A Study of Private Proofs of Proximity for Distribution Properties

Clément Canonne Abigail Gentle Sam Polgar Aditya Vikram Singh

November 2026

Abstract

We consider differentially private proofs of proximity for distribution properties, a natural generalization of the proof systems of Chiesa and Gur (2018) for interactively testing statistical properties of *sensitive* data. Despite a recent study by Du, Dwork, Tankala, and Zhang (2026), the feasibility of such private proof systems remains by and large unexplored. We study differentially private proofs of proximity in the non-interactive (MA) and interactive (IP) settings, revealing trade-offs between communication and sample complexity in prior work, making connections to the shuffle model of differential privacy, considering privacy guarantees for both the verifier and the (honest) prover, and obtaining new, improved, and more flexible algorithms in each setting.

Contents

1 Introduction **3**
- 1.1 The setting . . . 4
- 1.2 Our main results and contributions . . . 5
- 1.3 Comparison with [DDTZ26] . . . 6
- 1.4 Technical overview . . . 6
  - 1.4.1 The learning baselines . . . 6
  - 1.4.2 Overview of the public-coin protocol . . . 8
  - 1.4.3 Overview of the secret-coin protocol . . . 10
  - 1.4.4 What are uniform tagged samples good for? . . . 11
- 1.5 Further Prior Work . . . 12
- 1.6 Discussion and future directions . . . 13
- 1.7 AI methodology . . . 14

2 Preliminaries **14**
- 2.1 Defining the model . . . 15
- 2.2 Useful notions and notation . . . 16

3 Generating tagged samples **17**
- 3.1 Public-coin protocol . . . 18
  - 3.1.1 Setting the stage . . . 19
  - 3.1.2 The algorithm, and its analysis . . . 20
  - 3.1.3 Privacy for the public-coin protocol . . . 25
- 3.2 Private-coin protocol . . . 26
  - 3.2.1 Sending samples to Merlin privately (without sending too much) . . . 30
  - 3.2.2 The last two missing privacy building blocks . . . 32
- 3.3 Handling the 3 assumptions . . . 32

A Connection to Augmented Testing **38**

B The MA learning baselines **40**

1 Introduction

A proof of proximity is a (typically concise) piece of evidence provided by a computationally unbounded prover that has the following traits: (1) *completeness*—if the object of interest satisfies the property, then we will be convinced of that fact with high probability, and (2) *soundness*—if the object of interest is *far* from satisfying the property, then we will only rarely be convinced otherwise. Proofs of proximity for *distribution properties* were defined and studied by Chiesa and Gur in [CG18]. In that work, it was shown that a non-interactive proof of linear length can reduce the sample complexity of testing any property (defined loosely as a subset of all distributions over $k \gg 1$ elements) to $O(\sqrt{k})$.

We study differentially private proofs of proximity for distribution properties to better understand the feasibility of their use by data-holders, large and small, represented as prover and verifier, or Merlin and Arthur.[1] In our setting, large data holders maintain control over huge samples, but must ensure strong privacy guarantees for their users, either for purely benevolent reasons, because of legal requirements, or as a means to protect their intellectual property from bad actors. The smaller data holder will of course also seek to guarantee privacy of their naturally smaller dataset, and may have even more stringent privacy requirements – having more to lose if something leaks.



Figure 1: The motivating scenario: a data-rich, computationally unbounded wizard Merlin tries to convince a data-poor mortal Arthur that the distribution of their data satisfies some property of interest to Arthur, possibly lying and potentially with the nefarious intention of obtaining Arthur's sensitive data. Both Arthur and Merlin would prefer not to leak anything about their dataset to the world and to the other party.

To make the above concrete through an example, consider a small company, looking to confirm whether their data satisfies some property. Thankfully a larger company exists, whose data purportedly comes from the same distribution (say, European home-owners). This larger company offers to check for the smaller party, but neither is happy handing off their data — a standoff! In this instance, a doubly-private proof system can facilitate this interaction, with neither party able to cheat the other.

[1] The image to have in mind is that of Merlin as a physically unconstrained wizard with the singular goal of convincing the mortal Arthur that a certain object satisfies some property. Arthur has only normal, non-magical computers, and so must come up with a scheme that *validates* Merlin's suspicious proof.

1.1 The setting

Proofs of proximity for distribution testing and their analogous proofs of proximity for distribution estimation (both formally defined in Section 2.1) capture the following: through some interaction between a prover Merlin and a sample-limited Arthur with access to an unknown probability distribution $\mathbf{p}$, the protocol's output must satisfy the following: (1) if Merlin is honest, then the output is *valid* with high probability (*i.e.*, is the right binary outcome for a testing task, or is in a set of "good outputs for $\mathbf{p}$"); and (2) if Merlin is *dishonest*, then the protocol still outputs a valid output, or detects it and outputs reject. As detailed in Section 2.1, these proof systems come in different flavors, depending on the level of interaction allowed (MA or IP protocols), and constraints on the type of communication from Arthur (secret- or public-coin protocols).[2]

We study a generalisation of these proof systems where each round of communication is assumed to be public, and therefore should satisfy differential privacy [DMNS06]. As discussed above, we will naturally assume that Arthur and Merlin have their own distinct privacy concerns, and address those concerns independently, leading to the notions of *Arthur- and Merlin-(differentially) private proof systems*.

To make this formal, consider the following interaction. Let $r \in \{0,1\}^*$ be a random string, and let $\sigma \in \mathcal{X}^n$ be a database. Let Arthur, Merlin$\colon \{0,1\}^* \to \{0,1\}^*$ be functions and $\ell \geq 0$ be an integer. An ℓ-round interaction of Arthur and Merlin on an input $x \in \{0,1\}^*$, denoted by $\langle \text{Arthur}, \text{Merlin} \rangle(x)$ is the sequence of strings $a_1, \ldots, a_\ell \in \{0,1\}^*$ such that

$$
\begin{aligned}
a_1 &= \text{Merlin}(x) \\
a_2 &= \text{Arthur}(r, \sigma, x; a_1) \\
a_3 &= \text{Merlin}(x; a_1, a_2) \\
&\vdots \\
a_{2i+1} &= \text{Merlin}(x; a_1, \ldots, a_{2i}) && (2i < \ell) \\
a_{2i+2} &= \text{Arthur}(r, \sigma, x; a_1, \ldots, a_{2i} + 1) && (2i + 1 < \ell)
\end{aligned}
$$

This interaction is (ε, δ)-*Arthur-private* if the tuple $(a_{2i+2})_{i \geq 0}$ is (ε, δ)-differentially private (DP) with respect to the sample σ. If, at each step prior to the last Arthur sends only uniformly random bits, and $a_1 = \emptyset$, then the protocol is called AM. If $\ell = 1$, then the protocol is called MA. In each case the last message is the verifier's decision, assumed to be in $\{0,1\}$.

We will also consider interactions that are *Merlin-private*: the natural definition only makes sense in the setting of "doubly-efficient" proofs, where Merlin is taken not to be omniscient, but rather to simply have a much larger sample than Arthur [HR23]. In this setting, we can similarly say that the interaction is (ε, δ)-Merlin-private if the tuple $(a_{2i+1})_{i \geq 0}$ is (ε, δ)-DP with respect to Merlin's sample.[3]

[2] To avoid confusion between differentially *private* and *private*-coin algorithms, we choose to use *secret-coin* to refer to what is known as *private-coin* in the communication complexity literature and [CG18].

[3] Alternatively, if one assumes that Merlin holds an explicit description of the distribution $\mathbf{p}$, one approach to defining privacy for Merlin would be to first decide what a "neighbouring" distribution for Merlin would be, and, seeing $\mathbf{p}$ as a single input, define Merlin-*local* differential privacy (LDP) with respect to this input and neighboring notion.

1.2 Our main results and contributions

When discussing our results, we will omit for clarity the dependence on the distance and approximation parameters α, τ, setting them to be small positive constants. Our contributions are four-fold: the first one is *conceptual*: going beyond the single notion of privacy for the verifier, which fails to address a wide range of natural settings, and introducing Merlin privacy, and showing that it can be non-trivially achieved (as detailed in Section 1.4.1, see Theorem B.3), establishing a connection to doubly efficient proofs.[4]

The second is generalizing the public-coin protocol of Herman [Her24] to trade off communication and sample complexity:

Theorem 1.1 (Informal; see Theorem 3.1)**.** *There is a public-coin protocol which, for any $1 \leq m \leq k$ and n such that $n^2 m = \tilde{\Omega}(k^2)$, outputs with high probability a set of m tagged samples, using n samples from Arthur. Its communication complexity is $O(m \log k)$.*

This generalization allows us to easily make this protocol Arthur-private while preserving the flexibility these trade-offs offer, improving on the corresponding result of [DDTZ26]:

Theorem 1.2 (Informal; see Theorem 3.2)**.** *There is an ε-Arthur-private public-coin protocol which, for any $1 \leq m \leq k$ and $n \leq m$ such that $n^2 m = \tilde{\Omega}(k^2)$ and $n^2 m = \tilde{\Omega}(k^2/\varepsilon)$, outputs with high probability a set of m tagged samples, using n samples from Arthur. Its communication complexity is $O(m \log k)$.*

This implies, ignoring privacy and logarithmic factors, that we can trade off communication for sample complexity at a rate of around $m \asymp k^c$ for $n \asymp k^{1-c/2}$ for all $c \in [2/3, 1)$. For example, given $m \asymp k^{0.9}$ (still strongly sublinear), Arthur need only take $n \gtrsim k^{0.55}$ samples. Known lower bounds, such as Lemma 1.9 imply that we require $nm \gtrsim k$, yet existing results (including ours, even with this tradeoff) all have $nm \gtrsim k^{3/2}$. Whether this is necessary is an intriguing open question.

The above theorem enables Arthur to privately obtain a set of *uniformly* distributed tagged samples: while broadly useful (see Section 1.4.4), these tagged samples are not as versatile as tagged samples distributed according to the unknown distribution itself. Our third contribution is to design an Arthur-private version of the secret-coin protocol of [HR23], allowing Arthur to privately simulate the output of any tester working in the probability-revealing sample model [CR14; OS18]:

Theorem 1.3 (Informal; see Theorem 3.3)**.** *For any m, n such that $m \geq n = \tilde{\Omega}(\sqrt{k})$ and $mn = \tilde{\Omega}(k^{3/2}/\varepsilon^2)$, there is an (ε, δ)-Arthur-private secret-coin protocol with the following guarantees. Given any testing algorithm $\mathcal{T}$ which takes as input a set of approximate tagged samples, the protocol simulates with high probability the execution of $\mathcal{T}$ on a set of $\tilde{O}(\sqrt{k})$ approximate tagged samples from $\mathbf{p}$. The communication complexity is $O(m \log k)$, and the sample complexity $O(n/\varepsilon)$.*

(Note that the protocols underlying both Theorems 1.2 and 1.3 are doubly efficient and thus can be made Merlin-private as well, using the baseline approach described in Section 1.4.1)

Our last contribution is to establish a (simple) two-way connection between MA (and coMA) proof systems for distribution properties and the recently proposed model of *augmented testing* [AIRS24; ABCR25] (see Definition A.2), allowing one to carry results between the two models:

[4] See Section 2.1 for a definition of doubly efficient: roughly speaking, this involves Merlin being, when following the honest strategy, both sample- and computationally efficient.

Theorem 1.4 (Informal; see Theorems A.3 and A.4)**.** *If a property of distributions $\mathcal{P} = (\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ has an augmented tester with sample complexity n and advice complexity c, then both $\mathcal{P}$ and its complement $\mathcal{P}' = (\mathcal{P}_{\text{no}}, \mathcal{P}_{\text{yes}})$ have an MA proof system with the same sample complexity and communication complexity c. Conversely, if a property and its complement have an MA proof system with sample complexity n and communication complexity c, then it has an augmented tester with sample complexity $O(n)$ and advice complexity $O(c)$.*

1.3 Comparison with [DDTZ26]

The recent work of [DDTZ26] formally introduced the model of what we call Arthur-differentially private proofs of proximity for distribution properties. They correctly argued that all communication from Arthur to Merlin, along with any decisions made after interacting with Merlin must be differentially private. To achieve this, they make a connection to *replicability*, enabling one to leverage the results from this significant line of work to design Arthur-private public-coin protocols, in the high privacy regime $\varepsilon \lesssim 1/\sqrt{n}$.

In the low-privacy regime, they are the first to note that the secret-coin protocol of [HR22] satisfies $\log(n)$-local differential privacy in the communication step. In Section 3.2 we complete this analysis by making the soundness checks differentially private also.

The authors introduce the first MA protocol for independence testing — even non-privately, and use this to give an efficient private identity tester. They are the first to note that since Merlin has unbounded sensitivity, one might want to use the generic "private to non-private" reduction of [ADR18] to complete the soundness checks. Naturally this implies that any public-coin protocol with sample complexity n has a private analogue with sample complexity $O(n/\varepsilon)$. We use this generic technique (though in a more involved way) in Section 3.2 to make one of our secret-coin soundness checks private, after making Arthur's first message to Merlin differentially private. To surpass this generic approach, they finally introduce an Arthur-private AM protocol for generating *uniformly distributed tagged samples* (see Section 1.4.4 for a discussion of this primitive in testing). Their approach, much like ours, is to learn the tests up to maximum error using Laplace histograms. We generalize this approach in Section 3.1 by demonstrating trade-offs between communication and sample complexity in the non-private algorithm.

1.4 Technical overview

We here provide a high-level overview of the techniques underlying our results, starting with a discussion of the "learning baselines," which provide MA distribution estimators (see Definition 2.6).

1.4.1 The learning baselines

We here state and discuss our "baselines," general-purpose MA upper bounds which show what can be achieved when one allows near-linear proof length. The proof of all results in this section can be found in Section B. To set the stage for our protocols, we first recall the corresponding result of Chiesa and Gur:

Theorem 1.5 ([CG18, Theorem 1])**.** *There exists an MA distribution estimator with communication complexity $O(k \log(k/\alpha))$ and sample complexity $O(\sqrt{k}/\alpha^2)$, allowing Arthur to* learn *the distribution $\mathbf{p}$ to total variation distance α.*

The idea behind this protocol is simple: Merlin can send a purported description $\tilde{p}$ of the distribution $\mathbf{p}$ to Arthur, who then uses an out-of-the-box identity testing algorithm to check that $\mathbf{p}$ is equal to $\tilde{\mathbf{p}}$. This does not immediately work, as the exact description of $\tilde{\mathbf{p}}$ might require unbounded communication: instead, Merlin sends a quantization of the distribution, to suitable fine accuracy (each probability given to $\pm O(\alpha/k^{3/2})$ suffices), and Arthur then performs (mildly) tolerant identity testing to check that $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}) \lesssim \alpha/\sqrt{k}$. This reduces the communication, and the sample complexity for this "mildly tolerant testing" remains $O(\sqrt{k}/\alpha^2)$ (see,*e.g.*, [CJKL22] for a discussion). The upshot is that *a proof of (near) linear length reduces all property testing to identity testing*; this baseline should thus be the yardstick with which to assess all our algorithms.

One issue with the above algorithm, besides its long proof size, is that it "violates the spirit" of an MA proof: namely, *Merlin needs to know Arthur's target accuracy α in order to design the proof.* In many cases, this would require prior interaction between Arthur and Merlin, and might require Merlin to publish a different proof every time a new Arthur wants to run the protocol. To address this, we provide a different protocol, with essentially the same guarantees:

Theorem 1.6 (Oblivious-Merlin baseline (Informal; see Theorem B.1))**.** *There exists an* MA *distribution estimator with communication complexity* $O(k \log k)$ *and sample complexity* $O(\sqrt{k}/\alpha^2)$*, allowing Arthur to learn the distribution* $\mathbf{p}$ *to total variation distance* α*. Moreover, Merlin does not need to know* α*.*

The idea, once again, is quite simple: Merlin sends to Arthur a *permutation* $\tilde{\pi}$ of the domain, claimed to make the distribution $\tilde{\mathbf{p}} := \mathbf{p} \circ \tilde{\pi}$ monotone. Arthur then uses a monotonicity testing algorithm to check that $\tilde{\mathbf{p}}$ is monotone: if the test accepts, it can learn $\tilde{\mathbf{p}}$ with very few samples, and applying $\tilde{\pi}^{-1}$ to the learned distribution gives a good approximation of $\mathbf{p}$.

Arthur-privacy. Turning to another direction, we note that the protocol of Theorem 1.5 can easily be made Arthur-DP, by using an optimal differentially private tester for the identity testing step performed by Arthur.[5] Using the optimal DP identity tester of [ASZ18, Theorem 2], we get

Theorem 1.7 (Arthur-private baseline (Informal; see Theorem B.2))**.** *There exists an* ε*-Arthur private* MA *distribution estimator with communication complexity* $O(k \log(k/(\alpha\varepsilon)))$ *and sample complexity*

$$O\left(\frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha\sqrt{\varepsilon}} + \frac{k^{1/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{1}{\alpha\varepsilon}\right)$$

allowing Arthur to learn the distribution $\mathbf{p}$ *to total variation distance* α*.*

We note that the above protocol, while achieving optimal sample complexity[6] (at the cost of near-linear proof length) does suffer from the shortcoming mentioned above: namely, Merlin must

[5] There is a slight technicality here, as these testers may not provide exactly the claimed "mild tolerance" out-of-the-box – tolerance which comes for free for every collision-based tester, using ℓ_2 as a proxy for TV distance. However, we can always obtain a slightly worse, but sufficient tolerance by invoking the fact than any n-sample tester is $O(1/n)$-tolerant.

[6] It is easy to see that the simulation argument of [CG18, Observation 3.10] carries over to the differentially private setting, showing that any ε-Arthur-private proof system to verify uniformity must have sample complexity,

$$\Omega\left(\frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha\sqrt{\varepsilon}} + \frac{k^{1/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{1}{\alpha\varepsilon}\right)$$

by [ASZ18, Theorem 2], regardless of communication complexity. As learning the distribution enables Arthur to verify uniformity, the lower bound applies here.

know both Arthur's accuracy and privacy parameters α and ε. (To address this, one option would be to make Theorem B.1 Arthur-private, by designing a DP *monotonicity* testing algorithm.)

Merlin-privacy. Now, considering privacy from Merlin's point of view, one can use a very generic approach to make Theorem 1.5 Merlin-private: namely, the observation that *every doubly-efficient protocol*[7] can be made Merlin-private, by making Merlin first learn the distribution to sufficient accuracy, in an ε-DP way, before running the rest of the protocol using the approximation $\tilde{\mathbf{p}}$ thus obtained. Since ε-DP learning to accuracy τ can be done with $O(k/\tau^2 + k/(\tau\varepsilon))$ samples, this provides a generic upper bound for Merlin.[8]

This baseline, however, may be too costly (from Merlin's point of view). Our next result shows that if Merlin's privacy and sample complexity are a key concern, one can offload some of the sample complexity cost to Arthur – still allowing him, of course, to learn $\mathbf{p}$ with sublinear sample complexity.

Theorem 1.8 (Merlin-private baseline (Informal; see Theorem B.3))**.** *For every $\sqrt{k}/\alpha^2 \leq t \leq \varepsilon k$, there exists an ε-Merlin private MA distribution estimator with communication complexity $O(k \log(k/\alpha))$, sample complexity $O\left(\sqrt{k}/\alpha^2 + t\right)$ for Arthur, and $O(k^3/(\alpha^4 t^2) + k^2/(\alpha^2 \varepsilon t))$ for an honest Merlin, allowing Arthur to learn the distribution $\mathbf{p}$ to total variation distance α.*

1.4.2 Overview of the public-coin protocol

We first review the protocol of [Her24], which ours is based on, before detailing how to make it Arthur-private.

Comparison with [Her24]. Before our exposition we first detail — for familiar readers — a few ways our approach will differ from the protocol of [Her24]. First, we will decompose the sample complexity of the algorithm into m (that denotes the *uniform*-sample complexity) and n (that denotes the $\mathbf{p}$-sample complexity). Naturally, and especially under privacy, one would assume that samples from $\mathbf{p}$ are more expensive than samples from the uniform distribution.

Recall that the algorithm of [Her24] has sample and communication complexity (which is well-approximated by the uniform-sample complexity) $n \approx m \approx k^{0.67}$. By separating these quantities in our analysis, we found that there exist natural trade-offs between the communication and sample complexity of the algorithm — if Arthur sends $m = k^{0.9} \gg k^{0.67}$ uniform samples to Merlin, it is natural to assume that he will require fewer $\mathbf{p}$ samples in comparison. This turns out to be true; in this instance Arthur only requires $n = k^{0.55}$ samples to verify the proof supplied by Merlin.

In order to simplify our new exposition of the algorithm, we also move several checks out of the algorithm and into assumptions about the input distribution. We then demonstrate in Section 3.3 how the input distribution can be *made* to satisfy these assumptions in a computationally trivial manner. We hope that this exposition proves as not only a generalisation, but also as a means to better understand the results of [Her24].

[7] Recall that a *doubly-efficient* interactive proof for distribution properties [HR23] is a protocol where Merlin is sample-efficient, *i.e.*, does not know $\mathbf{p}$ but only has a near-linear number of samples, comparable to the cost of learning.

[8] Note that for Theorem 1.5, Merlin needs its accuracy τ to match the mild tolerance of Arthur's tester, setting $\tau \asymp \alpha/\sqrt{k}$.

Algorithm 1 A minimal pseudocode of the public-coin protocol

1: Arthur: Draw $S \sim \mathbf{u}_{[k]}^{\otimes m}$ and send it to Merlin.
2: Merlin: Summon $\tilde{\mathbf{p}}$, and send $(s, \tilde{\mathbf{p}}(s))_{s \in S}$ to Arthur.
3: Arthur: Using two new samples $X, Y \sim \mathbf{p}^{\otimes n}$, for each $0 \leq j \leq L$, check the overall mass and collision probability of purported bucket j. If each test passes, output accept.

To begin with this exposition, consider a different goal, which we will refine later. Suppose we ask that Merlin return $(x, \tilde{\mathbf{p}}(x))$ for each sample Arthur sends. How might we verify Merlin's honesty?

First, let $S \sim \mathbf{u}_{[k]}^{\otimes m}$ be a uniformly random sample, Arthur sends S to Merlin and receives $(s, \tilde{\mathbf{p}}(s))_{s \in S}$ where $\tilde{\mathbf{p}}(s)$ is the purported probability of s in $\mathbf{p}$. If Merlin is honest this quantity is the empirical frequency that Merlin has estimated with their own sample, otherwise it is completely arbitrary. Arthur then must verify these tags. Take a sample $X \sim \mathbf{p}^{\otimes n}$ and consider X restricted to the coverage of S. We must verify that the total mass of the purported tags makes sense on that subset of the domain. Now, suppose we receive a sample $\{\dots, a, a, a, b, b, b, c, \dots\}$ that contains elements a, b, c, all of which are tagged as having probability p. We might complete this check and see that $3p = 7/n$ which does in fact line up with the fact that we saw a, b and c seven times in all, but if they all have the same probability, why did we see c only once? What we would like to see is that within each probability "bucket" the elements are uniform (as we would expect, given that they have the same probability).

To check uniformity of each bucket, we will count the collisions that occur. Buckets that are uniform should have few collisions overall. Again, we will only consider parts of the domain that we have seen in the support of our uniform sample.

In the above sketch, we assumed that there exist three elements each with the same probability of occurring. However, this is far from guaranteed in actuality. We will have to also handle distributions where no two elements share the same probability. To do so, we will bucket the tags, grouping those that are "close enough" in probability. Namely, we partition the tagged uniform sample $(s, \tilde{\mathbf{p}}(s))_{s \in S}$ into the sets $(B_j)_{0 \leq j \leq L}$ where, for $0 \leq j \leq L$, where $L = O(\log(k/\tau)/\tau^3)$:

$$B_j := \left\{ s \in S : \tilde{\mathbf{p}}(s) \in [\bar{p}_j, (1+\tau^3)\bar{p}_j) \right\}, \tag{1}$$

with $\bar{p}_j := \frac{\tau(1+\tau^3)^j}{k}$. Notice that these buckets cover the range of probabilities from τ/k to 1; we will handle the "small-probability items" (with probability in $[0, \tau/k)$) separately.

To test the correctness of these buckets, draw two *new* samples X and Y from $\mathbf{p}^{\otimes n}$. Working with what we've got—we will use only the parts of these samples that fall into the support of B_j. As such, define the tuples

$$X_j = (x \in X : x \in B_j) \qquad Y_j = (y \in Y : y \in B_j) \qquad Z_j = ((x, y) \in X \times Y : x = y \in B_j)$$

for $0 \leq j \leq L - 1$. Since B_j is supposed to contain elements with probability $\bar{p}_j$, we expect that $|X_j| \approx n \cdot |B_j| \cdot \bar{p}_j$. This test shows that the bucket has the right amount of mass, but we require that our buckets be groupings of items with approximately the same probability, not a mishmash of elements that just add up to the right probability. To this end, we will test whether the elements are *uniform* within the bucket. As is typical of uniformity testing, we will do so by counting collisions. Let Z_j be the multiset of elements that appear in both samples and B_j. The expected size of this set is at least $n^2 \cdot |B_j| \cdot \bar{p}_j^2$, presuming Merlin is telling the truth. What we will have to argue is that these

tests passing *imply* that Merlin is telling the truth, or at least something close enough to the truth to remain useful.

What remains is a concentration argument, claiming that these statistics don't deviate far when Merlin is honest. In the case that Merlin is dishonest, we instead claim that these tests passing imply the usefulness of Merlin's advice, showing that Merlin failed to meaningfully deceive us.

Privacy of the public-coin protocol. By taking a public-coin approach, Arthur has naturally revealed very little about his samples to Merlin. Still, given that Merlin is resource-unbounded, we must remain on our guard. To ensure privacy of this whole protocol requires ensuring we are private with respect to the samples X and Y. As such, we focus on making the *tests* differentially private. To this end, we simply learn the histogram of tests up to small simultaneous (maximum) error, being careful about the sensitivity of the collisions test in particular: a naive approach would be too costly, as their sensitivity could be as large as n. However, a (somewhat standard) high-probability argument, using the fact that no element will appear more than $O(\log n)$ in any of the B_j's, allows us to bound the sensitivity by $O(\log n)$ instead. By thresholding the number of collisions appropriately, we then incur minimal loss in learning a histogram over $L \approx \log k$ items.

1.4.3 Overview of the secret-coin protocol

In the secret-coin setting, the high-level idea of the protocol of [HR23] to learn the bucketing is as follows: first, Arthur generates a sequence of $m+n$ samples σ by "mixing" n samples from $\mathbf{p}$ with m uniform samples, remembering secretly which ones are which.[9] After sending this sequence to Merlin to obtain the (purported) "(bucket) tags" for each sample, he is in possession of $L+1 := O(\log(k/\tau)/\tau)$ disjoint *multi*sets $\tilde{B}_0, \dots, \tilde{B}_L$, with each $\tilde{B}_j$ containing the samples tagged as having probability $\approx \bar{p}_j$. The key idea is that Merlin is not told which of the $m+n$ elements come from which distribution, and so, in tagging them, a malicious Merlin trying to fool Arthur would have to guess, and mislabel roughly the same proportions of "$\mathbf{p}$-samples" and "uniform samples." Since Arthur knows the provenance of the samples, he can check that the corresponding number of $\mathbf{p}$-samples $|\tilde{B}_j^{\mathbf{p}}|$ and uniform samples $|\tilde{B}_j^{\mathbf{u}}|$ within each B_j are what they should be. He also, taking some additional samples from $\mathbf{p}$, can estimate the total mass $\mathbf{p}(\tilde{B}_j^{\mathbf{p}})$ and check that this matches what it should be, $\mathbf{p}(\tilde{B}_j^{\mathbf{p}}) \approx |\tilde{B}_j^{\mathbf{p}}|\bar{p}_j$. The crux of the argument of Herman and Rothblum is that *these two tests suffice*: if they pass for every j, then the $\mathbf{p}$-samples tagged by Merlin must be accurate.

Our goal is to make this Arthur-private: this is not trivial, as the protocol requires first to send some of Arthur's samples "in the clear", and then to reuse them (using in the process the "secret" information of which parts of the message came from $\mathbf{p}$), while also computing with some other samples a high-sensitivity statistic to estimate $\mathbf{p}(\tilde{B}_j^{\mathbf{p}})$ for all j. Our approach will be to divide the samples in two disjoint sets S_1, S_2 and argue separately about their privacy, to avoid having to deal with intricate subtleties in the privacy analysis and be able to use composition theorems.

1. *Privacy for the samples sent to Merlin*: The very first message sent to Merlin, which includes samples from S_1 in the clear: we will make this part (ε, δ)-private *via* a combination of randomized response and privacy amplification by shuffling;
2. *Privacy for the samples in* S_1: after receiving the tagged samples back from Merlin, Arthur might potentially "reveal" information about them by using their provenance (which part of

[9] In [HR23], the protocol uses $m = n$: for our purposes, we generalize it straightforwardly to $m \geq n$.

the mixture they came from) when running the check on the relative sizes of the $|\tilde{B}_j^{\mathbf{p}}|$'s and $|\tilde{B}_j^{\mathbf{u}}|$'s. Rather than trying to argue privacy of these cheks, we instead view the whole process as computing a test f based on 3 inputs: (1) the response from Merlin, which is (ε, δ)-private by the first item; (2) the multiset of samples S_1; and (3) the disjoint multiset of samples S_2. To show this computation is ε-DP with respect to S_1, we will rely on a known generic private to non-private reduction (Lemma 3.17 [ADR18]) for tests, using that the set of n samples from $\mathbf{p}$ used to form S_1 is a uniformly random $O(\varepsilon)$-fraction of our total set of samples from $\mathbf{p}$.

3. *Privacy for the samples in* S_2: the estimation of the $\mathbf{p}(\tilde{B}_j^{\mathbf{p}})$ is the only step using S_2, and this can be done via the standard Laplace histograms technique. Still, to argue about privacy here, we need to be a bit careful, as the corresponding statistics could have high sensitivity: a single element of S_2 could collide with *many* elements in $\tilde{B}_j^{\mathbf{p}}$, since the latter is a *multi*set. Fortunately, we can avoid this by leveraging a high-probability bound: namely, that with overwhelming probability no element appears more than $O(\log n)$ times in $\tilde{B}_j^{\mathbf{p}}$.

Importantly, the reason for which we need this generic private to non-private reduction in the second item (instead of simply invoking postprocessing properties of DP along with the privacy guarantee of the first one) is that the tests in ll. 8 and 9 involve $\sigma^{\mathbf{p}}$, *not* the publicly released σ. That is, they are not just postprocessing of the first step's output, but require Arthur to use unreleased data (the labels indicating which samples from σ came from $\mathbf{p}$ and which came from $\mathbf{u}$), which are not public.

Now, this generic reduction may appear costly: we lost a factor $O(1/\varepsilon)$ in the sample complexity! The saving grace is that *we actually do not.* Indeed, we lose a factor $O(1/\varepsilon)$ in the sample complexity *of the first steps*, so that S_1 is indeed much larger than what it would be in the non-private protocol. But S_2 would need to be this large anyway for the second step, and we do not blow up the size of S_2 by this $1/\varepsilon$ factor: thus, the total private sample complexity, $|S_1| + |S_2| = O(|S_2|)$, does not increase by that much overall compared to the non-private version.

1.4.4 What are uniform tagged samples good for?

Assuming the checks in Algorithm 2 pass, we end up with $((\sigma_i, \mathrm{tag}(i)))_{i\in[m]}$ where σ_i's are uniformly generated from $\mathbf{u}_k$ and $\mathrm{tag}(i)$'s are τ-approximations of $\mathbf{p}(\sigma_i)$. Lemma 27 in [Her24] claims that given these tagged samples and $\mathbf{q}(\sigma_i)$ for some distribution $\mathbf{q}$, one can estimate the total variation distance $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{q})$ to accuracy $\tau + 1/\sqrt{m}$. However, there is a gap in the proof, which blows up the approximation by a multiplicative factor of k, rendering the bound useless. This, in particular, invalidates the main theorem (Theorem 1) in their work. However, not all is lost.

As shown in [CR14; OS18], tagged samples where the samples are generated from $\mathbf{p}$ are quite handy as they allow us to estimate quantities of the form $\mathbb{E}_{x\sim\mathbf{p}}[f(x, \mathbf{p}(x))]$ for bounded f (such as entropy, total variation distance from a known $\mathbf{q}$, among others). Being in possession of uniform tagged samples, similarly, would allow us to estimate quantities of the form $\mathbb{E}_{x\sim\mathbf{u}_k}[f(x, \mathbf{p}(x))]$. Are there useful quantities of this form? The question might appear moot since

$$\mathbb{E}_{x\sim\mathbf{p}}[f(x, \mathbf{p}(x))] = \mathbb{E}_{x\sim\mathbf{u}_k}[g(x, \mathbf{p}(x))], \text{ where } g(x, \mathbf{p}(x)) = k\mathbf{p}(x)f(x, \mathbf{p}(x))$$

and thus, uniform tagged samples seem equally useful. The catch is in the multiplicative factor $k\mathbf{p}(x)$, which, in the worst case can blow up the sample requirement of estimation by a factor of k (compared to if we had tagged samples from $\mathbf{p}$). However, there *are* quantities that one can estimate using uniform tagged samples, such as

- **Distance from $\mathbf{u}_k$**: Note that

$$2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}_k) = \sum_{x\in[k]} |\mathbf{p}(x) - \tfrac{1}{k}| = \mathbb{E}_{x\sim\mathbf{u}_k}[|k\mathbf{p}(x) - 1|].$$

  From Assumption 2, $\mathbf{p}(x) < 1/n$ and thus $|k\mathbf{p}(x) - 1| = O(k/n)$. This implies (by Hoeffding's inequality) that if $m = \Omega(k^2/n^2)$, one can estimate $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}_k)$ up to constant accuracy using m uniform tagged samples. By Theorem 3.1, we do indeed have the required m for this task.[10]

- **Effective support size**: the protocol allows us to estimate the 2τ-effective support size $|\{ x \in [k] : \mathbf{p}(x) \geq 2\tau/k \}|$ to additive $\pm\alpha k$, writing it as

$$\mathbb{E}_{x\sim\mathbf{u}_k}[\mathbb{1}_{\{\mathbf{p}(x)\geq 2\tau/k\}}]$$

- **KL divergence** $\mathrm{KL}(\mathbf{u}_k \,||\, \mathbf{p})$:

$$\mathrm{KL}(\mathbf{u}_k \,||\, \mathbf{p}) = \mathbb{E}_{x\sim\mathbf{u}_k}\left[\log \frac{1}{k\mathbf{p}(x)}\right]$$

1.5 Further Prior Work

Distribution testing is the problem of deciding whether a distribution has a certain property, or is far (in total variation distance) from all distributions with that property. The key goal of testing is to use (much) fewer samples than would be needed to learn the distribution to TV distance α, and in particular to use a *sublinear* number of samples ($o(k/\alpha^2)$, where k is the domain size). A landmark result in distribution testing is the fact that testing "goodness-of-fit" also—and hereafter—called *identity testing* requires $\Theta\left(\sqrt{k}/\alpha^2\right)$ samples [Pan08]. This subsumes—and is in fact equivalent to—testing whether a distribution is uniform [Gol20; Can22]. Another useful point of reference will be the complexity of *tolerant* testing; deciding whether a distribution is $\alpha/2$ close to a property, or α-far from it requires $\Omega(k/(\alpha^2 \log k))$ samples in general [Gol17]. Many properties are *label invariant*, in that if a distribution $\mathbf{p}$ is in the property, then so are all of its relabellings. Clearly uniformity is label invariant, but so too are other notable properties such as entropy and support size. When a property is label invariant, we can analyse its *fingerprint*, an object that can loosely be understood as the "histogram of its histogram," an object that just counts, for each $\mathbf{p}_i$ the number of domain elements with that probability [Val11; VV11]. Several surveys on distribution testing are available, and we point interested readers to [Gol17; Can22].

As is common in computer science, we are also interested in asking whether our algorithms can be improved by access to an untrustworthy oracle. A proof of proximity is a transcript of communication with such an oracle that is complete, in that it usually convinces us that a property holds whenever that really is the case—and sound, in that it rarely convinces us a property holds when it really does not. Such proofs are fundamental to the definition of important results in computational complexity, but here we limit our attention to proofs of proximity *for distribution properties*. These proofs, introduced by [CG18], have seen rapid development. First, [CG18] demonstrated that non-interactive proofs of linear size (such as a description of the distribution) reduce the verifier's sample complexity to $O(\sqrt{k})$ regardless of the property, and that $\Omega(\sqrt{k})$ samples were necessary to verify uniformity,

[10] Importantly, this argument also mends the gap in [Her24, Lemma 27], under the assumption that $\|\mathbf{p}\|_\infty, \|\mathbf{q}\|_\infty \leq 1/n$.

regardless of proof length. A line of follow-up work established that all *label invariant* properties admit 2-message interactive proofs with sample and communication complexity $\tilde{O}(\sqrt{k})$ [HR22], and 2-message AM proof (where the verifier sends only uniformly random strings) with sample and communication complexity $\tilde{O}(k^{2/3})$.
Chiesa and Gur [CG18] further established the following lower bound.

Lemma 1.9 ([CG18, Lemma 6.6])**.** *For every property $\mathcal{P} \subseteq \Delta_k$, if there exists a distribution tester with sample complexity n', then any AM distribution tester for $\mathcal{P}$ with communication complexity c and sample complexity n must have $c \cdot n = \Omega(n')$.*

Finally, to investigate when the $\tilde{O}\left(\sqrt{k}\right)$ barrier could be breached, the problem was studied with *conditional queries* by [BBCS25].

Differential privacy intersects all of the previously discussed topics many times over, but to pick a representative few results we note that private identity testing has sample complexity $\Theta(\sqrt{k}/\alpha^2 + \sqrt{k}/(\alpha\sqrt{\varepsilon}) + k^{1/3}/(\alpha^{4/3}\varepsilon^{2/3}) + 1/\alpha\varepsilon)$ in the central model [ASZ18; ADR17], the "typical" setting where the analyst holds the entire dataset. Under the more restrictive local model, there is a separation between public and secret-coin protocols, and the sample complexity becomes super-linear: $\Theta(k/(\alpha^2\varepsilon^2))$ and $\Theta\left(k^{3/2}/(\alpha^2\varepsilon^2)\right)$ respectively [ACFT19; ACT19; ACFST21].

Closely related to the concept of proofs is the notion of *augmented testing*. In the augmented testing model an algorithm is supplied a piece of *untrusted advice*, and is expected to output either a correct decision, or "bad advice" [AIRS24]. This model was extended to differential privacy in the work of [ABCR25]. In prior work this advice is always a description of the distribution, but one could generalize and consider any arbitrary advice that would help the protocol make its decision. Focusing on uniformity, both papers demonstrated that advice is most useful when it says that the distribution is *far* from uniform, *i.e.*, for the complement problem; this is to contrast with the result of [CG18] who showed no gain for verifying uniformity was possible, even with linear-sized proofs.

1.6 Discussion and future directions

We hope our work will spark interest and further research in differentially private proofs of proximity. We conclude this introduction by outlining three concrete directions of research which we believe to be promising.

- The first one is to extend the current protocols, both public- and secret-coin, to other types of bucketings than the standard one from (3). In particular, can one extend the "permutation-based" approach of Theorem 1.6 with a bucketing tailored to monotone distributions (such as the so-called Birgé) decomposition) to obtain alternative IP distribution estimators?

- The second, more generally, is about the *type of access* simulated. Namely, tagged samples (according to $\mathbf{p}$) are akin to simulating the probability-revealing access model of [CR14; OS18]. Can we use interaction to simulate other strong access models, *e.g.*, *conditional sampling* access [CRS15; CFGM16] or its variants [BC18]? (This would allow one to leverage the recent non-private results of [BBCS25], which endow Arthur with such conditional sampling access.)

- Finally, can we design other approaches to achieve Merlin-privacy, more sample-efficient than the ones we provided here? In particular, is there a connection to recent results on *differentially private sampling* [RSSS25] which would allow (honest) Merlin to engage in the protocol at a cost strictly lower than the one of *learning* the distribution?

1.7 AI methodology

The ideas, discussions, proofs, writing, research meetings, occasional thinking, whiteboard sketches and doodles, and sometimes poor choice of notation are solely due to the authors. No AI system was used in the process of writing this paper.

2 Preliminaries

We first recall the standard definition and properties of differential privacy:

Definition 2.1 (Differential Privacy [DMNS06])**.** An algorithm $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ satisfies (ε, δ)-differential privacy, if for all pairs of *neighboring* datasets $x \sim x'$, and all events $E \subseteq \mathcal{Y}$:

$$\frac{\Pr[\,\mathcal{A}(x) \in E\,] - \delta}{\Pr[\,\mathcal{A}(x') \in E\,]} \leq e^{\varepsilon}.$$

If $\delta = 0$ then we say that $\mathcal{A}$ satisfies *pure*-differential privacy.

Throughout this work, we will say that two datasets $x \sim x'$ are neighbouring if the Hamming distance between them equals 1, and take the size of the dataset n as being public. This is also known as the *replacement* or *substitution* neighboring relation.

The following two properties follow immediately from the definition, and are used extensively in our privacy analysis.

Simple Composition: The algorithm $\mathcal{A}'$ that, for a given dataset σ, first releases the output of an $(\varepsilon_1, \delta_1)$-differentially private algorithm $\mathcal{A}_1(\sigma)$, and then does the same with the $(\varepsilon_2, \delta_2)$-differentially private algorithm $\mathcal{A}_2(\sigma)$ satisfies $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differential privacy.

Postprocessing: No algorithm $\mathcal{M}$ applied to the output of a (ε, δ)-differentially private algorithm, that does not depend on the dataset σ, can decrease the privacy of that output.

Finally, we recall the definition of Randomized Response, a standard differentially private mechanism:

Definition 2.2 (k-ary randomized response (k-RR) [KOV16; KBR16])**.** The *k-ary randomized response* mechanism $\mathrm{RR}_k : \mathcal{X} \to \mathcal{X}$ is a ε-locally differentially private mechanism defined as having the following distribution on input x:

$$\mathrm{RR}_k(x) = \begin{cases} x & \text{with probability } \frac{e^{\varepsilon}-1}{e^{\varepsilon}+k-1}, \\ y \sim \mathbf{u}_{[k]} & \text{with probability} \frac{k}{e^{\varepsilon}+k-1}. \end{cases}$$

To establish some common notation, let $\mathbf{p}, \mathbf{q} \in \Delta(\mathcal{X})$ be distributions from the set of distributions on $\mathcal{X}$, and $\mathbf{u}_{\mathcal{X}}$ be the uniform distribution over $\mathcal{X}$. We write $\mathbf{p}^{\otimes n}$ to denote the n-fold product distribution. In this work we take $\mathcal{X} = [k] = \{1, \ldots, k\}$. We let n, m denote sample complexities, where n is typically the number of samples drawn by Arthur from $\mathbf{p}$, in contrast to m that represents either the samples drawn by Merlin, or additional samples drawn from $\mathbf{u}_{[k]}$. These distinctions are made unambiguous in each section.

We use the shorthand $x \gtrsim y$ to indicate that $x \leq C \cdot y$ for some absolute constant $C > 0$, and define $x \lesssim y$ similarly; we then write $x \asymp y$ whenever $x \lesssim y$ and $y \lesssim x$. We take the usual notion of total variation distance defined as

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) = \sup_{E \subseteq \mathcal{X}} |\mathbf{p}(E) - \mathbf{q}(E)| = \frac{1}{2}\|\mathbf{p} - \mathbf{q}\|_1 .$$

Finally, for $\gamma \in [0,1]$, we say that $x \overset{1+\gamma}{\approx} y$ if $(1-\gamma)y \leq x \leq (1+\gamma)y$ almost surely.

2.1 Defining the model

We here formally define the standard definition of distribution testing, central to this work.

Definition 2.3 (Distribution tester for a promise problem)**.** Let $\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}} \subseteq \Delta(\mathcal{X})$ be disjoint. A *distribution tester* for $(\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ is a probabilistic algorithm A representing Arthur the verifier, that given sample access to $\mathbf{p}$, satisfies:

- (Completeness) for every $\mathbf{p} \in \mathcal{P}_{\text{yes}}$, $\Pr[\, A^{\mathbf{p}} = \text{accept}\,] \geq 2/3$.
- (Soundness) for every $\mathbf{p} \in \mathcal{P}_{\text{no}}$, $\Pr[\, A^{\mathbf{p}} = \text{accept}\,] \leq 1/3$.

The *sample complexity* of A is the (worst-case) number of samples A draws from $\mathbf{p}$. $A^{\mathbf{p}}$ denotes the tester Arthur having access to the distribution $\mathbf{p}$.

We now recall the definitions of proof systems for distribution properties, as introduced by Chiesa and Gur [CG18, Definition 2.1].

Definition 2.4 (MA-D distribution tester ([CG18, Definition 3.1]))**.** Let $\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}} \subseteq \Delta(\mathcal{X})$ be disjoint. An *MA distribution tester* for $(\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ is a distribution tester A, the *verifier*, that receives explicit access to a proof string $\pi \in \{0,1\}^*$ written by an untrusted, unbounded *prover* M who knows $\mathbf{p}$:

- (Completeness) for every $\mathbf{p} \in \mathcal{P}_{\text{yes}}$ there exists $\pi \in \{0,1\}^*$ such that $\Pr[\, A^{\mathbf{p}}(\pi) = \text{accept}\,] \geq 2/3$.
- (Soundness) for every $\mathbf{p} \in \mathcal{P}_{\text{no}}$ and every $\pi \in \{0,1\}^*$, $\Pr[\, A^{\mathbf{p}}(\pi) = \text{accept}\,] \leq 1/3$.

The *proof complexity* of A is the (worst-case) length of the honest proof π.

An MA-D tester gets a single, static proof string from M. An *IP distribution tester* generalizes this by letting A adaptively interact with M over several rounds.

Definition 2.5 (IP distribution tester ([CG18, Definition 5.1]))**.** An *IP distribution tester* is a pair (A, M), A has sample access to $\mathbf{p}$, and M is adaptive and knows $\mathbf{p}$. The interaction is over one or more rounds; we write $\langle A^{\mathbf{p}}, M(\mathbf{p})\rangle$ for the verifier view at the end of the interaction. The pair satisfies:

- (Completeness) for every $\mathbf{p} \in \mathcal{P}_{\text{yes}}$, $\Pr[\, \langle A^{\mathbf{p}}, M(\mathbf{p})\rangle = \text{accept}\,] \geq 2/3$;
- (Soundness) for every $\mathbf{p} \in \mathcal{P}_{\text{no}}$ and every prover strategy $\tilde{M}$, $\Pr\left[\, \left\langle A^{\mathbf{p}}, \tilde{M}(\mathbf{p})\right\rangle = \text{accept}\,\right] \leq 1/3$.

The *communication complexity* of Π, denoted by c, is the (worst-case) number of bits exchanged between M and A in Π. The *round complexity* r is the (worst-case) number of rounds of interaction, each round consists of a message from one party to the other and its reply. We say Π is *efficient* when A runs in time polynomial in k and its sample complexity n. When all messages from Arthur consist only of uniformly random bits, Π is said to be *public-coin*; otherwise, it is *secret-coin*.

Efficient provers. One can modify in the definitions above to require that the (honest) prover Merlin is not given $\mathbf{p}$ explicitly as input, but instead also only has sample access to it. When this is the case, and both the (honest) prover sample and time complexity are polynomially bounded in k and the other parameters of the problem, we say that the proof system is *doubly efficient*.

Finally, our work will involve two key primitives, proof systems which allow the verifier Arthur to obtain *objects* (namely, either an approximation $\tilde{\mathbf{p}}$ of the unknown distribution, or a sequence of tagged samples (Definition 2.7) from it). To capture this, we formalize the notion of IP distribution estimator:

Definition 2.6 (IP distribution estimator)**.** An *IP distribution estimator* for a relation $\mathcal{R} \subseteq \Delta(k) \times \mathbb{R}^d$ is a pair (A, M), A has sample access to $\mathbf{p}$, and M is adaptive and knows $\mathbf{p}$. The interaction is over one or more rounds; we write $\langle A^{\mathbf{p}}, M(\mathbf{p})\rangle$ for the verifier view at the end of the interaction. The pair satisfies:

- (Completeness) for every $\mathbf{p} \in \Delta(k)$, $\Pr[\,(\mathbf{p}, \langle A^{\mathbf{p}}, M(\mathbf{p})\rangle) \in \mathcal{R}\,] \geq 2/3$;
- (Soundness) for every $\mathbf{p} \in \Delta(k)$ and every prover strategy $\tilde{M}$,

$$\Pr\Big[\,(\mathbf{p}, \Big\langle A^{\mathbf{p}}, \tilde{M}(\mathbf{p})\Big\rangle) \in \mathcal{R} \text{ or } \Big\langle A^{\mathbf{p}}, \tilde{M}(\mathbf{p})\Big\rangle = \text{reject}\,\Big] \geq 2/3\,.$$

That is, for a honest Merlin leads to a valid output (for $\mathbf{p}$) with high probability; and no matter what strategy a dishonest Merlin uses, with high probability the protocol either outputs a valid output, or reject. *Communication complexity*, *round complexity*, and *efficiency* of the proof system are defined as above, and so is the MA and *public-* and *secret-coin* variants.

In the above, letting $\mathcal{R}$ consist of all pairs $(\mathbf{p}, \mathbf{q})$ with $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) \leq \alpha$ leads to a proof system for distribution learning (for fixed α), and letting $\mathcal{R}$ be the set of pairs $(\mathbf{p}, \mathrm{tag}(\mathbf{p}))$ for all valid sequences of τ-approximate tagged samples for $\mathbf{p}$ leads to a proof system for tagged samples.

2.2 Useful notions and notation

Finally, the following notions will be central to our protocols and their stated guarantees. For any $\tau \in (0, 1]$ and distribution $\mathbf{p}$ over $[k]$, define the (τ, γ)-*bucketing of* $\mathbf{p}$ as the partition of the domain $[k]$ into

$$L := \left\lceil \frac{\log(k/\tau)}{\log(1+\gamma)} \right\rceil - 1 = O\Big(\frac{\log(k/\tau)}{\gamma}\Big) \tag{2}$$

"buckets" $\mathbb{B}_0, \ldots, \mathbb{B}_L$ defined by

$$\mathbb{B}_j := \left\{ x \in [k] \,:\, \mathbf{p}(x) \in \left[\frac{\tau(1+\gamma)^j}{k}, \frac{\tau(1+\gamma)^{j+1}}{k}\right) \right\}, \qquad 0 \leq j \leq L\,, \tag{3}$$

along with the "leftover bucket" $\mathbb{B}_\perp := \{\, x \in [k] \,:\, \mathbf{p}(x) < \frac{\tau}{k} \,\}$.[11] Further, let $\bar{p}_j := \frac{\tau(1+\gamma)^j}{k}$ such that the j-th bucket contains exactly the elements with probability in $[\bar{p}_j, (1+\gamma)\bar{p}_j)$ under $\mathbf{p}$.[12]

[11] As we will see later, a simple argument lets us ignore the leftover bucket, and we will not need to consider it afterwards. We here only define it for completeness, for $\mathbb{B}_\perp, \mathbb{B}_0, \ldots, \mathbb{B}_L$ to form a honest-to-goodness partition of the domain.

[12] For instance, for our public-coin protocol in Section 3.1, we will set $\gamma = O(\tau^3)$.

Definition 2.7 (Tagged samples)**.** For $k \geq 2$, $L \geq 1$, and a partition $\mathbb{B} = (\mathbb{B}_0, \mathbb{B}_1, \ldots, \mathbb{B}_L)$ of the domain $[k]$, a sequence of m *tagged samples* (for $\mathbb{B}$) is an m-tuple

$$(x_i, \pi_i)_{i \in [m]} \in ([k] \times \{0, 1, \ldots, L\})^m .$$

The tuple $(\pi_i)_{i \in [m]}$ then constitutes the *tags* for $(x_i)_{i \in [m]}$.

Definition 2.8. Given $\alpha \in [0, 1]$, we say that a set of tags $(\pi_i)_{i \in [n]}$ are α*-approximate for a tuple* $(x_i)_{i \in [n]}$ *with respect to* $\mathbf{p}$ if

$$(1 - \alpha)\mathbf{p}(x_i) \leq \pi_i \leq (1 + \alpha)\mathbf{p}(x_i)$$

for at least $(1 - \alpha)n$ of the indices $1 \leq i \leq n$.

It is not hard to see that this definition implies the following two measures of accuracy; the first is the one used in [HR22; HR23; Her24]:

$$\frac{1}{n} \sum_{i=1}^{n} \left(1 - \min \left\{ \frac{\mathbf{p}(x_i)}{\pi_i}, \frac{\pi_i}{\mathbf{p}(x_i)} \right\} \right) \leq \alpha. \tag{4}$$

while the second is akin to a "χ^2-type" guarantee (or, more precisely, is a bound on a quantity resembling a Le Cam divergence [PW25, Chapter 7.1]):

$$\frac{1}{n} \sum_{i=1}^{n} \frac{(\mathbf{p}(x_i) - \pi_i)^2}{\mathbf{p}(x_i) + \pi_i} \leq \alpha. \tag{5}$$

3 Generating tagged samples

In this section, we establish the following three theorems. The first is a generalization (and somewhat simplification) of the main result of [Her24]:

Theorem 3.1. *For any* $1 \leq m \leq k$ *and* $n \leq m$ *such that* $n^2 m \gtrsim \frac{k^2 L^2}{\tau^{18}}$*, Algorithm 2 outputs a* τ*-approximate set of* m *tagged samples with probability at least* $96/100$*. The communication complexity is* $O(m \log(k/\tau))$*, and the sample complexity* $O(n)$*.*

The second is the private version:

Theorem 3.2. *For any* $1 \leq m \leq k$ *and* $n \leq m$ *such that* $n^2 m \gtrsim \frac{k^2 L^2}{\tau^{18}}$ *and* $n \gtrsim \frac{kL \log(L)}{m\tau^4 \varepsilon} + \frac{k\sqrt{L \log(L) \log(n)}}{\tau^{5/2}\sqrt{m\varepsilon}}$*, there is an* ε*-Arthur-private protocol which outputs a* τ*-approximate set of* m uniformly distributed *tagged samples with probability at least* $96/100$*. The communication complexity is* $O(m \log(k/\tau))$*, and the sample complexity* $O(n)$*.*

The third shows how protocols enable more efficient testing for Arthur, by making differentially private the corresponding protocol of Herman and Rothblum [HR23].

Theorem 3.3. *For any* m*,* n*, and* s *such that* $m \geq n \geq 2s = \tilde{\Omega}(\sqrt{k}) \operatorname{poly}(1/\tau)$ *and* $\frac{mn}{s} = \tilde{\Omega}(k \log(1/\delta)/\varepsilon^2)$*, there is an* (ε, δ)*-Arthur-private protocol* Π *with the following guarantees. For any (possibly randomized) function* $f \colon ([k] \times [0, 1])^s \to \{0, 1\}$*, the protocol either outputs reject, or a value* $\nu \in \{0, 1\}$*, such that*

1. *If Merlin is honest, the protocol outputs reject with probability at most* $1/6$*;*

2. *Conditioned on the protocol not outputting reject,*

$$|\Pr_{\Pi,\mathbf{p}}[\,\nu=1\,]-\Pr_{x_1,\ldots,x_s\sim\mathbf{p}}[\,f((x_1,\pi(x_1)),\ldots,(x_s,\pi(x_s)))=1\,]|\leq\frac{1}{6}$$

where the probability is taken over the randomness of the protocol, f*, and the samples from* $\mathbf{p}$*, and* $(\pi(x_i))_{i\in[s]}$ *are* τ*-approximate for the tuple* $(x_i)_{i\in[s]}$ *with respect to* $\mathbf{p}$*.*

That is, the protocol either outputs reject, or gives with very high probability the same binary output as any fixed decision rule taking as input a τ*-approximate set of* s *tagged samples from* $\mathbf{p}$*. The communication complexity is* $O(m\log(k/\tau))$*, and the sample complexity* $O(n/\varepsilon)$*.*

As discussed in Section 1.4.4, tagged samples (either drawn from $\mathbf{p}$, or uniformly distributed ones) can then be leveraged to ultra-efficiently test a wide range of properties.

3.1 Public-coin protocol

Overview The high-level idea of the protocol of [Her24] to learn the bucketing is as follows: after sending a sequence σ of elements to Merlin and obtained the (purported) "(bucket) tags" for each of them, Arthur is now in possession of $L+1:=O(\log(k/\tau)/\tau)$ disjoint sets $B_0,\ldots,B_L$, where B_j is the subset of samples tagged as belonging to $\mathbb{B}_j$. He then seeks to verify the following two conditions, where, for bucket $1\leq j\leq L$, and $\gamma=\mathrm{poly}(\tau)$:

1. $\mathbf{p}(B_j)\overset{1\pm\gamma}{\approx}|B_j|\cdot\bar{p}_j$
2. $\sum_{x\in B_j}\mathbf{p}(x)^2\overset{1\pm\gamma}{\leq}|B_j|\cdot\bar{p}_j^2$

If both conditions hold, then this implies that

$$\mathbb{E}_{x\sim\mathbf{u}_{B_j}}[\mathbf{p}(x)]\overset{1\pm\gamma}{\approx}\bar{p}_j\,,\qquad\mathrm{Var}_{x\sim\mathbf{u}_{B_j}}[\mathbf{p}(x)]\leq 4\gamma\cdot\bar{p}_j^2 \tag{6}$$

and by Chebyshev, that most elements in B_j have probabilities within an $1\pm\tau$ multiplicative factor: recalling that $\gamma=\frac{\tau^3}{2}$,

$$\Pr_{x\sim\mathbf{u}_{B_j}}\Big[\,|\mathbf{p}(x)-\mathbb{E}_{x\sim\mathbf{u}_{B_j}}[\mathbf{p}(x)]|>\tau\cdot\bar{p}_j\,\Big]\leq\frac{4\gamma}{\tau^2}=2\tau \tag{7}$$

and so at least $(1-2\tau)|B_j|$ elements x of B_j satisfy

$$(1-\gamma-\tau)\bar{p}_j\leq\mathbf{p}(x)\leq\mathbb{E}_{x\sim\mathbf{u}_{B_j}}[\mathbf{p}(x)]+\tau\cdot\bar{p}_j\leq(1+\gamma+\tau)\bar{p}_j$$

and so $\mathbf{p}(x)\in[(1-2\tau)\bar{p}_j,(1+2\tau)\bar{p}_j]$ since $\gamma\leq\tau$. This implies the tags are 2τ-approximate for σ.

Algorithm 2 Public-Coin Tagged Sample Protocol (based on [Her24])

Require: $k \geq 2$, $\tau \in (0,1)$, sample access to a distribution $\mathbf{p} \in \Delta(k)$ satisfying Assumptions 1 to 3
Parameters: $\gamma \leftarrow \frac{\tau^3}{2}$; $L \leftarrow \left\lceil \frac{\log(k/\tau)}{\log(1+\gamma)} \right\rceil - 1$; n, m such that $1 \leq m \leq k$ and $n^2 m \geq c \cdot \frac{k^2 L^2}{\tau^2 \gamma^4}$ for some absolute constant $c > 0$.

1: Arthur: Draw m samples uniformly from $[k]$, getting a multiset $\sigma = (\sigma_i)_{i \in [m]}$. Send σ to Merlin.
2: Merlin: For every $i \in [m]$, "tag" σ_i with the unique $0 \leq \mathrm{tag}(i) \leq L$ such that $\sigma_i \in \mathbb{B}_{\mathrm{tag}(i)}$. Send back the tagged samples $(\langle \sigma_i, \mathrm{tag}(i) \rangle)_{i \in [m]}$ to Arthur.
3: Arthur: **return** reject if any tag is inconsistent (two samples with the same value but different tags, or a tag corresponding to a bucket not possible). ▷ "Is Merlin dishonest *and* sloppy?"
4: Arthur: For every $0 \leq j \leq L$, set ▷ Set of samples purported to be in $\mathbb{B}_j$

$$B_j = \{\, \sigma_i : \; i \in [m], \mathrm{tag}(i) = j \,\} .$$

5: Arthur: Draw two new multisets X, Y of n independent samples from $\mathbf{p}$. For every $0 \leq j \leq L$, define the multisets

$$X_j \leftarrow (x \in X : x \in B_j) , \qquad Y_j \leftarrow (y \in Y : y \in B_j) , \qquad Z_j \leftarrow ((x,y) \in X_j \times Y_j : x = y)$$

6: Arthur: for every possible bucket $0 \leq j \leq L$:
7: **if** $|X_j| < \frac{1}{8} \frac{n\tau m}{kL}$ **then return** reject ▷ Check all $\mathbb{B}_j$ have mass $\Omega(\tau/L)$
8: **else if** $||X_j| - n|B_j|\bar{p}_j| > \frac{\gamma}{2} n|B_j|\bar{p}_j$ **then return** reject ▷ Check all B_j have mass $\approx |B_j|\bar{p}_j$
9: **else if** $|Z_j| > (1 + \frac{\gamma}{2}) n^2 |B_j|(1+\gamma)^2 \bar{p}_j^2$ **then return** reject ▷ Check all B_j have small collision probability
10: Arthur: **return** $((\sigma_i, \mathrm{tag}(i)))_{i \in [m]}$.

3.1.1 Setting the stage

We here introduce several assumptions on the inputs to the algorithm. We then describe the process to make an arbitrary input distribution satisfy these assumptions in Section 3.3.

Assumption 1 (No probability is too small)**.** For each $x \in \mathcal{X}$, $\mathbf{p}(x) \geq \tau/k$.

Assumption 2 (No probability is too large)**.** For each $x \in \mathcal{X}$, $\mathbf{p}(x) < 1/n$.

Assumption 3 (No bucket is negligible)**.** Every possible bucket $\mathbb{B}$ satisfies $\mathbf{p}(\mathbb{B}) \geq \frac{\tau}{L}$, where a bucket is said to be *possible* if it is compatible with Assumption 2.

Note that Assumption 2 implies that only the first $O(\frac{1}{\tau} \log \frac{k}{\tau n})$ buckets are possible; or, put differently, that the last $\Theta(\log(n)/\tau)$ buckets are empty.

We start with an easy consequence of these assumptions: namely, that each possible bucket has non-negligible mass.

Claim 3.4. *For every $0 \leq j \leq L$ such that $\mathbb{B}_j$ is possible, $|\mathbb{B}_j| \geq \frac{n\tau}{L}$.*

Proof. This follows from Assumptions 2 and 3. □

In what follows, we define

$$\Lambda_{m,k} := 1 - \left(1 - \frac{1}{k}\right)^m \leq \frac{m}{k} , \tag{8}$$

and observe that $\Lambda_{m,k} \le \frac{m}{k}$ and $\Lambda_{m,k} \approx \frac{m}{k}$ for $m \ll k$. We will restrict ourselves to m such that

$$\frac{m}{2k} \le \Lambda_{m,k} \le \frac{m}{k} \tag{9}$$

which holds for any $1 \le m \le ck$, where $c > 0$ is an absolute constant. (Moreover, one can take $c = 1$.)

After receiving the tagged uniform sample $(\sigma_i, \tilde{\mathbf{p}}(\sigma_i))_{\sigma_i \in \sigma}$ from Merlin, Arthur partitions it into the sets $(B_j)_{j \in [L]}$ where, for $0 \le j \le L$, where $L := \left\lceil \frac{\log(k/\tau)}{\log(1+\gamma)} \right\rceil - 1 = O\left(\frac{\log(k/\tau)}{\gamma}\right)$,

$$B_j := \{\, s \in S : \ \tilde{\mathbf{p}}(s) \in [\bar{p}_j, \bar{p}_{j+1}) \,\}\,. \tag{10}$$

where, recalling Section 2, $\bar{p}_j := \frac{\tau(1+\gamma)^j}{k}$. Note that by Assumption 1 there is no element with probability in $[0, \tau/k)$ (and thus no corresponding tag is needed). To verify these buckets are consistent with the distribution $\mathbf{p}$, Arthur then draws two new samples X and Y from[13] $\mathbf{p}^{\otimes n}$, and defines

$$X_j \leftarrow (x \in X : x \in B_j)\,, \quad Y_j \leftarrow (y \in Y : y \in B_j)\,, \quad Z_j \leftarrow ((x,y) \in X_j \times Y_j : x = y)$$

for $0 \le j \le L-1$; note that, writing $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$, this implies

$$|X_j| = \sum_{i=1}^{n} \mathbb{1}_{\{x_i \in B_j\}}\,, \quad |Y_j| = \sum_{i=1}^{n} \mathbb{1}_{\{y_i \in B_j\}}\,,$$
$$|Z_j| = \sum_{1 \le i,i' \le n} \mathbb{1}_{\{x_i = y_{i'}\}} \mathbb{1}_{\{x_i \in B_j\}} = \sum_{1 \le i,i' \le n} \sum_{b \in B_j} \mathbb{1}_{\{x_i = y_{i'} = b\}}\,.$$

3.1.2 The algorithm, and its analysis

Arthur's verification After the basic check that all tags are valid (in particular, no sample was tagged by Merlin as belonging to a bucket which is not possible), recall that Arthur does the following, for every $0 \le j \le L$ corresponding to a possible bucket:

1. Output reject if $|X_j| < \frac{1}{8} \cdot n \cdot \frac{\tau}{L} \cdot \frac{m}{k}$. (Check that each real bucket has mass $\Omega(\tau/L)$)
2. Output reject if $|X_j| < (1 - \frac{\gamma}{2}) \cdot n|B_j|\bar{p}_j$ or $|X_j| > (1 + \frac{\gamma}{2}) \cdot n|B_j|(1+\gamma)\bar{p}_j$. (Check that each purported bucket has mass $\approx |B_j|\bar{p}_j$)
3. Output reject if $|Z_j| > (1 + \frac{\gamma}{2}) \cdot n^2|B_j|(1+\gamma)^2\bar{p}_j^2$. (Check that each purported bucket has small collision probability, $\approx |B_j|\bar{p}_j^2$)

Arthur's verification facts

Claim 3.5. *For every $0 \le j \le L$, $|X_j|$ is a $\mathrm{Bin}(n, \mathbf{p}(B_j))$ random variable, and $|Z_j|$ satisfies*

$$\mathbb{E}_{X,Y}[|Z_j|] = n^2 \sum_{x \in B_j} \mathbf{p}(x)^2, \qquad \mathrm{Var}_{X,Y}[|Z_j|] \le \mathbb{E}_{X,Y}[|Z_j|] + 2\mathbb{E}_{X,Y}[|Z_j|]^{3/2}$$

[13] Note that we could do a similar analysis only taking a new set of samples X. The argument would go through, with only some relatively minor (but mildly annoying) differences, and save a factor 2 in the sample complexity. We believe that these mildly annoying differences would sufficiently impede the readability that losing this factor 2 is warranted.

Proof. We will set $F_2(j) := \sum_{x\in B_j} \mathbf{p}(x)^2$ and $F_3(j) := \sum_{x\in B_j} \mathbf{p}(x)^3$ for ease of notation. The claimed distribution of X_j (and thus its independent copy Y_j) is immediate, and from there

$$\mathbb{E}_{X,Y}[|Z_j|] = \sum_{(x,y)\in X\times Y} \sum_{z\in B_j} \mathbb{E}_{X,Y}[\mathbb{1}_{\{x=y=z\}}] = \sum_{(x,y)\in X\times Y} \sum_{z\in B_j} \mathbf{p}(z)^2 = n^2 \sum_{z\in B_j} \mathbf{p}(z)^2 = n^2 F_2(j)\,.$$

Turning to the variance, we have, following the same analysis as for the variance of the bipartite collision tester in [Can22, Section 2.17] (leading to Eq. 2.37),

$$\mathrm{Var}_{X,Y}[|Z_j|] \le n^2 F_2(j) + 2n^3(F_3(j) - F_2(j)^2)\,.$$

Dropping the last (non-positive) term, and using that $F_3(j) \le F_2(j)^{3/2}$ by monotonicity of ℓ_p norms, we get

$$\mathrm{Var}_{X,Y}[|Z_j|] \le n^2 F_2(j) + 2n^3 F_2(j)^{3/2} = \mathbb{E}_{X,Y}[|Z_j|] + 2\mathbb{E}_{X,Y}[|Z_j|]^{3/2}\,,$$

as claimed. □

Completeness: Honest Merlin strategy. We will show here that there is a strategy for Merlin (namely, label the elements accurately) which guarantees Tests 1, 2, and 3 will pass with overall probability $97/100$. This is done in Lemma 3.7 below.

Claim 3.6. *Assume Merlin is honest. Then, for every $0 \le j \le L$, $|B_j|$ satisfies*

$$\Pr_S\left[\, |B_j| \overset{1\pm\gamma}{\approx} \Lambda_{m,k}|\mathbb{B}_j| \,\right] \ge \frac{99}{100}\,.$$

Moreover, whenever $|B_j| \overset{1\pm\gamma}{\approx} \Lambda_{m,k}|\mathbb{B}_j|$, we have $\mathbf{p}(B_j) \overset{1\pm2\gamma}{\approx} \Lambda_{m,k}\mathbf{p}(\mathbb{B}_j)$.

Proof. Assume Merlin correctly tags the samples in σ, and fix any $0 \le j \le L$. We have that

$$|B_j| = \sum_{x\in\mathbb{B}_j} \mathbb{1}_{x\in S}$$

where, for any $x \in [k]$,

$$\mathbb{E}[\mathbb{1}_{x\in S}] = 1 - \left(1 - \frac{1}{k}\right)^m$$

and so by (9)

$$\frac{m}{2k}|\mathbb{B}_j| \le \mathbb{E}[|B_j|] = \Lambda_{m,k}|\mathbb{B}_j| \le \frac{m}{k}|\mathbb{B}_j|\,.$$

Moreover, the random variables $(\mathbb{1}_{x\in\sigma})_{x\in\mathbb{B}_j}$ are negatively associated, and thus satisfy the Chernoff bound. Thus,

$$\Pr_S[\,||B_j| - \mathbb{E}[|B_j|]| > \gamma\mathbb{E}[|B_j|]\,] \le 2e^{-\frac{1}{2}\gamma^2\mathbb{E}[|B_j|]} \le 2e^{-\frac{1}{2}\gamma^2\frac{mn\tau}{2Lk}} \le \frac{1}{100(L+1)}$$

using at the end Claim 3.4 and our setting of m, n. A union bound over all $0 \le j \le L$ concludes the claim, as the "moreover" part follows from the definition of the (honest) buckets, since $\mathbf{p}(B_j) \overset{1\pm\tau}{\approx} |B_j|\bar{p}_j$ and $\mathbf{p}(\mathbb{B}_j) \overset{1\pm\tau}{\approx} |\mathbb{B}_j|\bar{p}_j$.

□

Lemma 3.7. *Assume Merlin is honest. Then, with probability at least $96/100$ over the random choice of σ, X, Y, all the following simultaneously holds for all $0 \le j \le L$:*

1. $|X_j| \ge \frac{1}{8}\frac{n\tau m}{Lk}$;
2. $(1-\frac{\gamma}{2})\cdot n|B_j|\bar{p}_j \le |X_j| \le (1+\frac{\gamma}{2})\cdot n|B_j|(1+\gamma)\bar{p}_j$;
3. $|Z_j| \le (1+\frac{\gamma}{2})\cdot n^2|B_j|(1+\gamma)^2\bar{p}_j^2$.

Proof. Assume Merlin correctly tags the samples in σ, and that the conclusion of Claim 3.6 holds (which happens with probability at least $99/100$). Then, from Assumption 3, we get

$$\mathbb{E}[|X_j|] = n\mathbf{p}(B_j) \ge (1-2\gamma)\frac{nm}{2k}\mathbf{p}(\mathbb{B}_j) \ge \frac{nm\tau}{4Lk} \tag{11}$$

for $\gamma \le 1/4$. We hereafter assume this holds. test 1 follows from a Chernoff bound: as $|X_j| \sim \mathrm{Bin}(n, \mathbf{p}(B_j))$ by Claim 3.5, we have

$$\Pr_X\left[\,|X_j| < \frac{1}{8}\frac{n\tau m}{Lk}\,\right] \le \Pr_X\left[\,|X_j| < \frac{1}{2}\mathbb{E}[|X_j|]\,\right] \le e^{-\frac{1}{8}n\mathbf{p}(B_j)} \le e^{-\frac{nm\tau}{16Lk}} \le \frac{1}{100(L+1)}\,,$$

where we used that $nm = \Omega(Lk\log(L)/\tau)$. The "whenever" holds for all j with probability at least $99/100$ by Claim 3.6; a union bound then shows test 1 holds for all j with probability $99/100$.

Turning to test 2, fix any j. Note that if Merlin is honest then $\mathbf{p}(x) \in [\bar{p}_j, (1+\gamma)\bar{p}_j]$ for all $x \in B_j$, and so

$$n|B_j|\bar{p}_j \le \underbrace{\mathbb{E}[|X_j|]}_{n\mathbf{p}(B_j)} \le (1+\gamma)n|B_j|\bar{p}_j\,. \tag{12}$$

Thus,

$$\Pr_X\big[\,|X_j| \notin [(1-\tfrac{\gamma}{2})n|B_j|\bar{p}_j, (1+\tfrac{\gamma}{2})n|B_j|(1+\gamma)\bar{p}_j]\,\big] \le \Pr_X\big[\,|X_j| \notin [(1-\tfrac{\gamma}{2})\mathbb{E}[|X_j|], (1+\tfrac{\gamma}{2})\mathbb{E}[|X_j|]]\,\big]$$

and, again by a Chernoff bound, recalling that $\mathbb{E}[|X_j|] = n\mathbf{p}(B_j) \ge \frac{nm\tau}{4Lk}$ by Eq. (11),

$$\Pr_X\big[\,||X_j| - \mathbb{E}[|X_j|]| > \tfrac{\gamma}{2}\mathbb{E}[|X_j|]\,\big] \le 2e^{-\frac{\gamma^2}{12}\mathbb{E}[|X_j|]} \le 2e^{-\frac{\gamma^2 nm\tau}{48Lk}} \le \frac{1}{100(L+1)}\,,$$

where we used that $nm = \Omega(Lk\log(L)/(\gamma^2\tau))$. A union bound then shows test 2 holds for all j with probability $99/100$.

Finally, in view of establishing test 3, Claim 3.5 guarantees that, in the honest-Merlin case,

$$n^2|B_j|\bar{p}_j^2 \le \mathbb{E}_{X,Y}[|Z_j|] \le n^2|B_j|(1+\gamma)^2\bar{p}_j^2$$

Now, since $\bar{p}_j \ge \tau/k$ and $|B_j|\bar{p}_j \ge \frac{\mathbf{p}(B_j)}{1+\gamma}$ (Eq. (12)),

$$\mathbb{E}_{X,Y}[|Z_j|] \ge n^2|B_j|\bar{p}_j^2 \ge \frac{n^2\tau}{(1+\gamma)k}\cdot\mathbf{p}(B_j) \ge \frac{n^2\tau}{(1+\gamma)k}\cdot\frac{m\tau}{4Lk} \ge \frac{n^2m\tau^2}{8Lk^2}\,, \tag{13}$$

the second inequality by Eq. (11). Altogether, by Chebyshev and the variance bound from Claim 3.5,

$$\begin{aligned}\Pr_{X,Y}\left[\,|Z_j| > (1+\tfrac{\gamma}{2})n^2|B_j|(1+\gamma)^2\bar{p}_j^2\,\right] &\le \Pr_{X,Y}\left[\,|Z_j| > (1+\tfrac{\gamma}{2})\mathbb{E}_{X,Y}[|Z_j|]\,\right] \\ &\le \frac{8}{\gamma^2\mathbb{E}_{X,Y}[|Z_j|]} + \frac{16}{\gamma^2\mathbb{E}_{X,Y}[|Z_j|]^{1/2}} \\ &\le \frac{1}{100(L+1)}\,,\end{aligned}$$

the last inequality using the bound from Eq. (13), along with the fact that $n^2m = \Omega(L^2k^2/(\gamma^4\tau^2))$. A union bound then shows test 3 holds for all j with probability $99/100$. The overall lemma follows by a union bound over the events of Claim 3.6 and tests 1 to 3. □

Soundness. Conversely, we will show that no matter what strategy Merlin uses, if all of Tests 1, 2, and 3 pass then the purported buckets B_j are good:

Lemma 3.8. *For every σ sent by Arthur, and any strategy from Merlin, if Tests 1, 2, and 3 pass with overall probability greater than $1/100$ (over the choice of X, Y), then buckets $(B_j)_{0\le j\le L}$ satisfy the following for all j:*

- $\mathbf{p}(B_j) \in [(1-\gamma)|B_j|\bar{p}_j, (1+\gamma)|B_j|(1+\gamma)\bar{p}_j]$;
- $\sum_{x\in B_j}\mathbf{p}(x)^2 \le (1+\gamma)|B_j|(1+\gamma)^2\bar{p}_j^2$.

Proof. We will establish this by combining the three claims below. Specifically,

- $\min_{0\le j\le L}\mathbf{p}(B_j) \ge \frac{m\tau}{400Lk}$ (otherwise Test 1 rejects with probability $99/100$, by Claim 3.9)
- $\mathbf{p}(B_j) \in [(1-\gamma)|B_j|\bar{p}_j, (1+\gamma)|B_j|(1+\gamma)\bar{p}_j]$ (otherwise Test 2 rejects with probability $99/100$, by Claim 3.10)
- $\sum_{x\in B_j}\mathbf{p}(x)^2 \le (1+\gamma)|B_j|(1+\gamma)^2\bar{p}_j^2$ (otherwise Test 3 rejects with probability $99/100$, by Claim 3.11).

Claim 3.9 (All purported buckets must have reasonable mass)**.** *For every σ sent by Arthur, and any strategy from Merlin, suppose there exists $0 \le j \le L$ such that $\mathbf{p}(B_j) < \frac{m\tau}{800Lk}$. Then Test 1 outputs reject with probability at least $99/100$ (over the choice of X).*

Proof. Fix any $0 \le j \le L$ such that $\mathbf{p}(B_j) < \frac{m\tau}{800Lk}$. For this particular j, since $\mathbb{E}_X[|X_j|] = n\mathbf{p}(B_j)$ by Claim 3.5, we have, by Markov's inequality,

$$\Pr_X\left[\,|X_j| \ge \frac{mn\tau}{8Lk}\,\right] \le \frac{1}{100}$$

and so Test 1 outputs reject on this j with probability at least $99/100$. □

Given that Test 1 will reject otherwise, we can then assume $\min_{0\le j\le L}\mathbf{p}(B_j) \ge \frac{m\tau}{800Lk}$. The next lemma analyzes Test 2:

Claim 3.10 (All purported buckets must have approximately the right mass)**.** *For every σ sent by Arthur, and any strategy from Merlin such that $\min_{0\leq j\leq L} \mathbf{p}(B_j) \geq \frac{m\tau}{800Lk}$, suppose that there exists $0 \leq j \leq L$ such that*

$$\mathbf{p}(B_j) \notin [(1-\gamma)|B_j|\bar{p}_j, (1+\gamma)|B_j|(1+\gamma)\bar{p}_j]\,.$$

Then Test 2 outputs reject with probability at least $99/100$ (over the choice of X).

Proof. Fix any $0 \leq j \leq L$ such that $\mathbf{p}(B_j) \notin [(1-\gamma)|B_j|\bar{p}_j, (1+\gamma)|B_j|(1+\gamma)\bar{p}_j]$, and recall that $|X_j|$ is Binomial with mean $n\mathbf{p}(B_j)$ by Claim 3.5.

- On the one hand, if $\mathbf{p}(B_j) < (1-\gamma)|B_j|\bar{p}_j$, then, by a Chernoff bound,

$$\Pr_X[\,|X_j| \geq (1-\tfrac{\gamma}{2})n|B_j|\bar{p}_j\,] \leq \Pr_X[\,|X_j| \geq (1+\tfrac{\gamma}{2})\mathbb{E}_X[|X_j|]\,] \leq e^{-\frac{\gamma^2}{12}\mathbb{E}_X[|X_j|]} \leq e^{-\frac{\gamma^2 nm\tau}{9600Lk}} \leq \frac{1}{100}\,,$$

  using that $\frac{1-\frac{\gamma}{2}}{1-\gamma} \geq 1+\frac{\gamma}{2}$, and then our lower bound on $\mathbf{p}(B_j)$ and $nm = \Omega(Lk/(\gamma^2\tau))$.

- On the other hand, if $\mathbf{p}(B_j) > (1+\gamma)|B_j|(1+\gamma)\bar{p}_j$, then, similarly

$$\Pr_X[\,|X_j| \leq (1+\tfrac{\gamma}{2})n|B_j|(1+\tau)\bar{p}_j\,] \leq \Pr_X[\,|X_j| \leq (1-\tfrac{\gamma}{3})\mathbb{E}_X[|X_j|]\,] \leq e^{-\frac{\gamma^2}{18}\mathbb{E}_X[|X_j|]} \leq \frac{1}{100}\,,$$

  using that $\frac{1+\frac{\gamma}{2}}{1+\gamma} \leq 1-\frac{\gamma}{3}$ for $\gamma \leq 1/2$, and then again our lower bound on $\mathbf{p}(B_j)$ and $nm = \Omega(Lk/(\gamma^2\tau))$.

In both cases, Test 2 outputs reject on this j with probability at least $99/100$. □

Given that Test 1 or Test 2 will reject otherwise, we can then assume both $\min_{0\leq j\leq L} \mathbf{p}(B_j) \geq \frac{m\tau}{800Lk}$ and $\min_{0\leq j\leq L} |B_j|\bar{p}_j \geq \frac{m\tau}{3200Lk}$.[14] The next lemma analyzes Test 3:

Claim 3.11 (All purported buckets must have small enough collision probability)**.** *For every σ sent by Arthur, and any strategy from Merlin such that $\min_{0\leq j\leq L} |B_j|\bar{p}_j \geq \frac{m\tau}{3200Lk}$, suppose that there exists $0 \leq j \leq L$ such that*

$$\sum_{x\in B_j} \mathbf{p}(x)^2 > (1+\gamma)|B_j|(1+\gamma)^2\bar{p}_j^2\,.$$

Then Test 3 outputs reject with probability at least $99/100$ (over the choice of X, Y).

Proof. Fix any $0 \leq j \leq L$ such that $\sum_{x\in B_j} \mathbf{p}(x)^2 > (1+\gamma)|B_j|(1+\gamma)^2\bar{p}_j^2$. By Claim 3.5 and Chebyshev,

$$\begin{aligned}
\Pr_{X,Y}\Big[\,|Z_j| \leq (1-\tfrac{\gamma}{2})n^2|B_j|(1+\gamma)^2\bar{p}_j^2\,\Big] &\leq \Pr_{X,Y}[\,|Z_j| \leq (1-\tfrac{\gamma}{2})\mathbb{E}_{X,Y}[|Z_j|]\,] \\
&\leq \frac{4}{\gamma^2\mathbb{E}_{X,Y}[|Z_j|]} + \frac{8}{\gamma^2\mathbb{E}_{X,Y}[|Z_j|]^{1/2}} \\
&\leq \frac{4}{\gamma^2 n^2|B_j|\bar{p}_j^2} + \frac{8}{\gamma^2 n|B_j|^{1/2}\bar{p}_j} \\
&\leq \frac{12800Lk^2}{\gamma^2 n^2 m\tau^2} + \frac{500L^{1/2}k}{\gamma^2 nm^{1/2}\tau} \\
&\leq \frac{1}{100}\,,
\end{aligned}$$

[14] This latter condition is implied by the combination of $\min_{0\leq j\leq L} \mathbf{p}(B_j) \geq \frac{m\tau}{800Lk}$ and $\mathbf{p}(B_j) \in [(1-\gamma)|B_j|\bar{p}_j, (1+\gamma)|B_j|(1+\tau)\bar{p}_j]$ for all j.

using our lower bound on $|B_j|\bar{p}_j$, the fact that $\bar{p}_j \geq \tau/k$, and $n^2 m = \Omega(Lk^2/(\gamma^4\tau^2))$. Thus, Test 3 outputs reject on this j with probability at least $99/100$. □

This concludes the proof of Lemma 3.8. □

3.1.3 Privacy for the public-coin protocol

Lemma 3.12. *There exists an ε-Arthur-DP algorithm for tests 1 to 3 whenever*

$$n \geq \max\left\{\frac{640kL\log(200(L+1))}{\gamma m\tau\varepsilon}, \frac{36kL^{1/2}\log^{1/2}(200(L+1))\log^{1/2}(100n)}{\gamma^{1/2}\tau m^{1/2}\varepsilon^{1/2}}\right\},$$

or

$$n = \Omega\left(\frac{kL\log(L)}{\gamma m\tau\varepsilon} + \frac{k\sqrt{L\log(L)\log(n)}}{\tau\sqrt{\gamma m\varepsilon}}\right).$$

Proof. Since the samples sent to Merlin are uniformly random, there are no privacy considerations at that step. Instead we have to privately compute the test statistics $|X_j|$ and $|Z_j|$. First, notice that learning each $|X_j|$ up to maximum (also known as simultaneous, or ℓ_∞) error

$$\alpha_1 = \min\left\{\frac{nm\tau}{16Lk}, \frac{\gamma}{4}n|B_j|\frac{\tau(1+\tau)^{j+1}}{k}\right\} \geq \frac{\gamma nm\tau}{64Lk}$$

gives enough accuracy for tests 1 and 2. Since the vector $(|X_j|)_{0\leq j\leq L}$ has sensitivity 2, this can be achieved using Laplace Histograms (see [BV19]) with failure probability $1/100$ and sample complexity

$$n \geq \frac{640kL\log(200(L+1))}{\gamma m\tau\varepsilon} \geq \frac{5\log(200(L+1))}{(\alpha_1/n)(\varepsilon/2)},$$

that is, $n = \tilde{\Omega}(kL/(\gamma m\tau\varepsilon))$.

We now proceed to investigate test 3. In fact, without careful restriction or analysis, $|Z_j|$ can have sensitivity as high as n. This occurs in the case that $Y = (y, \ldots, y)$ and we move from X to neighbouring dataset X' that contains one more, or one fewer, copy of y. Naturally this seems quite unlikely. We ensure that $\forall x\, \mathbf{p}(x) \leq 1/n$. Let Δ be the maximum sensitivity of $|Z_j|$ that we will bound using the fact above. Once we have this we can clearly use the same "histogram" approach as for the previous tests.

Towards bounding Δ, we will bound the maximum of $\hat{f}$, the empirical counts (histogram) of samples from X (or Y by the same argument). Using a standard "balls in bins" approach, one can see that

$$\Pr_X\left[\max_{x\in[k]}\hat{f}(x) > 2\log(100n)\right] \leq \frac{1}{100} \tag{14}$$

and as such the probability that either X_j or Y_j have more than this many repetitions is at most $2/100$. In the case that any element appears more than $2\log(100n)$, drop duplicates until we are below this quantity. One can quickly see that this will not incur any privacy loss, and so in the case that we have to threshold any point we will only incur additional error. Clearly the sensitivity after this preprocessing is

$$\Delta = 2\log(100n).$$

Using the fact that the previous test passed, we can also see that

$$\frac{\tau m}{8Lk} \leq \frac{|X_j|}{n} \leq 2|B_j|\bar{p}_j$$

so we can simplify

$$\alpha_2 = \frac{\gamma}{4}n^2(|B_j|\bar{p}_j)(1+\tau)^2\bar{p}_j \geq \frac{\gamma\tau mn^2}{64Lk}\bar{p}_j \geq \frac{\gamma\tau^2 mn^2}{64Lk^2}\,.$$

Learning a histogram up to maximum error α_2, when the sensitivity equals Δ requires

$$n \geq \Delta\frac{640k^2L\log(200(L+1))}{\gamma\tau^2 mn\varepsilon} \geq \Delta\frac{5\log(200(L+1))}{(\alpha_2/n)(\varepsilon/2)}$$

or, equivalently,

$$n \geq \frac{36kL^{1/2}\log^{1/2}(200(L+1))\log^{1/2}(100n)}{\gamma^{1/2}\tau m^{1/2}\varepsilon^{1/2}}\,.$$

Combing the bounds for each test implies a final sample complexity of

$$n \geq \max\left\{\frac{640kL\log(200(L+1))}{\gamma m\tau\varepsilon}, \frac{36kL^{1/2}\log^{1/2}(200(L+1))\log^{1/2}(100n)}{\gamma^{1/2}\tau m^{1/2}\varepsilon^{1/2}}\right\},$$

concluding the proof. □

3.2 Private-coin protocol

In this section, we will establish Theorem 3.3, building upon the (non-private) secret-coin tagged sample protocol of Herman and Rothblum [HR23]. In the overview below, we first recall this protocol (Algorithm 3), before discussing how to make it Arthur-private (Algorithm 4). We then provide the proof of Theorem 3.3 assuming guarantees for one of the building blocks, which we finally analyze in Section 3.2.1.

Overview In the secret-coin setting, the high-level idea of the protocol of [HR23] to learn the bucketing is as follows: as in the public-coin one covered in Section 3.1, having sent a sequence σ of elements to Merlin and obtained the (purported) "(bucket) tags" for each of them, Arthur is now in possession of $L+1 := O(\log(k/\tau)/\tau)$ disjoint *multi*sets $\tilde{B}_1, \ldots, \tilde{B}_L$, where $\tilde{B}_j$ is the subsequence of samples tagged as belonging to $\mathbb{B}_j$. The key difference now (besides looking at multisets instead of sets) is that instead of having drawn σ uniformly at random, it is now a mixture of elements drawn uniformly from the domain *and of elements drawn from* $\mathbf{p}$: namely, σ is a set of $m+n$ samples from the mixture

$$\frac{m}{m+n}\mathbf{u}_k + \frac{n}{m+n}\mathbf{p}\,.$$

Of course, Merlin is not told which of the $m+n$ elements come from which distribution, and so, in tagging them, a malicious Merlin trying to fool Arthur would have to guess.

Letting $\tilde{B}_j^{\mathbf{u}}$ and $\tilde{B}_j^{\mathbf{p}}$ denote the parts of $\tilde{B}_j$ coming from $\mathbf{u}_k$ and $\mathbf{p}$, respectively, Arthur then seeks to verify the following two conditions.

1. $\sum_{x\in\tilde{B}_j^{\mathbf{p}}} \mathbf{p}(x) \overset{1\pm O(\tau)}{\approx} \left|\tilde{B}_j^{\mathbf{p}}\right| \cdot \bar{p}_j$
2. $\frac{m}{k}|\tilde{B}_j^{\mathbf{u}}| \overset{1\pm O(\tau)}{\approx} \frac{1}{n\bar{p}_j}|\tilde{B}_j^{\mathbf{p}}|$

If both conditions hold, then Herman and Rothblum [HR23] show[15] that the set of tagged samples obtained is τ-accurate (specifically, the portion obtained from $\mathbf{p}$):

Theorem 3.13 ([HR23, Main theorem, mildly generalized]). *For any m, n, n' and $\tau \in (0,1]$ such that $m \geq n$ and $n, n' \geq \tilde{\Omega}(\sqrt{k})\,\mathrm{poly}(1/\tau)$, Algorithm 3 outputs a τ-approximate set of n tagged samples* distributed according to $\mathbf{p}$ *with probability at least $9/10$. The communication complexity is $O(m\log(k/\tau))$, and the sample complexity $O(n+n')$.*

Algorithm 3 Secret-Coin Tagged Sample Protocol ([HR23], restated)

Require: $k \geq 2$, $\tau \in (0,1)$, sample access to a distribution $\mathbf{p} \in \Delta(k)$ satisfying Assumptions 1 to 3

Parameters: $L \leftarrow \left\lceil \frac{\log(k/\tau)}{\log(1+\tau)} \right\rceil - 1$; n, n', m such that $m \geq n$ and $n, n' \geq \sqrt{k}\,\mathrm{poly}(L, 1/\tau)$; $N \leftarrow m+n$.

1: Arthur: Draw N samples from the mixture

$$\mathbf{q} := \frac{m}{N}\mathbf{u}_k + \frac{n}{N}\mathbf{p}$$

remembering which samples came from which component of the mixture. This gives a multiset $\sigma = (\sigma_i)_{i\in[N]}$, whose partition $\sigma = \sigma^{\mathbf{u}} \cup \sigma^{\mathbf{p}}$ is known to Arthur. Send σ to Merlin.

2: Merlin: For every $i \in [N]$, "tag" σ_i with the unique $0 \leq \mathrm{tag}(i) \leq L$ such that $\sigma_i \in \mathbb{B}_{\mathrm{tag}(i)}$. Send back the tagged samples $(\langle \sigma_i, \mathrm{tag}(i)\rangle)_{i\in[N]}$ to Arthur.

3: Arthur: **return** reject if any tag is inconsistent (two samples with the same value but different tags, or a tag corresponding to a bucket not possible). ▷ "Is Merlin dishonest *and* sloppy?"

4: Arthur: For every $0 \leq j \leq L$, set ▷ Multiset of samples purported to be in $\mathbb{B}_j$

$$\tilde{B}_j^{\mathbf{p}} = (\sigma_i \in \sigma^{\mathbf{p}} : i \in [N], \mathrm{tag}(i) = j), \tilde{B}_j^{\mathbf{u}} = (\sigma_i \in \sigma^{\mathbf{u}} : i \in [N], \mathrm{tag}(i) = j),$$

and $\tilde{B}_j = \tilde{B}_j^{\mathbf{p}} \cup \tilde{B}_j^{\mathbf{p}}$.

5: Arthur: Draw a new multiset X of n' independent samples from $\mathbf{p}$. For every $0 \leq j \leq L-1$, define the multiset $X_j \leftarrow (x \in X : x \in \tilde{B}_j^{\mathbf{p}})$.

6: Arthur: for every possible bucket $0 \leq j \leq L-1$: ▷ $c_1, c_2, c_3 > 0$ are absolute constants.

7: **if** $|\tilde{B}_j^{\mathbf{p}}| < c_1 \frac{n\tau}{L}$ **then return** reject

8: **else if** $\frac{mn\bar{p}_j}{k}|\tilde{B}_j^{\mathbf{u}}|/|\tilde{B}_j^{\mathbf{p}}| \notin [(1-c_3\cdot\tau),(1+c_3\cdot\tau)]$ **then return** reject

9: **else if** $||X_j| - n'|\tilde{B}_j^{\mathbf{p}}|\bar{p}_j| > c_2 \cdot \tau n'|\tilde{B}_j^{\mathbf{p}}|\cdot\bar{p}_j$ **then return** reject

10: Arthur: **return** $((\sigma_i^{\mathbf{p}}, \mathrm{tag}(i)))_{i\in|\sigma^{\mathbf{p}}|}$. ▷ Return the tagged samples which came from $\mathbf{p}$

Our goal is to make this Arthur-private. To do so, note that there are 4 parts which might leak information about Arthur's samples in different ways: in particular, there are two (disjoint)

[15] While their argument was phrased for the case $m = n$, it generalizes to the more general setting $m > n$ which we will require to ensure privacy. Moreover, their argument establishes accuracy of the tagged samples according to Eq. (4): one can check [HR23, Claim 5.3] to see that it extends to Definition 2.8, up to a constant factor in the accuracy.

Algorithm 4 Making the secret-coin algorithm (Algorithm 3) Arthur-DP

Require: multiset S of $2n'$ i.i.d. samples from $\mathbf{p}$, where $n' \geq 100n \geq 200s$; $f : [k]^s \times [0,1] \to \{0,1\}$.

1: Arthur: Pick a multiset S_1 of $100n$ elements uniformly at random from the first n' elements of S. Let S_2 be the second half of S (*i.e.*, multiset of n' samples).

2: Arthur: Draw N i.i.d. samples from the mixture

$$\mathbf{q} = \frac{m}{N}\mathbf{u}_k + \frac{n}{N}\mathbf{p}$$

using S_1 when sampling from $\mathbf{p}$. If more than $2n$ or fewer than $n/2$ samples from $\mathbf{p}$ are needed, **return** reject. Otherwise, let σ be the multiset of N samples. ▷ Rejects w.p. at most $1/10$.

3: Arthur: apply RR and shuffling to σ, as detailed in Theorem 3.14, for $T \leftarrow 2s$ and parameters $\frac{\varepsilon}{2}$, δ. Let σ' be the resulting sample of size $N + m$. ▷ σ' contains $\approx T$ samples from $\mathbf{p}$.

4: Arthur: if fewer than s samples from $\mathbf{p}$ remain in σ', **return** reject with probability $9/10$, and a uniformly random bit $b \in \{0,1\}$ otherwise. ▷ Rejects w.p. at most $1/10$.

5: Arthur and Merlin: run the protocol of Algorithm 3, with the following modifications:

- Arthur sends σ' for Merlin to tag, instead of σ.
- Upon receiving the tagged σ', Arthur removes the m additional uniform samples added in l. 3 to only keep the tagged σ for the rest of the protocol.
- Arthur does the check of l. 9 of Algorithm 3 in an ε-DP fashion (with respect to the multiset X obtained in l. 5 of Algorithm 3), as described in Lemma 3.16.

6: **if** the protocol of Algorithm 3 outputted a multiset of tagged samples **then**

7: run f on the first s tagged samples coming from $\mathbf{p}$

8: **return** the result with probability $9/10$, and a uniformly random choice of reject, or $\{0,1\} \setminus$ `result` otherwise

9: **else return** reject with probability $9/10$, and a uniformly random bit $b \in \{0,1\}$ otherwise

sources of samples from $\mathbf{p}$, those from $\sigma^{\mathbf{p}}$ (l. 1 of Algorithm 3) and those from X (l. 5), which in our private Algorithm 4 comes from disjoint samples, S_1 and S_2 (l.1 of Algorithm 4).

1. *Privacy for the samples sent to Merlin*: The very first message sent to Merlin, which includes samples from $\sigma^{\mathbf{p}} \subseteq S_1$ in the clear: we will make this part (ε, δ)-private *via* a combination of randomized response and privacy amplification by shuffling (Theorem 3.14);
2. *Privacy for the samples in* S_1: both the check in l. 7 and the check in l. 9 involve samples from $\sigma^{\mathbf{p}}$. Instead of trying to argue privacy (with respect to $\sigma^{\mathbf{p}}$) of these checks separately, we instead view the lines from l.3 onwards as computing a test f based on 3 inputs: (1) the response from Merlin, which is (ε, δ)-private by the first item; (2) the multiset of samples $\sigma^{\mathbf{p}}$; and (3) the multiset of samples $X \subseteq S_2$ (which does not overlap with S_1). By composition, to show the whole algorithm is $(2\varepsilon, \delta)$-DP with respect to S_1, we will rely on a known generic private to non-private reduction (Lemma 3.17 [ADR18]) for tests, using that the set of n samples from $\mathbf{p}$ used to form S_1 is a uniformly random $O(\varepsilon)$-fraction of our total set of samples from $\mathbf{p}$ (*i.e.*, of our overall $O(n + n')$ samples), which is possible as long as $n' \gtrsim n/\varepsilon$.
3. *Privacy for the samples in* S_2: the test in l. 9 is the only step using the multiset $X \subseteq S_2$ drawn in l. 5. To argue about privacy here, we need to be careful, as we estimate a quantity, $\sum_{x \in \tilde{B}_j^{\mathbf{p}}} \mathbf{p}(x)$, with a statistic which could have high sensitivity. (This is because $\tilde{B}_j^{\mathbf{p}}$ is a multiset, and so a single new element sampled in l. 5 could contribute up to $|\tilde{B}_j^{\mathbf{p}}|$ collisions in $\tilde{B}_j^{\mathbf{p}}$!). To avoid this, we will leverage a high-probability bound which follows from Assumption 2: namely, that with overwhelming probability no element appears more than $O(\log n)$ times in $\sigma_i^{\mathbf{p}}$, allowing us to bound the sensitivity: this is similar to Section 3.1.3, and is done in Lemma 3.16.

Proof of Theorem 3.3. The correctness and soundness of Algorithm 4 directly follow from Theorem 3.13, along with standard Binomial concentration arguments to bound the probabilities of (1) obtaining too few or too many samples from $\mathbf{p}$ in l. 2, and (2) keeping fewer than $T/2$ samples from $\mathbf{p}$ in l. 3 (using that this number is Binomially distributed by Theorem 3.14).

It remains to argue the Arthur-privacy guarantee, and the bounds on n and n'. We start with the former. By Theorem 3.14, the release of σ' is $(\varepsilon/2, \delta)$-DP with respect to S_1. By Lemma 3.16 applied to S_2 with $\Delta = O(\log n)$ (by the same argument as in Lemma 3.12), using Assumption 2), $b = \Theta(n\tau/L)$ and $\rho = \tau/k$, we get that the protocol is ε-DP with respect to S_2, for

$$n'n = O\left(\frac{kL \log n}{\varepsilon \tau^3}\right).$$

Now, we argue $\frac{\varepsilon}{2}$-DP with respect to S_1 of the steps from Algorithm 3 onwards. But this follows from Lemma 3.17, as long as $n' \geq C \cdot n/\varepsilon$ for a sufficiently large constant $C > 0$: indeed, the subsampling of S_1 from the first half of S, and the random flipping of the output in ll. 8 and 9 of Algorithm 4 simulate this generic transformation. Putting things together, by composition the algorithm is (ε, δ)-DP with respect to the first half of S, and ε-DP with respect to the second half of S: it is overall (ε, δ)-DP with respect to S. The two conditions on n', n above translate to the sufficient conditions

$$n' = \Omega\left(\frac{n}{\varepsilon}\right), \qquad n = \tilde{\Omega}\left(\frac{\sqrt{k}}{\tau^{3/2}}\right),$$

while the condition from Theorem 3.14 requires

$$mn = \tilde{\Omega}\left(\frac{ks \log(1/\delta)}{\varepsilon^2}\right),$$

using that $N = m + n = \Theta(m)$ as $m \geq n$. This concludes the proof: the total sample complexity is $O(n' + n) = O(n')$, and the communication complexity $O(N \log(k/\tau)) = O(m \log(k/\tau))$. □

3.2.1 Sending samples to Merlin privately (without sending too much)

Our core subroutine to make the first message sent by Arthur private is detailed below:

Theorem 3.14. *Let n, m, s be integers satisfying $k \geq N = n + m$ and $n \geq 2s \geq 4$. Let $\varepsilon, \delta \in (0, 1)$, and $\varepsilon_L = \log(1 + \frac{ks}{n-s})$. Taking n samples from $\mathbf{p}$, appending m uniformly random samples, applying a uniformly random permutation to the N total samples, then applying k-Randomized Response (Definition 2.2) with privacy parameter ε_L to each of them satisfies (ε, δ)-differential privacy, and sends s' unperturbed samples from $\mathbf{p}$, where $T' \sim \mathrm{Bin}(n, T/n)$, whenever*

$$n \geq \frac{1024ks \log(4/\delta)}{N\varepsilon^2}.$$

In particular, setting $s = \Theta\left(\sqrt{k}\right)$ and $N = \Theta(k^c)$ for some $c \in (3/4, 1)$, taking $n = O\left(\frac{k^{3/2-c} \log(1/\delta)}{\varepsilon^2}\right)$ suffices.

Proof. We will rely on the following result on privacy amplification by shuffling, noting that padding our dataset with arbitrary — or in this case, uniformly random — samples cannot decrease the effect of privacy amplification.

Theorem 3.15 (Amplification by Shuffling [FMT22])**.** *For any domain $\mathcal{X}$, let $Q : \mathcal{X} \to \mathcal{Y}$ be an algorithm such that Q is an ε_L local randomiser. Let $A_S : \mathcal{X}^n \to \mathcal{Y}^n$ be the algorithm that, given a dataset $x_{1:n} \in \mathcal{X}^n$, samples a uniform random permutation π over $[n]$, then sequentially computes $z_i = Q(x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Then for any $\delta \in [0, 1]$ such that $\varepsilon_L \leq \log\left(\frac{n}{16 \log(2/\delta)}\right)$, A_S is (ε_S, δ)-DP, where*

$$\varepsilon_S \leq \log\left(1 + 8\frac{e^{\varepsilon_L} - 1}{e^{\varepsilon_L} + 1}\left(\frac{\sqrt{e^{\varepsilon_L} \log(4/\delta)}}{\sqrt{n}} + \frac{e^{\varepsilon_L}}{n}\right)\right).$$

Now, set the parameter ε_L as

$$\varepsilon_L = \log\left(1 + \frac{ks}{n - s}\right),$$

which is well defined since $n > s$. Viewing k-Randomized Response as outputting x with probability $\frac{e^{\varepsilon_L} - 1}{e^{\varepsilon_L} + k - 1}$ and a uniformly random element of $[k]$ otherwise, we get

$$\Pr[k\text{-RR}(x) \text{ unflipped}] = \frac{e^{\varepsilon_L} - 1}{e^{\varepsilon_L} + k - 1} = \frac{s}{n}.$$

In particular, the mechanism sends $n \cdot \frac{s}{n} = s$ unperturbed samples from $\mathbf{p}$ in expectation.

We'd like to understand when

$$8\sqrt{\frac{e^{\varepsilon_L} \log(4/\delta)}{N}} \leq \frac{\varepsilon}{2},$$

which happens to be when

$$e^{\varepsilon_L} \leq \frac{N\varepsilon^2}{256 \log(4/\delta)}. \tag{15}$$

And this choice of ε_L also implies

$$8\frac{e^{\varepsilon_L}}{N} \leq \frac{\varepsilon^2}{32\log(4/\delta)} \leq \frac{\varepsilon}{2},$$

where the last inequality uses $\varepsilon < 1 < 16\log(4/\delta)$. Hence $\varepsilon_S \leq \log(1+\varepsilon) \leq \varepsilon$. Furthermore, since $\varepsilon < 1$ and $\log(2/\delta) \leq \log(4/\delta)$, this choice also implies the requirement that $\varepsilon_L \leq \log\left(\frac{N}{16\log(2/\delta)}\right)$ of Theorem 3.15 is satisfied.

Recall that $e^{\varepsilon_L} = 1 + \frac{ks}{n-s}$. Hence, provided $N\varepsilon^2 > 256\log(4/\delta)$, the condition above is equivalent to

$$1 + \frac{ks}{n-s} \leq \frac{N\varepsilon^2}{256\log(4/\delta)}$$
$$n \geq \frac{256ks\log(4/\delta)}{N\varepsilon^2 - 256\log(4/\delta)} + s.$$

Now notice that, by assumption, $n \geq \frac{1024ks\log(4/\delta)}{N\varepsilon^2}$. Since $k \geq n$ and $s \geq 2$, this gives

$$N\varepsilon^2 \geq \frac{1024ks\log(4/\delta)}{n} \geq 1024s\log(4/\delta) \geq 2048\log(4/\delta),$$

and in particular $N\varepsilon^2 - 256\log(4/\delta) \geq \frac{N\varepsilon^2}{2}$. And so,

$$n \geq \frac{512ks\log(4/\delta)}{N\varepsilon^2} + s$$

is sufficient. Since $a + b \leq \max\{2a, 2b\}$,

$$n \geq \max\left\{\frac{1024ks\log(4/\delta)}{N\varepsilon^2}, 2s\right\}$$

will also be sufficient. Notice that Eq. (15) implies that

$$nN \geq \frac{256ks\log(4/\delta)}{\varepsilon^2}.$$

Since $n \leq N$, we always have

$$N^2 \geq \frac{256ks\log(4/\delta)}{\varepsilon^2}$$

which, with $T = k^{c_1}$, implies that

$$N \geq \sqrt{\frac{256\log(4/\delta)}{\varepsilon^2}}k^{\frac{1+c_1}{2}} > k^{\frac{1+c_1}{2}}.$$

Therefore, we are constrained, for $s = \sqrt{k}$, to choosing $N = k^{c_2}$ for $c_2 \in (3/4, 1)$. □

3.2.2 The last two missing privacy building blocks

To conclude, we only miss two subroutines: the first to privately perform the test in l. 9 of Algorithm 3 (with respect to the multiset S_2 of samples), and the second to perform the computation of f privately overall (with respect to the multiset S_1 of samples).

Lemma 3.16. *There exists an algorithm which, given as input $\varepsilon \in (0,1]$, $L \geq 2$, $\tau \in (0,1]$, $b, \rho > 0$, along with a sequence of disjoint multisets $B_0, \dots, B_{L-1}$ such that $|B_j| \geq b$ for all j and no B_j contains more than Δ occurrences of any element, a sequence of values $p_0, \dots, p_{L-1} \geq \rho$, and a multiset S of size n', has the following guarantees. The algorithm returns either accept or reject, is ε-DP with respect to S, and, provided*

$$n' \geq C \cdot \Delta \frac{\log L}{\varepsilon \tau \rho b}$$

for some absolute constant $C > 0$, outputs estimates of $|S \cap B_0|, \dots |S \cap B_{L-1}|$ which are all simultaneously accurate (the estimate of $|S \cap B_j|$ being within an additive $\pm \tau p_j |B_j|$ of the true value) with probability at least $9/10$.

Proof. This is nearly identical to the proof of Lemma 3.12, using Laplace histograms to estimate simultaneously all quantities to ℓ_∞ error $\tau \cdot b \cdot \rho$. □

Finally, we recall the following "folklore" result, which explains how to convert any non-private testing algorithm into an ε-DP one, at the cost of a factor $O(1/\varepsilon)$ in its sample complexity.

Lemma 3.17 (Generic Differentially Private Tester [ADR18, Adapted from Theorem B.1.])**.** *Let $\mathcal{T}$ be a distribution tester that uses n samples from distribution $\mathbf{p}$ over $[k]$, outputs either $\perp$, accept, reject, and (1) outputs $\perp$ with probability at most $1/10$; conditioned on not outputting $\perp$, errs with probability at most $1/10$. Then Algorithm 5 is a ε-differentially private testing using $O(n/\varepsilon)$ samples, with the same guarantees as $\mathcal{T}$, with $1/10$ replaced by $1/5$.*

Algorithm 5 Generic Differentially Private Tester (adapted from [ADR18])

Input: Sample access to $\mathbf{p}$

1: $r \leftarrow \left\lceil \frac{20}{\varepsilon} \right\rceil$
2: $n' \leftarrow r \cdot n$
3: $X \leftarrow \mathbf{p}^{\otimes n'}$
4: $b \leftarrow$ Pick a uniformly random number from $[r]$
5: $O \leftarrow \mathcal{T}((X_i)_{i \in [(r-1)b+1, rb]})$
6: With probability $1/10$, flip O (to one of the other 2 outcomes uniformly at random)
7: Output O

3.3 Handling the 3 assumptions

We here explain how to address the 3 assumptions made, Assumptions 1 to 3, essentially without loss of generality: replacing $\mathbf{p}$ by a distribution $\mathbf{q}$ $O(\tau)$-close in total variation distance (and has an explicit expression as a function of $\mathbf{p}$), which is enough for most testing, learning, and estimation

purposes.[16] Recall that both Arthur and Merlin have sample access to the unknown distribution $\mathbf{p}$ and (in this setting) know the parameter τ.

Handling Assumption 1 (No probability is too small)**.** By having both parties (Arthur and honest Merlin) mix the unknown distribution $\mathbf{p}$ with the uniform distribution $\mathbf{u}_k$, to form the distribution

$$\mathbf{p}' := (1 - c_1\tau) \cdot \mathbf{p} + c_1\tau \cdot \mathbf{u}_k$$

(to which sample access can be easily simulated given sample access to $\mathbf{p}$), for a constant $c_1 > 0$ both Arthur and Merlin can choose, we get a distribution such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') \leq c_1\tau$, and which satisfies Assumption 1 whenever $c_1 \geq 1$.

Handling Assumption 3 (No bucket is negligible)**.** Similarly, by having both parties (Arthur and honest Merlin) mix the unknown distribution $\mathbf{p}$ with an explicit (to be defined soon) distribution $\mathbf{p}^*$ to form the distribution

$$\mathbf{p}' := (1 - c_2\tau) \cdot \mathbf{p} + c_2\tau \cdot \mathbf{p}^*$$

for an explicit constant $c_2 > 0$ both Arthur and Merlin can easily compute, we get a distribution such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') \leq O(\tau)$ and which satisfies Assumption 3.

The distribution $\mathbf{p}^*$ is defined as follows, where the idea is to put weight $1/L$ on each of the L' possible buckets[17]: fix L' (arbitrary) disjoint intervals of the domain, $I_0, \ldots, I_{L'}$, where

$$L' = O\left(\frac{1}{\tau} \log \frac{k}{n\tau}\right)$$

corresponds to the number of "possible buckets", and $|L_j| := \left\lceil \frac{1}{\bar{p}_j L} \right\rceil$. Note that

$$\sum_{j=0}^{L'} |L_j| \leq \sum_{j=0}^{L'} \left(\frac{k}{\tau(1+\tau)^j L} + 1 \right) = L' + O\left(\frac{k}{\tau^2 L}\right) = O\left(\frac{k}{\tau \log(k/\tau)}\right)$$

and so creating these disjoint intervals is indeed possible, potentially extending the size of the domain by a factor $O(1/(\tau \log k))$ (adding new elements) if $\tau \ll \log k$; this would only affect the complexities by a (small) $\mathrm{poly}(1/\tau)$ factor. Consider the (unnormalized) measure μ^* which, for all $0 \leq j \leq L'$, puts probability mass $\bar{p}_j$ on every element in L_j; and let $\mathbf{p}^*$ be its normalized version, $\mathbf{p}^* := \mu^* / \sum_{j=0}^{L'} \mu^*(L_j)$. Since

$$\frac{L'}{L} \leq \sum_{j=0}^{L'} \mu^*(L_j) = \sum_{j=0}^{L'} |L_j| \bar{p}_j \leq \sum_{j=0}^{L'} \left(\frac{1}{L} + \bar{p}_j \right) \leq \frac{L'}{L} + O\left(\frac{\tau}{k}(1+\tau)^{L'}\right) = \frac{L'}{L} + O\left(\frac{1}{n}\right)$$

and $\frac{L'}{L} = 1 - O\left(\frac{\log n}{\log(k/\tau)}\right)$, we get that $\sum_{j=0}^{L'} \mu^*(L_j) = \Theta(1)$, and so $\mathbf{p}^*$ is well-defined, and puts mass $\Theta(1/L)$ on each of the first L' buckets. Tweaking the constant $c_2 > 0$ (which only depends on L and the $\bar{p}_j$'s, known to both parties) concludes the argument, as $\mathbf{p}'$ satisfies Assumption 3 (and is on a domain of size $O(k/\tau)$).

[16] One could alternatively remove the two assumptions Assumptions 1 and 3 and handle the corner cases directly: this leads to additional technicalities to handle the "leftover bucket" $\mathbb{B}_\perp$ and the buckets B_j with low purported mass. We believe that the current argument below is simpler, and more elegant.

[17] So $\mathbf{p}^*$ looks like a "staircase distribution" with steps getting geometrically shorter as they get geometrically higher.

Handling Assumption 2 (No probability is too large)**.** There are several options for handling the last assumption: we here describe one particularly amenable to privacy, and only requires one extra round of interaction.

Merlin starts by sending to Arthur a set S of (purported) "large" elements, along with a 2-factor approximation of their probabilities. That is, using $O(n(\log n + \log k))$ bits of communication, Merlin sends a set S of pairs $(x, j(x))$ such that

- if $x \in [k]$ is such that $\mathbf{p}(x) > 1/n$, then $x \in S$;
- for every $x \in S$, $1/2^{j(x)} < \mathbf{p}(x) \leq 1/2^{j(x)-1}$;
- $|S| \leq 2n$.

This allows both of them to replace $\mathbf{p}$ by a new distribution $\mathbf{p}'$ on

$$k + \sum_{x \in S} \frac{n}{2^{j(x)-1}} \leq k + 2n$$

elements, where each element $x \in S$ is "divided" into $2n/2^{j(x)}$ elements, each of probability $\mathbf{p}(x)2^{j(x)-1}/n \leq 1/n$. Simulating sample access to $\mathbf{p}'$ given sample access to $\mathbf{p}$ is immediate, and total variation distance is preserved by this mapping (along with multiplicative approximation of individual probabilities).

If Merlin is honest, the above directly leads to satisfying Assumption 2. Moreover, if Merlin is not honest in this step, it is still possible to run the protocol: and it suffices to add an extra obvious check (if an element is tagged by Merlin as having probability greater than $1/n$, Arthur should reject, as this is evidence that Merlin is not only dishonest, but also sloppy). The correctness follows from the observation that, as the analysis of our protocol ignores the buckets that are not possible, if the various tests pass the guarantees apply to the possible buckets, and so the set of tagged samples is indeed approximate.[18]

Finally, note that the above is trivially Arthur-private, as the only communication comes from Merlin.

Conclusion. Combining the three reductions above (slightly tweaking the constants c_1, c_2 for the composition of the two mixtures to result in satisfying all assumptions simultaneously), one then obtains a distribution $\mathbf{p}'$ on a domain of size $O(k/\tau)$ which satisfies Assumptions 1 and 3, and, with high probability, Assumption 2 as well. Thus, it suffices to run the protocol on the resulting distribution $\mathbf{p}'$, which both parties either know or can get samples from.

References

[ABCR25] Maryam Aliakbarpour, Arnav Burudgunte, Clément L. Canonne, and Ronitt Rubinfeld. "Better Private Distribution Testing by Leveraging Unverified Auxiliary Data". In: *COLT*. Vol. 291. Proceedings of Machine Learning Research. PMLR, 2025, pp. 22–63 (cit. on pp. 5, 13).

[18] For privacy, we also may need, as in the proof of Theorem 3.3, to later use the high-probability bound which follows from this assumption: namely, that a value only appears $O(\log n)$ times, with overwhelming probability, in a multiset of n samples. This can be done easily after this preprocessing step, by having Arthur making this check in an ε-DP fashion, and rejecting if it is violated. This could be done, for example, by learning the histogram up to ℓ_∞ error, a task that can be done with $O(\log k)$ samples.

[ACFST21] Jayadev Acharya, Clément L. Canonne, Cody Freitag, Ziteng Sun, and Himanshu Tyagi. *Inference under Information Constraints III: Local Privacy Constraints*. arXiv:2101.07981 [cs, math, stat]. Jan. 2021. DOI: 10.48550/arXiv.2101.07981. URL: http://arxiv.org/abs/2101.07981 (visited on 05/30/2024) (cit. on p. 13).

[ACFT19] Jayadev Acharya, Clement Canonne, Cody Freitag, and Himanshu Tyagi. "Test without Trust: Optimal Locally Private Distribution Testing". en. In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 2019, pp. 2067–2076. URL: https://proceedings.mlr.press/v89/acharya19b.html (visited on 04/29/2022) (cit. on p. 13).

[ACT19] Jayadev Acharya, Clément L. Canonne, and Himanshu Tyagi. "Inference under information constraints: Lower bounds from chi-square contraction". In: *Proceedings of the thirty-second conference on learning theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of machine learning research. Phoenix, USA: PMLR, June 2019, pp. 3–17. URL: http://proceedings.mlr.press/v99/acharya19a.html (cit. on p. 13).

[ADK15] Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. "Optimal Testing for Properties of Distributions". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett. 2015, pp. 3591–3599 (cit. on pp. 40, 41).

[ADR17] Maryam Aliakbarpour, Ilias Diakonikolas, and Ronitt Rubinfeld. "Differentially Private Identity and Closeness Testing of Discrete Distributions". In: *CoRR* abs/1707.05497 (2017). arXiv: 1707.05497. URL: http://arxiv.org/abs/1707.05497 (cit. on p. 13).

[ADR18] Maryam Aliakbarpour, Ilias Diakonikolas, and Ronitt Rubinfeld. "Differentially Private Identity and Equivalence Testing of Discrete Distributions". en. In: *Proceedings of the 35th International Conference on Machine Learning*. PMLR, July 2018, pp. 169–178. URL: https://proceedings.mlr.press/v80/aliakbarpour18a.html (visited on 09/02/2025) (cit. on pp. 6, 11, 29, 32).

[AIRS24] Maryam Aliakbarpour, Piotr Indyk, Ronitt Rubinfeld, and Sandeep Silwal. "Optimal Algorithms for Augmented Testing of Discrete Distributions". In: *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. Ed. by Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang. 2024 (cit. on pp. 5, 13, 38, 39).

[ASZ18] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. "Differentially Private Testing of Identity and Closeness of Discrete Distributions". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett. 2018, pp. 6879–6891. URL: https://proceedings.neurips.cc/paper/2018/hash/7de32147a4f1055bed9e4faf3485a84d-Abstract.html (cit. on pp. 7, 13, 41).

[BBCS25] Ari Biswas, Mark Bun, Clément Canonne, and Satchit Sivakumar. *Interactive Proofs For Distribution Testing With Conditional Oracles*. arXiv:2511.22122 [cs]. Nov. 2025. DOI: 10.48550/arXiv.2511.22122. URL: http://arxiv.org/abs/2511.22122 (visited on 01/14/2026) (cit. on p. 13).

[BC18] Rishiraj Bhattacharyya and Sourav Chakraborty. "Property Testing of Joint Distributions using Conditional Samples". In: *ACM Trans. Comput. Theory* 10.4 (2018), 16:1–16:20 (cit. on p. 13).

[BV19] Victor Balcer and Salil Vadhan. "Differential Privacy on Finite Computers". en. In: *Journal of Privacy and Confidentiality* 9.2 (Sept. 2019). Number: 2. ISSN: 2575-8527. DOI: 10.29012/jpc.679. URL: https://journalprivacyconfidentiality.org/index.php/jpc/article/view/679 (visited on 10/03/2022) (cit. on p. 25).

[Can22] Clément L. Canonne. "Topics and Techniques in Distribution Testing: A Biased but Representative Sample". In: *Found. Trends Commun. Inf. Theory* 19.6 (2022), pp. 1032–1198. DOI: 10.1561/0100000114. URL: https://doi.org/10.1561/0100000114 (cit. on pp. 12, 21).

[CFGM16] Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. "On the Power of Conditional Samples in Distribution Testing". In: *SIAM J. Comput.* 45.4 (2016), pp. 1261–1296 (cit. on p. 13).

[CG18] Alessandro Chiesa and Tom Gur. "Proofs of Proximity for Distribution Testing". In: *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Ed. by Anna R. Karlin. Vol. 94. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 53:1–53:14. ISBN: 978-3-95977-060-6. DOI: 10.4230/LIPIcs.ITCS.2018.53. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2018.53 (cit. on pp. 3, 4, 6, 7, 12, 13, 15).

[CJKL22] Clément L. Canonne, Ayush Jain, Gautam Kamath, and Jerry Li. "The Price of Tolerance in Distribution Testing". In: *Conference on Learning Theory, 2-5 July 2022, London, UK*. Ed. by Po-Ling Loh and Maxim Raginsky. Vol. 178. Proceedings of Machine Learning Research. PMLR, 2022, pp. 573–624. URL: https://proceedings.mlr.press/v178/canonne22a.html (cit. on pp. 7, 42).

[CR14] Clément L. Canonne and Ronitt Rubinfeld. "Testing Probability Distributions Underlying Aggregated Data". In: *ICALP (1)*. Vol. 8572. Lecture Notes in Computer Science. Springer, 2014, pp. 283–295 (cit. on pp. 5, 11, 13).

[CRS15] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. "Testing Probability Distributions using Conditional Samples". In: *SIAM J. Comput.* 44.3 (2015), pp. 540–616 (cit. on p. 13).

[DDSVV13] Constantinos Daskalakis, Ilias Diakonikolas, Rocco A. Servedio, Gregory Valiant, and Paul Valiant. "Testing k-Modal Distributions: Optimal Algorithms via Reductions". In: *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*. Ed. by Sanjeev Khanna. SIAM, 2013, pp. 1833–1852 (cit. on pp. 40, 41).

[DDTZ26] Elbert Du, Cynthia Dwork, Pranay Tankala, and Linjun Zhang. “Differentially Private Verification of Distribution Properties”. In: *CoRR* abs/2604.10819 (2026). DOI: 10.48550/ARXIV.2604.10819. arXiv: 2604.10819. URL: https://doi.org/10.48550/arXiv.2604.10819 (cit. on pp. 5, 6).

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *TCC*. Vol. 3876. Lecture Notes in Computer Science. Springer, 2006, pp. 265–284 (cit. on pp. 4, 14).

[FMT22] Vitaly Feldman, Audra McMillan, and Kunal Talwar. “Hiding Among the Clones: A Simple and Nearly Optimal Analysis of Privacy Amplification by Shuffling”. In: *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. Denver, CO, USA: IEEE, Feb. 2022, pp. 954–964. ISBN: 978-1-6654-2055-6. DOI: 10.1109/FOCS52979.2021.00096. URL: https://ieeexplore.ieee.org/document/9719772/ (visited on 03/19/2022) (cit. on p. 30).

[Gol17] Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017. ISBN: 978-1-107-19405-2. URL: http://www.wisdom.weizmann.ac.il/~oded/pt-intro.html (cit. on p. 12).

[Gol20] Oded Goldreich. “The Uniform Distribution Is Complete with Respect to Testing Identity to a Fixed Distribution”. In: *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*. Ed. by Oded Goldreich. Vol. 12050. Lecture Notes in Computer Science. Springer, 2020, pp. 152–172. DOI: 10.1007/978-3-030-43662-9_10. URL: https://doi.org/10.1007/978-3-030-43662-9%5C_10 (cit. on p. 12).

[Her24] Tal Herman. “Public Coin Interactive Proofs for Label-Invariant Distribution Properties”. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, London School of Economics, London, UK, August 28-30, 2024*. Ed. by Amit Kumar and Noga Ron-Zewi. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, 72:1–72:23. DOI: 10.4230/LIPICS.APPROX/RANDOM.2024.72. URL: https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2024.72 (cit. on pp. 5, 8, 11, 12, 17–19).

[HR22] Tal Herman and Guy N. Rothblum. “Verifying the unseen: interactive proofs for label-invariant distribution properties”. In: *STOC ’22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*. Ed. by Stefano Leonardi and Anupam Gupta. ACM, 2022, pp. 1208–1219. DOI: 10.1145/3519935.3519987. URL: https://doi.org/10.1145/3519935.3519987 (cit. on pp. 6, 13, 17).

[HR23] Tal Herman and Guy N. Rothblum. “Doubley-Efficient Interactive Proofs for Distribution Properties”. In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. IEEE, 2023, pp. 743–751. DOI: 10.1109/FOCS57990.2023.00049. URL: https://doi.org/10.1109/FOCS57990.2023.00049 (cit. on pp. 4, 5, 8, 10, 17, 26, 27).

[KBR16] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. “Discrete distribution estimation under local privacy”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML’16. New York, NY, USA: JMLR.org, June 2016, pp. 2436–2444. (Visited on 08/05/2026) (cit. on p. 14).

[KOV16] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. “Extremal mechanisms for local differential privacy”. In: *The Journal of Machine Learning Research* 17.1 (Jan. 2016), pp. 492–542. ISSN: 1532-4435 (cit. on p. 14).

[OS18] Krzysztof Onak and Xiaorui Sun. “Probability-Revealing Samples”. In: *AISTATS*. Vol. 84. Proceedings of Machine Learning Research. PMLR, 2018, pp. 2018–2026 (cit. on pp. 5, 11, 13).

[Pan08] Liam Paninski. “A Coincidence-Based Test for Uniformity Given Very Sparsely Sampled Discrete Data”. In: *IEEE Trans. Inf. Theory* 54.10 (2008), pp. 4750–4755. DOI: 10.1109/TIT.2008.928987. URL: https://doi.org/10.1109/TIT.2008.928987 (cit. on p. 12).

[PW25] Y. Polyanskiy and Y. Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2025. ISBN: 9781108832908. URL: https://books.google.com.au/books?id=CySo0AEACAAJ (cit. on p. 17).

[RSSS25] Sofya Raskhodnikova, Satchit Sivakumar, Adam Smith, and Marika Swanberg. “Differentially Private Sampling from Distributions”. en. In: *SIAM Journal on Computing* 54.2 (Apr. 2025), pp. 419–468. ISSN: 0097-5397, 1095-7111. DOI: 10.1137/22M1538703. URL: https://epubs.siam.org/doi/10.1137/22M1538703 (visited on 07/22/2026) (cit. on p. 13).

[Val11] Paul Valiant. “Testing Symmetric Properties of Distributions”. In: *SIAM Journal on Computing* 40.6 (2011), pp. 1927–1968. DOI: 10.1137/080734066 (cit. on p. 12).

[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 43rd Annual ACM Symposium on Theory of Computing (STOC)*. 2011, pp. 685–694. DOI: 10.1145/1993636.1993727 (cit. on p. 12).

A Connection to Augmented Testing

In the *augmented testing* model [AIRS24], we have a regular testing problem but augmented with advice from some untrusted third party. We are expected to perform the testing problem well if the advice is good, but are allowed to abort if the advice is obviously bad (in this case one can imagine backing off to try another approach). Specifically, we have the following definition:

Definition A.1 (Augmented tester [AIRS24])**.** Given parameters $\tau \in [0,1]$, $\alpha \in (0,1)$, and $\delta \in (0,1]$, as well as (sample) access to two distributions $\mathbf{p}, \mathbf{q} \in \Delta([k])$, along with the description of a *predicted distribution* $\hat{\mathbf{p}} \in \Delta([k])$, an algorithm $\mathcal{A}$ is an (τ, α, δ)-*augmented tester* for closeness testing if the following holds for every $\mathbf{p}$, $\mathbf{q}$, and $\hat{\mathbf{p}}$:

- If $\mathrm{d}_{\mathrm{TV}}(\hat{\mathbf{p}}, \mathbf{p}) \leq \tau$, the algorithm outputs $\perp$ with probability at most $\delta/2$.
- If $\mathbf{p} = \mathbf{q}$, then the algorithm outputs reject with a probability at most $\delta/2$.
- If $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) > \alpha$, then the algorithm outputs accept with probability at most $\delta/2$.

In this definition, we can replace $\mathbf{q}$ with a description of $\mathbf{q}$ to acquire an identity testing algorithm, and if $\mathbf{q}$ is the uniform distribution then we can say that $\mathcal{A}$ is a uniformity tester.

Roughly speaking, the above requires that the algorithm cannot be wrong (often), either about the distribution or the advice. The above definition, due to [AIRS24], is tied to a specific notion of advice $\hat{\mathbf{p}}$. We generalize it to allow for other forms of advice as follows.

Definition A.2 (Augmented tester, alternative definition)**.** Let $\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}} \subseteq \Delta([k])$ be disjoint. Given parameters $\delta \in (0,1]$, as well as (sample) access to a distribution $\mathbf{p} \in \Delta([k])$ and an untrusted advice string π, an algorithm $\mathcal{A}$ is an *augmented tester* for $\mathcal{P} = (\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ with (distribution-dependent) advice sets $(\Pi_{\mathbf{p}})_{\mathbf{p}\in\Delta([k])}$ if the following holds for every $\mathbf{p}$ and π:

- If the advice is "good", *i.e.*, $\pi \in \Pi_{\mathbf{p}}$, then the algorithm outputs $\perp$ with probability at most $\delta/2$.
- If $\mathbf{p} \in \mathcal{P}_{\text{yes}}$, then the algorithm outputs reject with probability at most $\delta/2$.
- If $\mathbf{p} \in \mathcal{P}_{\text{no}}$, then the algorithm outputs accept with probability at most $\delta/2$.

The advice complexity of the tester is the length $|\pi|$ of the advice string (in the worst case, over all $\mathbf{p}$).

Note that this clearly generalises Definition A.1 when $\Pi_{\mathbf{p}} = \{(\tau, \hat{\mathbf{p}}) : \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \hat{\mathbf{p}}) \leq \tau\}$. Now, recall that a distribution testing promise problem $(\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ is in MA-D if there exists a verifier $\mathcal{A}$, who receives explicit access to a proof string π written by an untrusted unbounded prover who knows $\mathbf{p}$, and has the following properties:

- **Completeness:** For all $\mathbf{p} \in \mathcal{P}_{\text{yes}}$, $\exists \pi \in \{0,1\}^*$ s.t. $\Pr_{r,\mathbf{p}}[A^{\mathbf{p}}(\pi, r) = \text{accept}] \geq 2/3$
- **Soundness:** For all $\mathbf{p} \in \mathcal{P}_{\text{no}}$, $\forall \pi \in \{0,1\}^*$, $\Pr_{r,\mathbf{p}}[A(\pi, r) = \text{accept}] \leq 1/3$.

With this in mind, we show that if an MA-D promise problem $(\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ is also in coMA-D (that is, if $(\mathcal{P}_{\text{no}}, \mathcal{P}_{\text{yes}})$ is in MA-D), then the combination of the two corresponding proofs/verifiers yields an augmented tester for that problem.

Theorem A.3. *If $\mathcal{P} = (\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}}) \in$ MA-D $\cap$ coMA-D, then there exists an augmented tester for $\mathcal{P}$ with advice complexity $|\pi_{\text{yes}}| + |\pi_{\text{no}}|$, and sample complexity $n_{\text{yes}} + n_{\text{no}}$.*

Proof. Let $P_{\text{yes}} := (\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ be a distribution testing promise problem in MA-D $\cap$ coMA-D and let $P_{\text{no}} := (\mathcal{P}_{\text{no}}, \mathcal{P}_{\text{yes}})$ be its complement. This means that we have two verifiers $\mathcal{A}_{\text{yes}}$ and $\mathcal{A}_{\text{no}}$, augmented with access to the same distribution $\mathbf{p}$, which accept two different proofs π_{yes} and π_{no} from two different provers $\mathcal{M}_{\text{yes}}$ and $\mathcal{M}_{\text{no}}$.

Define the augmented testing algorithm as follows: Let the advice sets be the pair of proofs given by each (competing) prover in the promise problem $\Pi_{\mathbf{p}} = (\pi_{\text{yes}}, \pi_{\text{no}})$. Then, running the corresponding verifier on each one we get the following tables of possibilities.

truth	$\mathcal{A}_{\text{yes}}$	$\mathcal{A}_{\text{no}}$	occurs with probability	Outcome
yes	yes	yes	high probability	yes
yes	no	no	$\leq \delta$	no (in error)
yes	yes	no	$\leq \delta$	$\perp$
yes	no	yes	$\leq \delta$	$\perp$
no	yes	yes	$\leq \delta$	yes (in error)
no	no	no	high probability	no
no	yes	no	$\leq \delta$	$\perp$
no	no	yes	$\leq \delta$	$\perp$

Note that to avoid confusion, we refer to the set indicated by the output of $\mathcal{A}$, rather than the acceptance or rejection. So $\mathcal{A}_{\text{no}}$ outputs accept when it would usually output reject, since that is the case indicated by its answer.

As we can see, all of the quantities required by the augmented testing algorithm are achieved by answering according to this table. This is ensured by the fact that we only output "bad advice" when the verification algorithms disagree, meaning one of them made an error, and only output accept or reject mistakenly when *both* of the verifiers error. □

Conversely, an augmented tester implies MA-D and coMA-D protocols for the property:

Theorem A.4. *If $\mathcal{P} = (\mathcal{P}_{\text{yes}}, \mathcal{P}_{\text{no}})$ has an augmented tester with sample complexity n and advice complexity c, then $\mathcal{P} \in$ MA-D $\cap$ coMA-D with sample complexity n and proof complexity c.*

Proof. Consider the MA-D protocol which uses as a proof some advice from the advice set of the augmented testing problem. If we run the same verifier as from that problem, and output reject whenever we see $\perp$, we get the theorem as stated. □

B The MA learning baselines

Theorem B.1 (Oblivious-Merlin baseline)**.** *There exists an MA protocol with communication complexity $O(k \log k)$ and sample complexity $O(\sqrt{k}/\alpha^2 + \log k/\alpha^4)$, allowing Arthur to* learn *the unknown distribution $\mathbf{p}$ to total variation distance α. Moreover, Merlin does not need to know α.*

Proof. The protocol is presented as Algorithm 6. In step 1, Merlin sends a permutation over k elements, which requires $\log k! = O(k \log k)$ bits of communication. The monotonicity tester from [ADK15] (with accuracy parameter $\alpha/3$) has sample complexity (see Theorem 2 in [ADK15])

$$O\left(\frac{\sqrt{k}}{\alpha^2} + \frac{\log k}{\alpha^4}\right)$$

and has the property that it outputs reject (with high probability) if $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathcal{M}) > \alpha/3$, where $\mathcal{M}$ is the class of monotone distributions. In particular, if the monotone tester outputs accept, it must be that $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathcal{M}) \leq \alpha/3$. In this case, the agnostic learning algorithm from Section 3 of [DDSVV13] gives us a $\mathbf{q}$ satisfying $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{q}) \leq 3(\alpha/3) = \alpha$. This learning algorithm has sample complexity

$$O\left(\frac{\log(\alpha k)}{\alpha^3}\right).$$

We now argue completeness and soundness of Algorithm 6.

Completeness. If Merlin is honest, $\tilde{\mathbf{p}}$ is indeed a monotone distribution and Algorithm 6 outputs accept and a distribution $\hat{\mathbf{p}}$. Since $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{q}) \leq \alpha$ and the total variation distance is invariant to permutation applied to both arguments, we have $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \hat{\mathbf{p}}) \leq \alpha$.

Soundness. If Merlin is dishonest, and $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathcal{M}) > \alpha/3$, in which case the monotonicity tester outputs reject with high probability. Otherwise if Merlin sends $\tilde{\pi}$ such that $0 < \mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathcal{M}) \leq \alpha/3$ and the tester outputs accept, we will be in possession of $\mathbf{q}$ satisfying $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathcal{M}) \leq \alpha/3$ and the corresponding $\hat{\mathbf{p}}$ satisfies $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \hat{\mathbf{p}}) \leq \alpha$. □

Algorithm 6 Oblivious-Merlin MA protocol for learning $\mathbf{p}$

1: Merlin: Send a permutation $\tilde{\pi}$ such that the induced distribution $\tilde{\mathbf{p}} := \mathbf{p} \circ \tilde{\pi}$ is monotone. ▷ This requires $O(k \log k)$ bits of communication
2: Arthur: Run monotonicity tester from [ADK15] with accuracy parameter α (samples from $\tilde{\mathbf{p}}$ can be simulated using samples from $\mathbf{p}$, and permutation $\tilde{\pi}$).
If the tester outputs accept:

- output accept
- use agnostic monotone distribution learning algorithm from [DDSVV13] with parameter α to obtain $\mathbf{q}$, and output $\hat{\mathbf{p}} = \mathbf{q} \circ \tilde{\pi}^{-1}$.

Else, output reject

Theorem B.2 (Arthur-private baseline)**.** *There exists an ε-Arthur private MA protocol with communication complexity $O(k \log(k/(\varepsilon\alpha)))$ and sample complexity*

$$n = O\left(\frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha\sqrt{\varepsilon}} + \frac{k^{1/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{1}{\alpha\varepsilon}\right)$$

allowing Arthur to learn the distribution $\mathbf{p}$ to total variation distance α.

Proof. The basic idea here is the same as in Theorem 1.5: Merlin sends a purported quantized description $\tilde{\mathbf{p}}$ of the actual distribution $\mathbf{p}$. Arthur then uses DP identity tester from Theorem 2 of [ASZ18] to reject if $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}) > \alpha$, leading to the claimed sample complexity. Note that Merlin needs to know Arthur's privacy parameter ε because the quantization level depends on the tolerance property of the optimal DP tester; in particular, if the optimal DP tester uses n samples, the identity tester accepts with high probability with tolerance parameter $O(1/n)$. This, in turn, implies that it suffices for Merlin to quantize each probability to accuracy $\pm O(1/(kn))$, leading to communication complexity $O(k \log(kn)) = O(k \log(k/(\varepsilon\alpha)))$. □

Theorem B.3 (Merlin-private baseline)**.** *For every $\sqrt{k}/\alpha^2 \le t \le \varepsilon k$, there exists an ε-Merlin private MA distribution estimator with communication complexity $O(k \log(k/\alpha))$, sample complexity $O\left(\sqrt{k}/\alpha^2 + t\right)$ for Arthur, and $O(k^3/(\alpha^4 t^2) + k^2/(\alpha^2 \varepsilon t))$ for an honest Merlin, allowing Arthur to learn the distribution $\mathbf{p}$ to total variation distance α.*

Proof. The idea here again is as in Theorem 1.5 in that Merlin sends a purported description $\tilde{\mathbf{p}}$ of $\mathbf{p}$. Here however, instead of assuming that Merlin knows $\mathbf{p}$, we assume that Merlin learns $\mathbf{p}$ using its own supply of samples (which is why the question of Merlin-privacy arises). It is a folklore result that to ε-privately learn a distribution with accuracy parameter τ, one needs $\Theta(k/\tau^2 + k/\tau\varepsilon)$ samples. Thus Merlin can ε-privately learn $\mathbf{p}$ with accuracy parameter $\tau = \alpha/\sqrt{k}$ (as in Theorem 1.5) using $O\left(k^2/\alpha^2 + k^{3/2}/\alpha\varepsilon\right)$ samples and the rest of the protocol works exactly as in Theorem 1.5 (where Arthur would need now $\sqrt{k}/\alpha^2$ samples to run the standard identity testing protocol).

To give Merlin some respite in terms of sample complexity, here we assume that Arthur has a (known to Merlin) budget of $n = O\left(\sqrt{k}/\alpha^2 + t\right)$ samples. The extra $O(t)$ samples with Arthur allows Merlin to learn $\mathbf{p}$ with a bigger accuracy parameter τ, thus requiring fewer samples on its end. Arthur, in turn, would then need to run identity testing with tolerance parameters (τ, α) (τ was $\alpha/\sqrt{k}$ in

Theorem 1.5). For this, we use Theorem 1 from [CJKL22] that says that one can do tolerant identity testing with tolerance parameters (τ, α) using

$$\tilde{\Theta}\left(\frac{\sqrt{k}}{\alpha^2} + k \max\left\{\frac{\tau}{\alpha^2}, \frac{\tau^2}{\alpha^4}\right\}\right)$$

samples. Thus, Merlin equates

$$t = \tilde{\Theta}\left(k \max\left\{\frac{\tau}{\alpha^2}, \frac{\tau^2}{\alpha^4}\right\}\right)$$

to obtain τ and then privately learns $\mathbf{p}$ to accuracy τ using $O(k/\tau^2 + k/\tau\varepsilon)$ samples. This results in the claimed sample complexity for Merlin.

□