactive proofs for *verifying* distribution properties under conditional sampling models. There is a long line of work on *testing* with access to conditional samples. Chakraborty et al. [2013], Canonne et al. [2014] introduced the conditional sampling (`Cond`) model and its more restricted variants (`PCond`, `ICond`). They gave algorithms for uniformity testing, tolerant uniformity testing, identity testing etc. in these conditional sampling models with query and sample complexity significantly better than the `Samp` model. Follow-up work shows improved bounds for identity testing (and its tolerant version), tolerant uniformity testing, and new algorithms for other tasks such as equivalence testing and support size problem in the `Cond` model [Falahatgar et al., 2015, Narayanan, 2021, Chakraborty et al., 2023, 2024]. There is also a line of work studying the power of non-adaptive queries in the conditional sampling model [Acharya et al., 2015, Kamath and Tzamos, 2019]. The `PCond` model was studied in detail by Narayanan [2021] who gave optimal bounds for identity testing and tolerant uniformity testing in this model, improving on results from Canonne et al. [2015]. Testing under other types of conditional sampling has also been studied in the literature including subcube conditioning, where the distribution is supported on the hypercube, and the tester is allowed to ask for conditional samples from subcubes [Bhattacharyya and Chakraborty, 2018, Canonne et al., 2021, Chen et al., 2021, Kumar et al., 2023, Chakrabarty et al., 2025], and coordinate conditional sampling [Blanca et al., 2025] (a version of subcube conditioning where all but one coordinate is fixed to a specific configuration and a sample is obtained from the remaining coordinate). Testing under other types of access to the distribution such as Probability Mass Function queries or Cumulative Distribution Function queries has also been studied in the literature [Batu et al., 2002, Rubinfeld and Servedio, 2005, Guha et al., 2009, Canonne and Rubinfeld, 2014, Onak and Sun, 2018].

## References

J. Acharya, C. L. Canonne, and G. Kamath. A chasm between identity and equivalence testing with conditional queries. In N. Garg, K. Jansen, A. Rao, and J. D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPIcs*, pages 449–466. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi: 10.4230/LIPICS.APPROX-RANDOM.2015.449. URL https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2015.449.

A. Arun, S. Setty, and J. Thaler. Jolt: Snarks for virtual machines via lookups. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–33. Springer, 2024. URL https://eprint.iacr.org/2023/1217.

T. Batu and C. L. Canonne. Generalized uniformity testing. In *FOCS*, pages 880–889. IEEE Computer Society, 2017. URL https://arxiv.org/abs/1708.04696.

T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, pages 259–269. IEEE Comput. Soc. Press, Los Alamitos, CA, 2000. doi: 10.1109/SFCS.2000.892113. URL https://doi.org/10.1109/SFCS.2000.892113.

T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld. The complexity of approximating entropy. In J. H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 678–687. ACM, 2002. doi: 10.1145/509907.510005. URL https://doi.org/10.1145/509907.510005.

I. Berman, R. D. Rothblum, and V. Vaikuntanathan. Zero-knowledge proofs of proximity. In A. R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 19:1–19:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi: 10.4230/LIPICS.ITCS.2018.19. URL https://doi.org/10.4230/LIPIcs.ITCS.2018.19.

R. Bhattacharyya and S. Chakraborty. Property testing of joint distributions using conditional samples. *ACM Trans. Comput. Theory*, 10(4):16:1–16:20, 2018. doi: 10.1145/3241377. URL https://doi.org/10.1145/3241377.

A. Blanca, Z. Chen, D. Stefankovic, and E. Vigoda. Complexity of high-dimensional identity testing with coordinate conditional sampling. *ACM Trans. Algorithms*, 21(1):7:1–7:58, 2025. doi: 10.1145/3686799. URL https://doi.org/10.1145/3686799.

C. L. Canonne. *A Survey on Distribution Testing: Your Data is Big. But is it Blue?* Number 9 in Graduate Surveys. Theory of Computing Library, 2020. doi: 10.4086/toc.gs.2020.009. URL http://www.theoryofcomputing.org/library.html.

C. L. Canonne. Topics and techniques in distribution testing: A biased but representative sample. *Foundations and Trends® in Communications and Information Theory*, 19(6): 1032–1198, 2022. ISSN 1567-2190. doi: 10.1561/0100000114. URL http://dx.doi.org/10.1561/0100000114.

C. L. Canonne and R. Rubinfeld. Testing probability distributions underlying aggregated data. In J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 283–295. Springer, 2014. doi: 10.1007/978-3-662-43948-7_24. URL https://doi.org/10.1007/978-3-662-43948-7_24.

C. L. Canonne, D. Ron, and R. A. Servedio. Testing equivalence between distributions using conditional samples. In *ACM–SIAM Symposium on Discrete Algorithms (SODA)*, 2014. doi: http://dx.doi.org/10.1137/1.9781611973402.87. URL http://www.cs.columbia.edu/~rocco/papers/soda14cond.html.

C. L. Canonne, D. Ron, and R. A. Servedio. Testing probability distributions using conditional samples. *SIAM Journal on Computing*, 44(3):540–616, 2015. URL https://epubs.siam.org/doi/10.1137/130945508.

C. L. Canonne, X. Chen, G. Kamath, A. Levi, and E. Waingarten. Random restrictions of high dimensional distributions and uniformity testing with subcube conditioning. In D. Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 321–336. SIAM, 2021. doi: 10.1137/1.9781611976465.21. URL https://doi.org/10.1137/1.9781611976465.21.

M. C. Caro, J. Eisert, M. Hinsche, M. Ioannou, A. Nietner, and R. Sweke. Interactive proofs for verifying (quantum) learning and testing. *CoRR*, 2024a. doi: 10.48550/ARXIV.2410.23969. URL https://doi.org/10.48550/arXiv.2410.23969.

M. C. Caro, M. Hinsche, M. Ioannou, A. Nietner, and R. Sweke. Classical verification of quantum learning. In V. Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 24:1–24:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024b. doi: 10.4230/LIPICS.ITCS.2024.24. URL https://doi.org/10.4230/LIPIcs.ITCS.2024.24.

D. Chakrabarty, X. Chen, S. Ristic, C. Seshadhri, and E. Waingarten. Monotonicity testing of high-dimensional distributions with subcube conditioning. In M. Koucký and N. Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 1019–1030. ACM, 2025. doi: 10.1145/3717823.3718297. URL https://doi.org/10.1145/3717823.3718297.

D. Chakraborty, G. Kumar, and K. S. Meel. Support size estimation: The power of conditioning. In J. Leroux, S. Lombardy, and D. Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France*, volume 272 of *LIPIcs*, pages 33:1–33:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPICS.MFCS.2023.33. URL https://doi.org/10.4230/LIPIcs.MFCS.2023.33.

D. Chakraborty, S. Chakraborty, and G. Kumar. Tight lower bound on equivalence testing in conditional sampling model. In D. P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4371–4394. SIAM, 2024. doi: 10.1137/1.9781611977912.153. URL https://doi.org/10.1137/1.9781611977912.153.

S. Chakraborty, E. Fischer, Y. Goldhirsh, and A. Matsliah. On the power of conditional samples in distribution testing. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 561–580, 2013. URL https://arxiv.org/abs/1210.8338.

X. Chen, R. Jayaram, A. Levi, and E. Waingarten. Learning and testing junta distributions with sub cube conditioning. In M. Belkin and S. Kpotufe, editors, *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pages 1060–1113. PMLR, 2021. URL http://proceedings.mlr.press/v134/chen21b.html.

A. Chiesa and T. Gur. Proofs of proximity for distribution testing. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. URL https://drops.dagstuhl.de/storage/00lipics/lipics-vol094-itcs2018/LIPIcs.ITCS.2018.53/LIPIcs.ITCS.2018.53.pdf.

I. Diakonikolas, D. M. Kane, and A. Stewart. Sharp bounds for generalized uniformity testing. In *NeurIPS*, pages 6204–6213, 2018. URL https://arxiv.org/abs/1709.02087.

M. Falahatgar, A. Jafarpour, A. Orlitsky, V. Pichapati, and A. T. Suresh. Faster algorithms for testing under conditional sampling. In *Conference on Learning Theory*, pages 607–636. PMLR, 2015. URL https://arxiv.org/abs/1504.04103.

R. Ferreira Pinto Jr. and N. Harms. Testing support size more efficiently than learning histograms. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 995–1006, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715105. doi: 10.1145/3717823.3718134. URL https://doi.org/10.1145/3717823.3718134.

O. Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 68–75. Springer, 2011. URL https://link.springer.com/chapter/10.1007/978-3-642-22670-0_9.

O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, July 1998. URL https://dl.acm.org/doi/10.1145/285055.285060.

S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 291–304, New York, NY, USA, 1985. Association for Computing Machinery. ISBN 0897911512. doi: 10.1145/22145.22178. URL https://doi.org/10.1145/22145.22178.

S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4), Sept. 2015. ISSN 0004–5411. doi: 10.1145/2699436. URL https://doi.org/10.1145/2699436.

S. Goldwasser, G. N. Rothblum, J. Shafer, and A. Yehudayoff. Interactive proofs for verifying machine learning. In J. R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 41:1–41:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPICS.ITCS.2021.41. URL https://doi.org/10.4230/LIPIcs.ITCS.2021.41.

S. Guha, A. McGregor, and S. Venkatasubramanian. Sublinear estimation of entropy and information distances. *ACM Trans. Algorithms*, 5(4):35:1–35:16, 2009. doi: 10.1145/1597036.1597038. URL https://doi.org/10.1145/1597036.1597038.

T. Gur, M. M. Jahanara, M. M. Khodabandeh, N. Rajgopal, B. Salamatian, and I. Shinkar. On the power of interactive proofs for learning. In B. Mohar, I. Shinkar, and R. O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1063–1070. ACM, 2024. doi: 10.1145/3618260.3649784. URL https://doi.org/10.1145/3618260.3649784.

T. Herman. Public coin interactive proofs for label-invariant distribution properties. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2024. URL https://drops.dagstuhl.de/storage/00lipics/lipics-vol317-approx-random2024/LIPIcs.APPROX-RANDOM.2024.72/LIPIcs.APPROX-RANDOM.2024.72.pdf.

T. Herman and G. Rothblum. Doubley-efficient interactive proofs for distribution properties. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–751. IEEE, 2023. URL https://eccc.weizmann.ac.il/report/2023/161/download.

T. Herman and G. N. Rothblum. Verifying the unseen: interactive proofs for label-invariant distribution properties. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1208–1219, 2022. URL https://dl.acm.org/doi/10.1145/3519935.3519987.

T. Herman and G. N. Rothblum. Interactive proofs for general distribution properties. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 528–538. IEEE, 2024. doi: 10.1109/FOCS61266.2024.00041. URL https://doi.org/10.1109/FOCS61266.2024.00041.

T. Herman and G. N. Rothblum. How to verify any (reasonable) distribution property: Computationally sound argument systems for distributions. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025.* OpenReview.net, 2025. URL https://openreview.net/forum?id=GfXMTAJaxZ.

G. Kamath and C. Tzamos. Anaconda: A non-adaptive conditional sampling algorithm for distribution testing. In T. M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA,* *January 6-9, 2019*, pages 679–693. SIAM, 2019. doi: 10.1137/1.9781611975482.43. URL https://doi.org/10.1137/1.9781611975482.43.

G. Kumar, K. S. Meel, and Y. Pote. Tolerant testing of high-dimensional samplers with subcube conditioning. *CoRR*, abs/2308.04264, 2023. doi: 10.48550/ARXIV.2308.04264. URL https://doi.org/10.48550/arXiv.2308.04264.

S. Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, Oct. 2000. ISSN 0097-5397. doi: 10.1137/S0097539795284959. URL https://doi.org/10.1137/S0097539795284959.

S. Mutreja and J. Shafer. PAC verification of statistical algorithms. In G. Neu and L. Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 5021–5043. PMLR, 2023. URL https://proceedings.mlr.press/v195/mutreja23a.html.

S. Narayanan. On tolerant distribution testing in the conditional sampling model. In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '21, pages 357–373, USA, 2021. Society for Industrial and Applied Mathematics. ISBN 9781611976465. URL https://dl.acm.org/doi/10.5555/3458064.3458087.

K. Onak and X. Sun. Probability-revealing samples. In *AISTATS*, volume 84 of *Proceedings of Machine Learning Research*, pages 2018–2026. PMLR, 2018.

L. Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Trans. Inf. Theor.*, 54(10):4750—-4755, Oct. 2008. ISSN 0018-9448. doi: 10.1109/TIT.2008.928987. URL https://doi.org/10.1109/TIT.2008.928987.

S. Raskhodnikova, D. Ron, A. Shpilka, and A. D. Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM J. Comput.*, 39 (3):813–842, 2009. URL https://cs-people.bu.edu/sofya/pubs/dss-sicomp.pdf.

G. N. Rothblum, S. Vadhan, and A. Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 793–802, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320290. doi: 10.1145/2488608.2488709. URL https://doi.org/10.1145/2488608.2488709.

R. Rubinfeld. Taming big probability distributions. *XRDS: Crossroads, The ACM Magazine for Students*, 19(1):24, sep 2012. doi: 10.1145/2331042.2331052. URL http://dx.doi.org/10.1145/2331042.2331052.

R. Rubinfeld and R. A. Servedio. Testing monotone high-dimensional distributions. In H. N. Gabow and R. Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 147–156. ACM, 2005. doi: 10.1145/1060590.1060613. URL https://doi.org/10.1145/1060590.1060613.

R. Rubinfeld and M. Sudan. Robust characterizations of polynomials withapplications to program testing. *SIAM J. Comput.*, 25(2):252–271, Feb. 1996. ISSN 0097-5397. doi: 10.1137/S0097539793255151. URL https://doi.org/10.1137/S0097539793255151.

G. Valiant and P. Valiant. Estimating the unseen: an n/log(n)-sample estimator for entropy and support size, shown optimal via new clts. In *STOC*, pages 685–694. ACM, 2011. URL https://dl.acm.org/doi/10.1145/1993636.1993727.

G. Valiant and P. Valiant. Estimating the unseen: improved estimators for entropy and other properties. *Journal of the ACM (JACM)*, 64(6):1–41, 2017. URL https://dl.acm.org/doi/10.1145/3125643.

P. Valiant. Testing symmetric properties of distributions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 383–392, New

York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580470. doi: 10.1145/1374376.1374432. URL https://doi.org/10.1145/1374376.1374432.

Y. Wu and P. Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *Ann. Statist.*, 47(2):857–883, 2019. ISSN 0090-5364. doi: 10.1214/17-AOS1665. URL https://doi.org/10.1214/17-AOS1665.