Testing Histograms Nearly Optimally

Anonymous Author(s)

ABSTRACT

We investigate the problem of testing whether a discrete probability distribution over an ordered domain is a histogram on a specified number of bins. Histograms are one of the most common tools for the succinct approximation of data. Formally, a *k-histogram* over $[n]$ is a probability distribution that is piecewise constant over a set of k intervals. Given samples from an unknown distribution $\mathbf{p}$ on $[n]$, we want to distinguish between the cases that $\mathbf{p}$ is a k-histogram versus far from any k-histogram, in ℓ_1-distance. Our main result is a sample near-optimal computationally efficient algorithm for this testing problem and a nearly-matching (within logarithmic factors) sample complexity lower bound.

CCS CONCEPTS

• **Mathematics of computing → Hypothesis testing and confidence interval computation**; **Probabilistic algorithms**; • **Theory of computation → Streaming, sublinear and near linear time algorithms**; **Lower bounds and information complexity**;

KEYWORDS

distribution testing, histograms, binning, probability distributions, lower bounds

ACM Reference Format:

Anonymous Author(s). 2018. Testing Histograms Nearly Optimally. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

1.1 Background and Motivation

A classical approach for the efficient exploration of massive datasets involves the construction of succinct data representations, see, e.g., the survey [13]. One of the most useful and commonly used compact representations are *histograms*. For a dataset S, whose elements are from the universe $[n] := \{1, \ldots, n\}$, a k-histogram is a function that is piecewise constant over k interval pieces. Histograms constitute the oldest and most popular synopsis structure in databases and have been extensively studied in the database community since their introduction in the 1980s [28], see, e.g., [2, 8, 12, 21–23, 25, 26, 33], for a partial list of references. In both the statistics and computer science literatures, several methods have been proposed to estimate histogram distributions in a range of natural settings [2, 3, 11, 14, 19, 20, 27, 29, 32].

In this work, we study the algorithmic task of deciding whether a (potentially very large) dataset S over the domain $[n]$ is a k-histogram (i.e., it has a succinct histogram representation with k interval pieces) or is "far" from *any* k-histogram representation (in a well-defined technical sense). Our focus is on *sublinear time* algorithms [31]. Instead of reading the entire dataset S, which could be highly impractical, one can instead use randomness to sample a small subset of the dataset. Note that sampling a (uniformly) random element from S is equivalent to drawing a sample from the underlying probability distribution $\mathbf{p}$ of relative empirical frequencies. This observation brings our algorithmic problem of "histogram testing" in the framework of distribution property testing (statistical hypothesis testing) [4, 5], see, e.g., [9] for a survey.

Formally, we study the following task: Given access to i.i.d. samples from an unknown distribution $\mathbf{p}$ on $[n]$ and a desired error tolerance $0 < \varepsilon < 1$, we want to correctly distinguish (with high probability) between the cases that $\mathbf{p}$ is a k-histogram versus ϵ-far from any k-histogram, in ℓ_1-norm (or, equivalently, total variation distance). It should be noted that the histogram testing problem studied here is not new. Prior work within the algorithms and database theory community has investigated the complexity of the problem in the past ten years (see, e.g., [2, 8, 25] and Section 1.3 for a detailed summary of prior work). However, known algorithms for this task are suboptimal, and in particular there is a polynomial gap between the best known upper and lower bounds on the sample complexity of the problem. At a high level, the difficulty of our histogram testing problem in the sublinear regime lies in the fact that the location and "length" of the k intervals defining the histogram representation (if one exists) is a priori unknown to the algorithm.

We believe that the histogram testing problem is natural and interesting in its own right. Moreover, a sample-efficient algorithm for this testing task can be used as a key primitive in the context of *model selection*, where the goal is to identify the "most succinct" data representation. Indeed, various algorithms are known for learning k-histograms from samples whose sample complexities (and running times) scale proportionally to the succinctness parameter k (and are completely independent of the domain size n) [2, 3, 11]. Combined with an efficient tester for the property of being a k-histogram (used to identify the smallest possible value of k such that $\mathbf{p}$ is a k-histogram, e.g., via binary search), one can obtain a sketch of the underlying dataset. See Section 4 for a detailed description.

1.2 Our Contributions

Our main contribution is a near-characterization of the complexity of the histogram testing problem. Specifically, we provide (1) a sample near-optimal (within logarithmic factors) and computationally efficient algorithm for the problem, and (2) a nearly matching sample complexity lower bound.

In particular, we establish the following theorems.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

Theorem 1.1 (Upper Bound). *For any $k \in [n]$, there exists a testing algorithm for the class of k-histograms on $[n]$ with sample complexity $m = \widetilde{O}((\sqrt{n}+k)/\varepsilon^2)$ and running time* $\mathrm{poly}(m)$.

Theorem 1.2 (Sample Complexity Lower Bound). *For any $k \in [n]$, any testing algorithm for the class of k-histograms on $[n]$ requires at least $\widetilde{\Omega}((\sqrt{n}+k)/\varepsilon^2)$ samples.*

In the theorems above, the $\tilde{O}$ and $\tilde{\Omega}$ hide (small) poly-logarithmic factors in $\log(k/\varepsilon)$. Taken together, Theorems 1.1 and 1.2 nearly characterize the complexity of the histogram testing problem. We note that the best previously known upper and lower bounds were sub-optimal by polynomial factors in $1/\epsilon$ [8]. Concretely, our results significantly improve prior work for either large k (specifically, $k = \Omega(\sqrt{n})$) or small tolerance parameter ε. Interestingly, the latter is a key regime of interest, as it corresponds to the "high-accuracy" testing setting.

At the technical level, our sample complexity lower bound construction conceptually differs from previous work in distribution testing, drawing instead from sophisticated techniques from the distribution *estimation* literature. In the following subsection, we elaborate on prior works, and how our results relate to them.

1.3 Prior Work

Motivated by the question of building provably good succinct representations of a dataset from only a small subsample of the data, [25] first introduced histogram testing as a preliminary, ultraefficient decision subroutine to find the best parameter k for the number of bins. They provided an algorithm for this task which required $\widetilde{O}(\sqrt{kn}/\varepsilon^5)$ samples from the dataset, a sample complexity which beats the naïve approach (reading and processing the whole dataset) for small values of k and relatively large values of the accuracy parameter ε. Subsequent work [10] reduced the dependence on ε from quintic to cubic, giving an algorithm with sample complexity $\widetilde{O}(\sqrt{kn}/\varepsilon^3)$. This bound was, however, still quite far from the "trivial" lower bound of $\Omega(\sqrt{n}/\varepsilon^2)$, which follows from a reduction to uniformity testing (i.e., the case $k = 1$) [30].

Prior to the current work, the state-of-the-art bounds were obtained in [8]. The latter work decoupled the dependence between k and n, providing a (somewhat complicated) testing algorithm achieving sample complexity $\widetilde{O}(\sqrt{n}/\varepsilon^2 + k/\varepsilon^3)$, and complementing it with an information-theoretic sample lower bound of $\widetilde{\Omega}(\sqrt{n}/\varepsilon^2 + k/\varepsilon)$.[1] While this decoupling allowed for a much larger range of parameters, and improved on the algorithms of [2, 10] for all $1 \ll k \ll n$, the dependence on the accuracy parameter ε still remained problematic, and led to suboptimal data- and time-efficiency for many important parameter regimes, e.g., any setting of $\sqrt{n} \ll k \ll n$ and $\varepsilon \ll 1$. Moreover, the lower bound of [8], based on a reduction of histogram testing to the well-studied problem of support size estimation, provably cannot be improved to provide a quadratic dependence on ε, i.e., $\widetilde{\Omega}(k/\varepsilon^2)$, and thus inherently leads to an $\Omega(1/\varepsilon^2)$-factor gap between the upper and lower bounds. Our work remedy all those issues, fully resolving the question of histogram testing, for the whole parameter range, with logarithmic factors.

Finally, we note that a number of works have obtained algorithms and lower bounds for related, yet significantly different, testing problems. Specifically, [15] gave a sample-optimal testing algorithm for the special case of our problem where the k intervals are known *a priori*. Moreover, a number of works [16–18] have obtained identity and equivalence testers *under the assumption* that the input distributions are k-histograms.

1.4 Overview of Techniques

We now proceed to highlight the key aspects of our techniques, and how they depart from previous work, starting with our algorithmic contribution (Theorem 1.1).

Upper Bound. Our algorithmic result leverages the "testing by learning" paradigm proposed by [1], a variant of which was applied to histogram testing by [8]. At a high level, one first learns (in a very stringent sense) a good approximation of an unknown k-histogram that is accurate on "most" parts of the domain. (The choice of the "metric" under which learning is performed in this step is crucial for the algorithm. Importantly, the χ^2-distance is used, as opposed to the total variation distance.) One then verifies that the learned description **(a)** satisfies the histogram property and **(b)** is close to the underlying distribution. However, since the approximation could be *arbitrarily bad* on a small number of parts of the domain, in step **(b)** one needs to apply the testing procedure *iteratively*, in order to remove those "potentially bad" parts which could otherwise lead the decision algorithm astray.

The subtlety is that one needs to perform the removal carefully: if the total probability mass of the removed domain is more than ε, the tester may erroneously accept distributions far from histograms. The approach taken by [8] is to sub-divide the domain into many "small" intervals of (almost) equal probability mass. Specifically, they divided the domain into $\Omega((k\log k)/\varepsilon)$ many intervals ("quantiles"), and then argued that their "sieving" procedure removed at most $O(k\log k)$ intervals. As a result, the total mass of the removed intervals was at most ε, as desired. The issue with this approach is that the learning procedure is not sample efficient. Specifically, due to the large number of quantiles, they require many samples to achieve sufficient accuracy: in fact, one can check that at least $\Omega((k\log k)/\varepsilon^3)$ samples are needed, resulting in a (sub-optimal) cubic dependency on $1/\varepsilon$.

To circumvent this issue, unlike [8], our algorithm performs the dividing step *iteratively* as well (as opposed to only the sieving step). We begin by dividing the domain into only $\Theta(k\log k)$ intervals. After the "potentially bad" intervals are selected among those, we focus on those, subdividing them into further smaller parts, and iterating on them. We show that after roughly $\log(1/\varepsilon)$ iterations, we are then able to locate all $O(k\log k)$ "bad" intervals, with total mass bounded by ε, by dividing the domain into at most $O(\log(1/\varepsilon)\cdot k\log k)$ intervals, thus saving a factor $\widetilde{\Theta}(1/\varepsilon)$. The algorithm and its analysis can be found in Section 2.

Sample Complexity Lower Bound. We now turn to the ideas behind our information-theoretic lower bound (Theorem 1.2) which

[1] In more detail, [8] established the bounds $O((\sqrt{n}\log k)/\varepsilon^2 + (k\log^2 k)/\varepsilon^3)$ and $\Omega(\sqrt{n}/\varepsilon^2 + k/(\varepsilon\log k))$; our lower bound strictly improves on theirs, while the key improvement in our upper bound lies in the second term, which as discussed above dominates for large k and/or small ε.

establishes near-optimality of our algorithm. As mentioned above, the $\Omega(\sqrt{n}/\varepsilon^2)$ term of the lower bound was previously shown in [8], and follows from an easy reduction to the case $k = 1$. In the same paper, an $\Omega(k/(\varepsilon \log k))$ lower bound was shown, via a reduction from *Support Size Estimation*, for constant $\varepsilon = \Omega(1)$. The linear dependency on $1/\varepsilon$ for general ε was then obtained by a simple embedding technique, where one adds one extra heavy element with mass $1-\varepsilon$ to the hard instance. Unfortunately, this "embedding trick" inherently cannot yield any better dependence than $1/\varepsilon$, and in particular falls short of the quadratic $1/\varepsilon^2$ we establish.

At a high level, our $\Omega(k/(\varepsilon^2 \log k))$ lower bound is also based on the hard instance of *Support Size Estimation*. However, instead of using this hard instance as a "black box" in our reduction, we carefully modify it by mixing the hard instance with the uniform distribution, thus allowing the discrepancy between "good" and "bad" instances to be "evenly spread" across the entire domain Ω. Intuitively, this amounts to adding some noise *everywhere*, instead of just at a localized point, making the task of any algorithm trying to distinguish between good and bad instances much more challenging. However, this comes with its own technical challenges, as we are no longer able to rely on the existing (and rather involved) sample complexity lower bounds for Support Size Estimation, but instead must prove those lower bounds directly for our modified construction.

Our starting point to achieve this is the hard instance for support size estimation constructed in [34], which first constructs univariate random variables U, U' with matching moments (that is, $\mathbf{E}\left[|U|^j\right] = \mathbf{E}\left[|U'|^j\right]$ for all sufficiently small j) and then use those variables to stochastically generate a pair of adversarial distributions $p_U, p_{U'}$ over $\Omega = \{1, 2, \ldots, n\}$, one element at a time. In our proof, we generate our distributions from another pair of derived random variables we define, $B = \frac{1}{k}(1 + \varepsilon \cdot U)$ and $B' = \frac{1}{k}(1 + \varepsilon \cdot U')$ (where we first reduced the problem to the extremal case $k \approx n$). Due to the moment-matching property of U, U' and our mixing with the uniform distribution, the samples coming from the pair of distributions will have similar statistical profiles (even though the two source distributions B, B' are themselves statistically far) as long as the number m of samples is $m \ll k/(\varepsilon^2 \log k)$, making them impossible to distinguish. The details of our lower bound are provided in Section 3.

1.5 Preliminaries

We will denote by $\mathrm{TV}(\mathbf{p}, \mathbf{q})$ the total variation (TV) distance between two probability distributions $\mathbf{p}, \mathbf{q}$ over $[n] := \{1, 2, \ldots, n\}$, defined as $\mathrm{TV}(\mathbf{p}, \mathbf{q}) := \sup_{S \subseteq [n]} (\mathbf{p}(S) - \mathbf{q}(S)) = (1/2) \sum_{i=1}^{n} |\mathbf{p}(i) - \mathbf{q}(i)|$ where $\mathbf{p}(S) := \sum_{i \in S} \mathbf{p}(i)$. We will make essential use of the chi-square distance between $\mathbf{p}, \mathbf{q}$, defined as $\chi^2(\mathbf{p} \,\|\, \mathbf{q}) := \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{q}_i)^2/\mathbf{q}_i$.

We will also consider generalizations of these distances on restrictions of the domain. In particular, given $\mathcal{I}$ consisting of a set of disjoint sub-intervals in $[n]$, we use the notation $\mathrm{TV}^{\mathcal{I}}(\mathbf{p}, \mathbf{q}) := (1/2) \sum_{i \in \cup_{I \in \mathcal{I}} I} |\mathbf{p}(i) - \mathbf{q}(i)|$ and $\chi^2_{\mathcal{I}}(\mathbf{p} \,\|\, \mathbf{q}) := \sum_{i \in \cup_{I \in \mathcal{I}} I} (\mathbf{p}_i - \mathbf{q}_i)^2/\mathbf{q}_i$.

The asymptotic notation $\tilde{O}$ suppresses logarithmic factors in the argument, so that $\tilde{O}(f(n)) = O(f(n) \log^c f(n))$ for some absolute constant c (and similarly for $\tilde{\Omega}$).

For an integer $k \in [n]$, we denote by $\mathcal{H}^n_k$ the set of k-histogram distributions over $[n]$, i.e., the set of all distributions $\mathbf{p}$ such that there exists a partition $\mathcal{I}$ of $[n]$ into k consecutive intervals (not necessarily of the same size) with $\mathbf{p}$ being uniform on each interval.

2 SAMPLE NEAR-OPTIMAL ALGORITHM

In this section, we describe and analyze the different components of our algorithm, before putting them all together in Section 2.4. To simplify the analysis, and without loss of generality, we will rely on the so-called "Poissonization trick," and assume throughout that the algorithms are provided with $\mathbf{Poi}(m)$ samples from the unknown distribution $\mathbf{p}$ instead of exactly m.[2]

2.1 First Component: Learning in χ^2-Distance

Following the general outline introduced in [8], we first learn the unknown distribution $\mathbf{p}$ before making sure that, *if* $\mathbf{p} \in \mathcal{H}^n_k$ then the output distribution $\hat{\mathbf{p}}$ is close to $\mathbf{p}$ in χ^2 distance, except on a small (but unknown) portion of the domain. The learning stage is accomplished in two steps:

(1) Divide the domain $[n]$ into $K \gg k$ many intervals.
(2) Output a succinct distribution $\hat{\mathbf{p}} \in \mathcal{H}^n_K$. Namely, $\hat{\mathbf{p}}$ is constant on each interval specified by Step 1.

The dividing step is accomplished through the following lemma, whose correctness follows from a standard application of Chernoff bounds. We provide its proof in Appendix A for completeness.

Lemma 2.1 (Claim 1 in [1].). *There exists an algorithm* ***Approx-Part*** *that, given parameters $\delta \in (0, 1]$ and integer $B > 1$, takes $O(B \log(B/\delta))$ samples from a distribution $\mathbf{p}$ on $[n]$ and, with probability at least $1 - \delta$, outputs a partition of $[n]$ into $K \le 8B$ intervals $I_1, I_2, \ldots, I_K$ such that the following holds:*

(1) For each $i \in [n]$ such that $\mathbf{p}(i) > 1/B$, there exists $\ell \in [K]$ such that $I_\ell = \{i\}$. (All "heavy" elements are left as singletons)
(2) Every other interval is such that $\mathbf{p}(I) \le 16/B$.

After the division step, we perform the learning procedure as if we are learning a $\mathcal{H}^n_K$ histogram. In particular, we have the following algorithm which takes the partition $\mathcal{I}$ consisting of the intervals $\{I_1, \cdots, I_K\}$ and outputs $\hat{\mathbf{p}} \in \mathcal{H}^n_K$.

Algorithm 1 Laplace-Learning

Require: Samples from distribution $\mathbf{p}$; partition $\mathcal{I} = \{I_1, I_2, \cdots, I_K\}$ of the domain $[n]$; accuracy ε

1: Let X_j be the number of samples falling in interval I_j.
2: Output the distribution $\hat{\mathbf{p}}$ given by

$$\hat{\mathbf{p}}(i) = \frac{X_j + 1}{|I_j|(\sum_{\ell=1}^{K} X_\ell + K)} \qquad 1 \le j \le K, i \in I_j.$$

Notice that, if $\mathbf{p}$ is indeed a k-histogram, its probability mass is constant on most of the intervals in $\mathcal{I}$, since $|\mathcal{I}| \gg k$. An error is only incurred on a special type of intervals (of which there are at most k) which we refer to as the *break-point intervals*.

Definition 2.2 (Break-point Intervals). Given a k-histogram $\mathbf{p}$, we say that $i \in [n]$ is a break-point of $\mathbf{p}$ if $\mathbf{p}(i) \neq \mathbf{p}(i+1)$ and I a break-point interval if I contains any break-point.

[2] For more on Poissonization, the reader is referred to, e.g., [9, Appendix D.3].

With the definition in mind, we will now specify the formal learning guarantee of the routine.

Lemma 2.3 (Lemma 3.5 of [8]). *Suppose $\mathbf{p} \in \mathcal{H}_k^n$. There exists an algorithm **Laplace-Learning** that, on input the domain size n, a partition of $[n]$ into K intervals $\mathcal{I} = \{I_1, \cdots, I_K\}$ where $K \geq k$ and $\varepsilon \in (0, 1]$, takes $O((K/\varepsilon^2) \cdot \log(1/\delta))$ samples from $\mathbf{p}$ and outputs a distribution $\hat{\mathbf{p}} \in \mathcal{H}_K^n$ that satisfies the following with probability at least $1 - \delta$. Letting $\mathcal{B} = \{\mathcal{B}_1, \cdots, \mathcal{B}_{r-1}\} \subseteq \mathcal{I}$ (with $r \leq k$) denote the breakpoints intervals of $\mathbf{p}$ with respect to $\mathcal{I}$, then $\chi^2_{\mathcal{I}\setminus\mathcal{B}}(\mathbf{p} \| \hat{\mathbf{p}}) \leq \varepsilon^2$.*

2.2 Second Component: Tolerant Identity Testing

Note that we performed the learning step under the assumption that $\mathbf{p}$ were a k-histogram, *which is what we are trying to decide.* The learning step is therefore only guaranteed to succeed in the good case where $\mathbf{p} \in \mathcal{H}_k^n$, and could have dramatically failed otherwise. That is, (1) if $\mathbf{p} \in \mathcal{H}_k^n$, then we will have that $\mathbf{p}$ and $\hat{\mathbf{p}}$ are close *and* (as a consequence of this) that $\hat{\mathbf{p}}$ is close to being a k-histogram. Yet, (2) if $\mathbf{p}$ is far from being a k-histogram, then by the triangle inequality we must have either that $\hat{\mathbf{p}}$ is far from being a k-histogram, or that $\mathbf{p}$ and $\hat{\mathbf{p}}$ are far from each other in total variation distance.

After learning the distribution, we first make sure that the explicit description, which is a K-histogram, is indeed close to a k-histogram. This can be performed efficiently through dynamic programming.

However, as discussed above, it could be the case that $\mathbf{p}$ and $\hat{\mathbf{p}}$ are far away from each other. To verify that, we will use a result of [1] on tolerant identity testing.[3] The tester relies on computing a χ^2-based statistic: given an explicit partition of $[n]$ on K intervals $I_1, \cdots, I_K$ and a fully specified distribution $\mathbf{p}^*$, it takes $\mathbf{Poi}(m)$ samples and computes K (independent) statistics $Z_1, \cdots, Z_K$ defined as

$$Z_j = \sum_{i \in I_j \cap \mathcal{A}_\varepsilon} \frac{(N_i - m\mathbf{p}^*(i))^2 - N_i}{m \cdot \mathbf{p}^*(i)}, \tag{1}$$

where $\mathcal{A}_\varepsilon = \{i \in [n] : \mathbf{p}^*(i) \geq \varepsilon/(50n)\}$ and N_i is the number of occurrences of $i \in [n]$ among the samples drawn from $\mathbf{p}$. Furthermore, we denote their sum as $Z = \sum_{j=1}^K Z_j$. As one can see, the expectation of Z, even after normalized by the number of samples used, is not exactly the χ^2 distance between the two distributions (since some very light intervals are omitted in the computation). For convenience, we define the quantity $\tilde{\chi}^2_{Z,\mathcal{I}}(\mathbf{p} \| \hat{\mathbf{p}}) := \frac{1}{m} \sum_{j=1}^K \mathbf{E}[Z_j]$. Still, the test statistics do enjoy desirable properties, described in the following proposition.

Proposition 2.4 (Adapted from Lemmas 2 and 3 [1]). *The statistic Z has the following guarantees provided that $m \geq C \cdot \sqrt{n}/\varepsilon^2$ where C is a sufficiently large universal constant.*

(1) If $\chi^2(\mathbf{p} \| \mathbf{p}^*) \leq \varepsilon^2$, then $\mathbf{E}[Z] \leq O(m\varepsilon^2)$ and $\mathrm{Var}[Z] \leq O(m^2\varepsilon^4)$.

(2) If $\mathrm{TV}(\mathbf{p}, \mathbf{p}^*) \geq \Omega(\varepsilon)$, then $\mathbf{E}[Z] \geq \Omega(m\varepsilon^2)$ and $\mathrm{Var}[Z] \leq O(\mathbf{E}[Z]^2)$.

[3] Identity Testing, also known as one-sample testing, is a standard tool in distribution testing where one tries to distinguish whether an unknown distribution is the same as a given explicit distribution.

By Chebyshev's Inequality, it is easy to see that, with constant probability, the statistic concentrates around its mean. To go from this constant-probability guarantee to a high-probability one, we can use the so-called Median Trick to boost up the probability of our estimate falling in the desirable interval:

Corollary 2.5 (Median Trick). *Fix an interval j, we can recompute the statistics Z_j for $O(\log(1/\delta))$ many iterations with independent samples and take the median over all iterations. The median $\bar{Z}_j$ then satisfies the following properties*

- ***Markov-Based** It holds that $\bar{Z}_j \leq 3\,\mathbf{E}[Z_j]$ with probability at least $1 - \delta$.*
- ***Chebyshev-Based** If $\mathbf{E}[Z_j] \geq \frac{1}{5}m\varepsilon^2$, where m is the number of samples used to compute Z_j, it holds that $\left|\bar{Z}_i - \mathbf{E}[Z_i]\right| \leq \frac{1}{5}m\varepsilon^2$ with probability at least $1 - \delta$.*

Lastly, given a set of intervals $\mathcal{I} = \{I_1, I_2, \cdots\}$, the statistic $\bar{Z}$ on $\mathcal{I}$ is $\bar{Z} := \sum_{I_j \in \mathcal{I}} \bar{Z}_j$.

2.3 Third Component: Sieving

If we run the testing steps right after the learning step, due to the error incurred in break-point intervals (see Definition 2.2), the identity tester could still fail even if $\mathbf{p} \in \mathcal{H}_k^n$. Our next goal is therefore to "sieve" out those break-point intervals. The idea is to do this adaptively, by running the tester for multiple iterations and using its result to guide the filtering process.

Algorithm 2 Learn-And-Sieve

Require: Samples access to distribution $\mathbf{p}$; a partition $\mathcal{I}$ of the domain $[n]$; accuracy ε; failure probability δ

1: Set $\mathcal{Q} = \{\}$, $\mathcal{I}_{\text{rem}} = \mathcal{I}$
2: $\hat{\mathbf{p}} \leftarrow$ **Laplace-Learning**$(\mathbf{p}, \mathcal{I}, \varepsilon, \delta/100)$.
3: Use Dynamic Programming to check whether there exists $\mathbf{p}^* \in \mathcal{H}_k^n$ such that $\mathrm{TV}(\mathbf{p}^*, \hat{\mathbf{p}}) \leq \varepsilon$; otherwise, return Reject.
4: ▷ Can be done in time $\mathrm{poly}(k, 1/\varepsilon)$ as shown in Lemma 4.11 of [10].
5: Compute the statistic $\bar{Z}$ (median) on $\mathcal{I}$ with $m = \Theta(\sqrt{n}/\varepsilon^2 \cdot \log(1/\delta))$ samples from $\mathbf{p}$ and the explicit description $\hat{\mathbf{p}}$ as specified in Eq. (1) and Corollary 2.5.
6: Let $\alpha \leftarrow m\varepsilon^2$, $T_{\text{iter}} \leftarrow 1$
7: **while** $\bar{Z} > 4\alpha$ and $T_{\text{iter}} < 12 \log k$ **do**
8: Search for $\mathcal{R} \subseteq \mathcal{I}_{\text{rem}}$ such that $|\mathcal{R}| \leq k$ and $\sum_{i \in \mathcal{I}_{\text{rem}} \setminus \mathcal{R}} \bar{Z}_i \leq 3\alpha$; Reject if not possible
9: Add the set of bad intervals $\mathcal{R}$ to $\mathcal{Q}$.
10: Remove those bad intervals from $\mathcal{I}_{\text{rem}}$.
11: Compute $\bar{Z}$ on $\mathcal{I}_{\text{rem}}$ with $\Theta(\sqrt{n}/\varepsilon^2 \cdot (\log(1/\delta) + \log k))$ new samples, using the median trick.
12: $T_{\text{iter}} \leftarrow T_{\text{iter}} + 1$
13: **end while**
14: Compute one last time $\bar{Z}$ on $\mathcal{I}_{\text{rem}}$ with $\Theta(\sqrt{n}/\varepsilon^2 \cdot \log(1/\delta))$ new samples.
15: Reject if $\bar{Z} > 30\alpha$.
16: **return** $\mathcal{Q}$, $\hat{\mathbf{p}}$.

Loosely speaking, the output distribution will be close to the input distribution in χ^2 distance on some "selected" interval if the

input distribution is indeed a k-histogram. The guarantee of the sieving is summarized in the following lemma and its proof follows from the analysis of the sieving stage in section 3.2.1 of [8].

Lemma 2.6 (Sieving Lemma). *Assume Algorithm 2 is given a partition $\mathcal{I}$ satisfying $|\mathcal{I}| \geq 100k \log k$ and outputs the set of intervals $\mathcal{Q}$. Moreover, denote $\mathcal{G} = \mathcal{I} \backslash \mathcal{Q}$.*

- *Suppose $\mathbf{p} \in \mathcal{H}_k^n$. Then the algorithm returns $\hat{\mathbf{p}}$ and $\mathcal{Q}$ such that $\widehat{\chi}^2_{Z,\mathcal{I}\backslash\mathcal{Q}}(\mathbf{p} \| \hat{\mathbf{p}}) \leq \varepsilon$ with probability at least $1-\delta$.*
- *Suppose $\mathrm{TV}^{\mathcal{I}\backslash\mathcal{Q}}(\mathbf{p}, H) > 40\varepsilon$ for every $H \in \mathcal{H}_k^n$ and any $\mathcal{Q}$ such that $\mathcal{Q} \subseteq \mathcal{I}$ and $|\mathcal{Q}| \leq 12k \log k$. Then, the algorithm outputs Reject with probability at least $1-\delta$.*

Lastly, the algorithm uses at most $O((\sqrt{n}/\varepsilon^2) \cdot \log k \cdot \log(k/\delta) + |\mathcal{I}| \log(1/\delta)/\varepsilon^2)$ samples.

2.4 Final Algorithm

The algorithm **Learn-and-Sieve** from Lemma 2.6 will be crucial for our final testing algorithm, which it will use as a subroutine. On one hand, we need to sieve out the break-point intervals; on the other hand, we need to be careful that in total we should move no more than a $O(\varepsilon)$ portion of the total probability mass. If we run the sieving procedure once, we will remove a constant portion of the entire distribution. To achieve more fine-grained sieving procedure, we can sub-divide the problematic intervals and invoke the sieving procedure iteratively. Combining everything together then gives our main algorithm. Now, we are ready to give the main testing algorithm (Algorithm 3).

Algorithm 3 Divide-And-Learn-And-Sieve

Require: Sample access to the distribution $\mathbf{p}$

1: Set $\mathcal{I}^{(0)} = \mathcal{B}^{(0)} = \{[n]\}$.
2: $t \leftarrow 0$
3: **while** $\mathbf{p}(\mathcal{B}^{(t)}) := \sum_{I \in \mathcal{B}^{(t)}} \mathbf{p}(I) \geq \varepsilon$ **do**
4: $\quad \mathcal{S}^{(t+1)} \leftarrow$ **Approx-Divide**$\left(1000k \log k, \mathcal{B}^{(t)}, \frac{1}{100\log(1/\varepsilon)}\right)$.
5: $\quad$ Set $\mathcal{I}^{(t+1)} = \mathcal{I}^{(t)} \backslash \mathcal{B}^{(t)} \cup \mathcal{S}^{(t+1)}$. ▷ Note that $\mathcal{I}^{(t+1)}$ is still a partition of $[n]$.
6: $\quad \mathcal{Q}^{(t+1)}, \hat{\mathbf{p}}^{(t+1)} \leftarrow$ **Learn-And-Sieve**$\left(\mathcal{I}^{(t+1)}, \varepsilon/\log(1/\varepsilon)\right)$
7: $\quad$ Reject if **Learn-And-Sieve** outputs Reject.
8: $\quad$ Set $\mathcal{B}^{(t+1)} = \mathcal{S}^{(t+1)} \cap \mathcal{Q}^{(t+1)}$.
9: $\quad t \leftarrow t+1$
10: **end while**
11: Denote $\mathcal{G}^{(j)} = \mathcal{S}^{(j)} \backslash \mathcal{B}^{(j)}$ for all $j \geq 1$. ▷ Note that the union of $\mathcal{G}^{(1)}, \cdots \mathcal{G}^{(t)}, \mathcal{B}^{(t)}$ forms a partition of the domain $[n]$.
12: We will consider the measure $\widetilde{\mathbf{p}}$ such that on intervals $I \in \mathcal{G}^{(j)}$, $\widetilde{\mathbf{p}}(i) = \hat{\mathbf{p}}^{(j)}(i)$; and on intervals $I \in \mathcal{B}^{(t)}$, $\widetilde{\mathbf{p}}(i) = \hat{\mathbf{p}}^{(t)}(i)$.
13: Use Dynamic Programming to check whether there is a k-histogram that is 10ε-close to $\widetilde{\mathbf{p}}$ in TV distance. Otherwise Reject.
14: Compute the statistic $\bar{Z}$ on the partition $\mathcal{G}^{(1)} \cup \cdots \cup \mathcal{G}^{(t)}$ with $m = \Theta(\sqrt{n}/\varepsilon^2)$ samples from $\mathbf{p}$ and the explicit description $\widetilde{\mathbf{p}}$ as specified in Eq. (1). Output Accept if $\bar{Z} \leq O(\varepsilon^2)$. Otherwise Reject.

Proof of Theorem 1.1. We first argue that the algorithm terminates in $O(\log(1/\varepsilon))$ rounds with high probability. By Lemma 2.1, we have $\mathbf{p}(I) \leq \frac{16}{1000k \log k} \cdot \mathbf{p}(\mathcal{B}^{(t)})$ for every interval $I \in \mathcal{S}^{(t+1)}$ with probability at least $1 - \frac{1}{100\log(1/\varepsilon)}$. Since the subroutine **Learn-And-Sieve** selects (removes) at most $50k \log k$ intervals, it holds that $\mathbf{p}(\mathcal{B}^{(t+1)}) \leq \mathbf{p}(\mathcal{Q}^{(t+1)}) \leq 0.8\mathbf{p}(\mathcal{B}^{(t)})$. Hence, $\mathcal{B}^{(t)}$ will drop below ε after at most $12\log(1/\varepsilon)$ iterations with probability at least $(1 - \frac{1}{200\log(1/\varepsilon)})^{12\log\frac{1}{\varepsilon}} \geq \frac{9}{10}$. Conditioning on the algorithm terminates in $O(\log(1/\varepsilon))$ iterations, we proceed to argue it outputs the correct testing result with high probability.

Completeness. At the t-th iteration, we claim that, with probability at least $1 - 1/(100\log(1/\varepsilon))$, it holds

$$\widehat{\chi}^2_{Z,\mathcal{G}^{(t)}}\left(\mathbf{p} \,\middle\|\, \hat{\mathbf{p}}^{(t)}\right) \leq O(\varepsilon^2/\log^2(1/\varepsilon)), \tag{2}$$

where $\mathcal{G}^{(t)} = \mathcal{S}^{(t)} \backslash \mathcal{B}^{(t)}$ as defined in Algorithm 11 of Algorithm 3. By Lemma 2.6, it holds

$$\widehat{\chi}^2_{Z,\mathcal{I}^{(t)}\backslash\mathcal{Q}^{(t)}}\left(\mathbf{p} \,\middle\|\, \hat{\mathbf{p}}^{(t)}\right) \leq O(\varepsilon^2/\log^2(1/\varepsilon)).$$

Since $\mathcal{G}^{(t)} = \mathcal{S}^{(t)} \backslash \mathcal{B}^{(t)}$ is a subset of $\mathcal{I}^{(t)} \backslash \mathcal{Q}^{(t)}$, the claim in Eq. (2) follows. Recall that the algorithm runs for at most $O(\log(1/\varepsilon))$ iterations. Combining this with Eq. (2), if we denote $\mathcal{G} = \bigcup \mathcal{G}^j$ for $1 \leq j \leq t$, it holds

$$\widehat{\chi}^2_{Z,\mathcal{G}}(\mathbf{p} \,\|\, \widetilde{\mathbf{p}}) \leq \sum_{j=1}^{t} \widehat{\chi}^2_{Z,\mathcal{G}^{(j)}}\left(\mathbf{p} \,\middle\|\, \hat{\mathbf{p}}^{(j)}\right) \leq O(\varepsilon^2) \tag{3}$$

with probability at least $(1 - \frac{1}{200\log(1/\varepsilon)})^{12\log(1/\varepsilon)} \geq \frac{9}{10}$. Conditioning on Equation (3), by Markov's Inequality, line 14 will output accept with probability at least $9/10$. Then, as Eq. (3) together with the fact that $\mathbf{p}(\mathcal{B}^{(t)}) \leq \varepsilon$ also implies $\mathrm{TV}(\mathbf{p}, \widetilde{\mathbf{p}}) \leq O(\varepsilon)$ and $\mathbf{p} \in \mathcal{H}_k^n$, line 13 will also pass. Overall, the algorithm accepts with probability at least $3/5$.

Soundness. Suppose now that $\mathrm{TV}(\mathbf{p}, H) \geq C \cdot \varepsilon$ for every $H \in \mathcal{H}_k^n$, where C is some sufficiently large absolute constant. For the sake of contradiction, assume that line 13 and line 14 both pass. By line 14, we have $\widehat{\chi}^2_{Z,\mathcal{G}}(\mathbf{p} \,\|\, \widetilde{\mathbf{p}}) \leq O(\varepsilon)$ with probability at least $\frac{9}{10}$. This implies that $\mathrm{TV}^{\mathcal{G}}(\mathbf{p}, H) \leq O(\varepsilon)$ by Proposition 2.4. Since $\mathbf{p}(\mathcal{B}^{(t)}) \leq \varepsilon$, it holds that $\mathrm{TV}(\mathbf{p}, H) \leq O(\varepsilon)$. Then, by line 13, there exists a k-histogram H satisfying $\mathrm{TV}(H, \widetilde{\mathbf{p}}) \leq O(\varepsilon)$. By the triangle inequality, we have $\mathrm{TV}(H, \mathbf{p}) \leq O(\varepsilon)$. This contradicts the assumption that $\mathrm{TV}(H, \mathbf{p}) \geq C \cdot \varepsilon$. Hence, at least one of the two lines will output reject when $\mathrm{TV}(H, \widetilde{\mathbf{p}}) \geq \Omega(\varepsilon)$ for any $H \in \mathcal{H}_k^n$.

Finally, the samples are consumed by three different types of routines – dividing, learning, and testing. By Lemma 2.1, in one iteration, the dividing phase consumes $O\left(k \log^2 k D\left(\mathcal{B}^{(t)}\right)^{-1}\right)$ samples where $\mathbf{p}(\mathcal{B}^{(t)})$ is the mass of the the to-be-divided intervals at the t-th iteration. It is not hard to see the mass shrinks exponentially in every iteration. Thus, the samples consumed are dominated by the last iteration. Hence, at most $O\left(\frac{k}{\varepsilon} \log^2 k\right)$ samples are consumed by the dividing phase in total.

By Lemma 2.6, at the t-th iteration, the **Learn-And-Sieve** procedure consumes

$$O\left(\frac{\sqrt{n}}{\varepsilon^2}\cdot\log k\cdot(\log k+\log(1/\varepsilon))+t\cdot\frac{k\log k}{\varepsilon^2}\cdot\log(1/\varepsilon)\right) \quad (4)$$

samples. Thus, in total, samples used by learning and testing are

$$O\left(\frac{\sqrt{n}}{\varepsilon^2}\cdot\log k\cdot\log(1/\varepsilon)\cdot\log(k/\varepsilon)+\log(1/\varepsilon)^3\cdot\frac{k\log k}{\varepsilon^2}\right) \quad (5)$$

if we sum up the samples consumed from $t=1$ to $12\log(1/\varepsilon)$. Combining the samples used by dividing, learning and testing, the overall sample complexity is bounded by

$$O\left(\frac{\sqrt{n}}{\varepsilon^2}(\log k)\log(1/\varepsilon)\log(k/\varepsilon)+\frac{k\log k}{\varepsilon^2}\log(1/\varepsilon)^3+\frac{k}{\varepsilon}\log^2 k\right) \quad (6)$$

which concludes the proof. □

3 SAMPLE COMPLEXITY LOWER BOUND

As discussed in the introduced, it was shown it [8] how to derive a sample complexity lower bound for histogram testing from one on *support estimation*, essentially by "randomly embedding" a support estimation task in a slightly larger domain, so that with high probability each element of the support is isolated and becomes its own (singleton) interval of a histogram. However, such a direct, blackbox reduction intrinsically leads to a sub-optimal dependency on ε in the resulting lower bound. To remedy this, our starting point is instead a variant of the support estimation problem, which we define and term *Bump Testing*, on domain $[k]$, which will be parameterized by $\varepsilon\in(0,0.01]$ (its correspondance shall become clear after we connect it to the original histogram testing problem). Moreover, as standard in the distribution testing literature, we will first obtain our lower bound under the so-called "Poisson Sampling Model," where instead of m samples from $\mathbf{p}$ we are provided with $\mathbf{Poi}(m\mathbf{p}_i)$ samples for each element i; and then connect it back to the fixed-sample-size ("Multinomial") model. Under the Poisson Sampling Model, the object of interest need no longer be a probability distribution, leading us to consider approximate probability vectors (finite measures) whose deviation from having sum 1 is parameterized by an additional quantity ν.

Definition 3.1 (Bumpy approximate probability vector). For $0<\varepsilon<1$, $0<\nu<1$, define the set of *bumpy approximate probability vectors* on domain $[k]$ as

$$\mathcal{M}_k^\varepsilon(\nu):=\left\{P\in\left(\left\{\frac{1}{k}\right\}\cup\left[\frac{1+\varepsilon(1+\nu)}{k},1\right]\right)^k:|\|P\|_1-1|\le\nu\right\}.$$

We are interested in the sample complexity lower bound of testing the number of "bumps" (i.e., for $P=(p_1,\dots,p_k)\in\mathcal{M}_k^\varepsilon(\nu)$, the number $|\{i:p_i>1/k\}|$) given Poisson sample access to an unknown bumpy approximate probability vector. The formal definition for *Bump Testing* is given below.

Definition 3.2 (Bump testing). For $0<\nu<1$, $0<\varepsilon<1$, $0\le s_1<s_2\le k$ define the sample complexity of bump testing as

$$m^{\mathrm{poi}}_{\mathrm{bump}}(k,s_1,s_2,\varepsilon,\nu):=\min\Big\{m\ge0:\exists\hat{I},\text{s.t.}\Pr\left[\hat{I}=1,\mathbb{1}\{S(P)<s_1\}\right]+\Pr\left[\hat{I}=0,\mathbb{1}\{S(P)>s_2\}\right]\le\frac{1}{10},\forall P\in\mathcal{M}_k^\varepsilon(\nu)\Big\},$$

where $\hat{I}$ is an integer-valued estimator measurable with respect to $N=(N_1,N_2,\dots)\sim\mathrm{Poi}(mp_i)$ and $S(P)$ counts the number of positions such that $p_i>1/k$.[4]

The idea behind the lower bound argument of *Bump Testing* is based on the observation that if we mix a measure with the uniform measure, the non-zero elements in the original distribution become "bumps" in the mixture distribution, and those small bumps will be statistical hard to distinguish from the uniform "baseline." Following this high-level idea, we will construct the hard instance for *Bump Testing* by mixing a hard instance for support size testing with the uniform measure.

It remains to define this hard instance for support size. To do so, we follow the outline from [34], which reduces the question to a univiariate problem: defining each $\mathbf{p}_i$ independently from a suitable random variable. Namely, we first obtain two random variables U,U' in range $[0,\lambda]$ whose probability mass functions differ at 0 but have many *matching moments*. We will then take k i.i.d. copies of those random variables to randomly generate the adversarial measures P,P', defining our (randomized) hard-to-distinguish instances. The random variables U,U' will be constructed from the following Linear Program.

Fix parameters $\lambda=\lambda(k,\varepsilon)>0$, $\nu\in(0,1)$, and $L=L(k,\varepsilon)\in\mathbb{N}$. U,U' will be two random variables supported on $\{0\}\cup[1+\nu,\lambda]\subseteq[0,\lambda]$ solution of

$$\begin{aligned}\text{maximize}\ &\Pr[U=0]-\Pr[U'=0]\\ &\mathbf{E}[U]=\mathbf{E}[U']=1\\ &\mathbf{E}[U^\ell]=\mathbf{E}[U'^\ell],\qquad 1\le\ell\le L.\end{aligned}$$

We then define V,V' by $V:=\frac{m}{k}(1+\varepsilon U)$, $V':=\frac{m}{k}(1+\varepsilon U')$ The random variables enjoy the following property:

Lemma 3.3. *Let V,V' as above. Then, if $\frac{k}{\varepsilon^2}\cdot\frac{L}{\lambda^2}>m$, we have*

$$\mathrm{TV}\left(\mathbf{E}[\mathrm{Poi}(V)],\mathbf{E}[\mathrm{Poi}(V')]\right)\le e^{-L/4}.$$

With this technical lemma in hand, we will establish the lower bound for *Bump Testing*. The argument is analogous to the one in [34] where one first bounds the TV between the distributions of Poisson samples and then applies Le Cam's two-point method [6].

Lemma 3.4. *For $r\in[0.1,0.4]$, $\nu\in(0,1)$ and $\varepsilon\le\frac{\nu}{20}$, there exists some $0<s_1<s_2<k$ where $s_2-s_1\ge r\cdot k$ such that*

$$m^{\mathrm{poi}}_{\mathrm{bump}}(k,s_1,s_2,\varepsilon,\nu)\ge\Omega\left(\frac{k}{\varepsilon^2\log k}\right).$$

Proof. Let U,U' be random variables satisfying $U,U'\in\{0\}\cup[1+\nu,\lambda]$, $\mathbf{E}[U]=\mathbf{E}[U']=1$, $\mathbf{E}[U^j]=\mathbf{E}[U'^j]$ for all $j\in[L]$ where L,λ are parameters to be specified. By Lemma 5 in [34], such a pair of variables exist for

$$d:=\left|\Pr[U>0]-\Pr[U'>0]\right|=\frac{\left(1+\sqrt{\frac{1+\nu}{\lambda}}\right)^2}{1+\nu}\cdot\left(\frac{1-\sqrt{\frac{1+\nu}{\lambda}}}{1+\sqrt{\frac{1+\nu}{\lambda}}}\right)^L.$$

[4] In other words, $\hat{I}$ separates with probability at least 9/10 the cases (1) the approximate probability vector has a lot of bumps from (2) the vector has few bumps.

We proceed to construct the vectors

$$B = \left(\frac{1+\varepsilon U_1}{k}, \cdots, \frac{1+\varepsilon U_k}{k}\right), \quad B' = \left(\frac{1+\varepsilon U'_1}{k}, \cdots, \frac{1+\varepsilon U'_k}{k}\right),$$

where U_i are i.i.d. copies of U. Since $\mathbf{E}[U_i] = 1$, we have that $\sum_{i=1}^{k} \mathbf{E}[B_i] = 1+\varepsilon$. As $\varepsilon \le \frac{\nu}{20}$, it is not hard to check that, by Markov's inequality, B and B' are indeed bumpy approximate probability vectors with high probability. We then define the following high probability events,

$$E := \left\{ |S(B) - \mathbf{E}[S(B)]| \le \alpha k d \right\}, E' := \left\{ \left|S(B') - \mathbf{E}[S(B')]\right| \le \alpha k d \right\}.$$

Under events E, E', it holds $|S(B) - S(B')| \ge (1-2\alpha)kd$ (by triangle inequality).

In order to apply Le Cam's Lemma [6], we want

$$(1-2\alpha)kd \ge r \cdot k \tag{7}$$

(essentially saying that the vectors B and B' correspond to the two different cases of *Bump Testing*).

Since $d \ge \frac{1}{1+\nu} \cdot \left(1 - \sqrt{\frac{1+\nu}{\lambda}}\right)^L \left(1 + \sqrt{\frac{1+\nu}{\lambda}}\right)^{-L}$, $r < 0.4$ and $\nu \le 1$, if we set $\alpha := \frac{1}{20}$, a sufficient condition for Eq. (7) is

$$\left(1 - \sqrt{\frac{1+\nu}{\lambda}}\right)^L \left(1 + \sqrt{\frac{1+\nu}{\lambda}}\right)^{-L} \ge 0.4 \cdot \frac{10}{9} \cdot 2 \approx 0.9,$$

which is equivalent to $\sqrt{\frac{1+\nu}{\lambda}} \ge \frac{1+0.9^{1/L}}{1-0.9^{1/L}}$. We now set $L := 5\log k$. As a result, one can check that it suffices to set

$$\lambda \ge \Omega(\log^2 k) \tag{8}$$

Therefore, under the events, B, B' are valid instances for the bump testing problem with a gap at least rk between their bump numbers. It remains to show it is hard for any tester to separate the two. The inputs seen by the tester are vectors $(N_1, N_2, \cdots N_k)$ denoting the number of samples falling in each bucket after taking $\mathbf{Poi}(m)$ samples. We will use N, N' to denote the samples from B, B' respectively. Furthermore, we denote their distributions as P_N, $P_{N'}$ and their distributions conditioned on the events E, E' as $P_{N|E}, P_{N'|E'}$. We proceed to bound the TV between the two conditional distributions.

$$\begin{aligned}\mathrm{TV}\left(P_{N|E}, P_{N'|E'}\right) &\le \mathrm{TV}\left(P_{N|E}, P_N\right) + \mathrm{TV}\left(P_{N'|E'}, P_{N'}\right) + \mathrm{TV}(P_N, P_{N'}) \\ &\le \Pr(E^c) + \Pr(E'^c) + \mathrm{TV}(P_N, P_{N'}). \end{aligned} \tag{9}$$

Notice that, by definition of B, B', it holds $\mathrm{TV}(P_N, P_{N'}) \le k \cdot TV(\mathbf{E}[\mathbf{Poi}(\frac{m}{k}(1+\varepsilon U))], \mathbf{E}[\mathbf{Poi}(\frac{m}{k}(1+\varepsilon U'))])$. Recall that we have set $L = 5\log k$. Hence, by Lemma 3.3, the term is $o(1)$ when $m = \frac{k}{\varepsilon^2} \cdot \frac{5\log k}{\lambda^2}$. Also recall that we need to set $\lambda = \Theta(\log^2 k)$ to satisfy Eq. (7). Thus, we must have $m = \Omega(k/(\varepsilon^2 \log k))$. On the other hand, recall that U, U' lie in $(0, \lambda]$. Thus, their variances are bounded by λ^2. Hence, Chebyshev's inequality yields that

$$\Pr(E^c), \Pr(E'^c) \le \frac{\lambda}{k^2\alpha^2 d^2} \le O\left(\frac{\log^2 k}{k^2}\right) \le o(1). \tag{10}$$

where the second inequality follows from $\alpha = \frac{1}{20}$, $d \ge \frac{r}{1-2\alpha}$ and $r \ge 0.1$. Combining Eqs. (10) and (9) yields that $\mathrm{TV}\left(P_{N|E}, P_{N'|E'}\right) \le 6/10$ when k is large enough. Applying Le Cam's Lemma [6] enables us to conclude that no tester can distinguish B, B' with failure probability less than $1/5$ under $E \cap E'$ when $m < O(\frac{k}{\varepsilon^2 \log k})$. □

The last step of our lower bound is to go back from *bump testing* to histogram testing; for the latter, the domain n will be bigger than k. We start by defining the notion of ℓ-histogram approximate probability vectors below.

Definition 3.5 (Histogram approximate probability vector).

$$\tilde{\mathcal{H}}^n_\ell(\nu) := \left\{ P =\in [0,\infty) : |\|P\|_1 - 1| \le \nu, \frac{P}{\|P\|_1} \in \mathcal{H}^n_\ell \right\}$$

We define the Poissonized Histogram Testing problem formally.

Definition 3.6 (Histogram Testing under Poisson Model). For $0 < \nu < 1$, $0 < \varepsilon < 1$, define the sample complexity of histogram testing (under Poisson model) as

$$\begin{aligned} m^{\mathrm{poi}}_{\mathrm{hist}}(n, \ell, \varepsilon, \nu) := \min \Big\{ & m \ge 0 : \exists \hat{I}, \\ & \text{s.t.} \Pr\left[\hat{I} = 1, \mathbb{1}\{ \inf_{P' \in \tilde{\mathcal{H}}^n_\ell(\nu)} \mathrm{TV}(P', P) \ge \varepsilon(1+\nu)\}\right] \\ & + \Pr\left[\hat{I} = 0, \mathbb{1}\left\{P \in \tilde{\mathcal{H}}^n_\ell(\nu)\right\}\right] \le 0.1, \\ & \text{for all } P \text{ s.t. } |\|P\|_1 - 1| \le \nu \Big\}, \end{aligned}$$

where $\hat{I}$ is an integer-valued estimator measurable with respect to $N = (N_1, N_2, \dots) \sim \mathrm{Poi}(mp_i)$.

Following [8], we can obtain the lower bound via reduction. Specifically, with a ℓ-histogram tester where $\ell := 2s_1 + 1$, we can use it to solve the bump testing problem as long as $s_2 - s_1 > rk$.

Lemma 3.7. *Assuming $s_2 - s_1 \ge \frac{7}{60}k$, we have $m^{\mathrm{poi}}_{\mathrm{bump}}(k, s_1, s_2, \nu) \le m^{\mathrm{poi}}_{\mathrm{hist}}(100k, 2s_1+1, \varepsilon, \nu)$.*

Proof. Suppose that B_1, B_2 are a pair of hard distributions for Bump Testing with the bump numbers $S(B_1) \le s_1, S(B_2) \ge s_2$ with s_1, s_2 satisfying the condition of the lemma, and set $r := (s_2 - s_1)/k \ge \frac{21}{180}$. We then embed the distribution in a larger domain $[n]$ where $n = 100k$ and fill the rest of the domains with the "uniform measure." In particular, we obtain a pair of new measures $P_i(j) = \frac{1}{100}B_i(j)$ when $j \le k$ and $P_i(j) = \frac{1}{100k}$ when $j > k$. It is easy to see that the new measures satisfy $|\|P_i\|_1 - 1| \le \frac{1}{100}\nu$. Then, pick a permutation σ uniformly from all possible ones on the domain $[n]$ and apply it on the measures P_i. Define $H(P) := \min_\ell P \in \tilde{\mathcal{H}}^n_\ell(\nu)$. We can easily see that $H(\sigma(P_1)) \le 2s_1 + 1$. Namely, after permutation, the distribution is at most a $2s_1+1$ histogram. On the other hand, $H(\sigma(P_2)) \ge 2\frac{9}{10}s_2 + 1$ with probability at least $9/10$ by Lemma 4.4 in [8]. Hence, with high probability, it holds

$$\begin{aligned} H(\sigma(B_2)) - H(\sigma(B_1)) &\ge 2\left(\frac{9}{10}s_2 - s_1\right) = 2\left(\frac{9}{10}(s_2 - s_1) - \frac{1}{10}k\right) \\ &= 2\left(\frac{9}{10}rk - \frac{1}{10}k\right) \ge \frac{1}{10}k \end{aligned}$$

where the last inequality uses our lower bound on $r \ge \frac{7}{60}$. If we pick $\ell := 2s_1 + 1$, it is easy to see that $\sigma(P_1)$ is always an ℓ-histogram. On

the other hand, by Lemma 4.4 in [8], $\sigma(P_2)$ is at least an $(\ell + \frac{1}{20}k)$-histogram with high probability. To "prune" it into an ℓ-histogram, we have to remove at least $\Omega(k)$ isolated bumps that have mass at least $\frac{1}{100k} + \frac{\varepsilon}{100k}(1+\nu)$. Hence, $\inf_{P' \in \tilde{\mathcal{H}}^n_\ell(\nu)} \mathrm{TV}(P', \sigma(P_2)) \geq \Omega(\varepsilon(1+\nu))$. If we use a histogram tester to examine whether $\sigma(P_i)$ is an ℓ-histogram, we can tell whether we are given B_1 or B_2 according to whether they are $2s_1 + 1$ histogram after permutation. Therefore, by reduction, we show the corresponding histogram testing problem $m^{\mathrm{poi}}_{\mathrm{hist}}(100k, \ell, \varepsilon, \nu)$ requires at least as many samples as $m^{\mathrm{poi}}_{\mathrm{test}}(k, s_1, s_2, \nu)$. □

Combining the reduction with Lemma 3.4 gives the $\Omega(\frac{k}{\varepsilon^2 \log k})$ result.

Theorem 3.8. *Any algorithm that decides whether an unknown distribution on $[k]$ is an s-histogram or ε-far from any such distribution in total variation with probability at least $2/3$ must use $\Omega(\frac{k}{\varepsilon^2 \log k} + \sqrt{n}/\varepsilon^2)$ samples.*

Proof. By Lemmas 3.4 and Lemmas 3.7, it holds we need at least $\Omega(k/(\varepsilon^2 \log k))$ samples to distinguish whether a distribution on $[k]$ is an s-histogram with probability at least $2/3$ under the Poisson sampling model.

We proceed to argue for the sample complexity lower bound under standard sampling. Suppose we are given a tester for fixed sample size such that it succeeds with high probability as long as it is given more than m^* samples. Assume that we want to use it to solve the hard instance P under Poisson sampling model with $\tilde{m} \sim \mathbf{Poi}(m^{\mathrm{poi}}_{\mathrm{hist}}(n, \ell, \varepsilon)\,\|P\|_1)$ samples. We can construct an estimator which invokes the fixed sample size tester whenever $\tilde{m} \geq m^*$ and outputs fail otherwise.

By our lower bound result for the Poisson sampling model, the estimator cannot succeed with high probability. On the other hand, the tester succeeds with high probability whenever $\tilde{m} > m^*$. Together this implies that

$$m^* \geq (1-\nu) \cdot m \cdot \left(1 - O\left((1-\nu)^{-1} \cdot m^{-\frac{1}{2}}\right)\right),$$

where $m := m^{\mathrm{poi}}_{\mathrm{hist}}(n, \ell, \varepsilon, \nu) = \Omega\big(k/(\varepsilon^2 \log k)\big)$. Choosing $\nu = 1/10$ gives that $m_{\mathrm{hist}}(n, \ell, \varepsilon) > \Omega(k/(\varepsilon^2 \log k))$. Finally, by Proposition 4.1 in [8], we have $m_{\mathrm{hist}}(n, \ell, \varepsilon) > \Omega(\sqrt{n}/\varepsilon^2)$. This concludes the proof. □

4 APPLICATION TO MODEL SELECTION

In this section, we detail how our testing algorithm, by a standard reduction, can be used for *model selection*, i.e., to select a suitable parameter k in order to succinctly represent the data.

Theorem 4.1. *There exists an algorithm which, given samples from an unknown distribution $\mathbf{p}$ on $[n]$, an error parameter $\varepsilon \in (0,1]$, and a failure probability $\delta \in (0,1]$, outputs a parameter $1 \leq K \leq n$ such that the following holds. Denote by $1 \leq k \leq n$ the smallest integer such that $\mathbf{p} \in \mathcal{H}^n_k$. With probability at least $1-\delta$, (1) the algorithm takes $\widetilde{O}((\sqrt{n}+k)/\varepsilon^2 \cdot \log(1/\delta))$ samples, (2) $1 \leq K \leq 2k$, and (3) $\mathbf{p}$ is at TV distance at most ε from some $H \in \mathcal{H}^n_K$. Moreover, the algorithm runs in time $\mathrm{poly}(m)$, where m is the number of samples taken.*

Before proving the theorem, we discuss the various aspects of its statement. (1) guarantees that, with high probability, the algorithm does not take more samples than what it would if it were *given* k and just needed to test whether it was the right value. Moreover, as most tasks on k-histograms require at least a number of samples growing linearly in k, this guarantes that whenever $k \gg \sqrt{n}$ the cost of the model selection step is negligible in front of all subsequent tasks. Item (2), ensures that the output of the algorithm is a good approximation of the true, *optimal* value k; that is, that the model selected is essentially as succinct as it gets. Finally, (3) guarantees that even when the output is such that $K \ll k$, the parameter K is still good: that is, approximating $\mathbf{p}$ by a K-histogram still leads to a sufficiently accurate representation (even though it is even more succinct that what the *true* parameter k would yield).

Proof. The model selection algorithm is quite simple, and works by maintaining a current "guess" K as part of a doubling search, and using the tester of to check if the current value is good. Specifically: for $0 \leq j \leq \lceil \log_2 n \rceil$, run the testing algorithm of Theorem 1.1 (**Divide-And-Learn-And-Sieve**) with parameters $n, \varepsilon, 2^j$, and $\delta_j := \frac{\delta}{2(j+1)^2}$. If the testing algorithm returns Accept, then return $K := 2^j$ and stop; otherwise, continue the loop.

Let $L \leq \lceil \log_2 n \rceil$ be the number of iterations before the algorithm stops. By a union bound, the probability that all L invocations of the testing algorithm behaved as they should is, by a union bound, at least

$$1 - \sum_{j=0}^{L} \delta_j = 1 - \sum_{j=0}^{L} \frac{\delta}{2(j+1)^2} \geq 1 - \sum_{j=0}^{\infty} \frac{\delta}{2(j+1)^2} \geq 1 - \delta.$$

Condition on this being the case. Since the algorithm, as soon as $j \geq \lceil \log_2 k \rceil$, returns Accept (since then $\mathbf{p} \in \mathcal{H}^n_{2^j}$), we get that $L := \lceil \log_2 k \rceil$, and that the output satisfies $K \leq 2k$, giving (2). Moreover, if $K < k$, then by the guarantee of the testing algorithm this means that $\mathbf{p}$ was *not* ε-far from $\mathcal{H}^n_K$, showing (3). Finally, the sample complexity is then

$$\sum_{j=0}^{L} \tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^2} \log \frac{1}{\delta_j} + \frac{2^j}{\varepsilon^2} \log \frac{1}{\delta_j}\right) = \tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^2} + \frac{k}{\varepsilon^2}\right) \log \frac{1}{\delta}$$

since $L = O(\log k)$. This shows (1), and concludes the proof. □

As a final remark, we note that the guarantee provided in (2) can be improved to the optimal $1 \leq K \leq k$, by modifying slightly the above procedure. Namely, after finding some K such that $1 \leq K \leq 2k$ as before, one can run the testing algorithm for $K/2 \leq i \leq K$ (not a binary search anymore), each time with parameters n, ε, i, and $\delta_i = \delta/K$. By a union bound, this incurs an extra δ probability of failure, and an additional $\tilde{O}\big((\sqrt{n}+k)/\varepsilon^2 \cdot \log(1/\delta)\big)$ samples overall, but now the output after this second step will be guaranteed to be at most k (with high probability).

REFERENCES

[1] J. Acharya, C. Daskalakis, and G. Kamath. 2015. Optimal Testing for Properties of Distributions. In *NeurIPS*. 3591–3599.

[2] J Acharya, I. Diakonikolas, C. Hegde, J. Li, and L. Schmidt. 2015. Fast and Near-Optimal Algorithms for Approximating Distributions by Histograms. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015*, Tova Milo and Diego Calvanese (Eds.). ACM, 249–263.

[3] J. Acharya, I. Diakonikolas, J. Li, and L. Schmidt. 2017. Sample-Optimal Density Estimation in Nearly-Linear Time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*. 1278–1289. Full version available at https://arxiv.org/abs/1506.00671.

[4] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. 2000. Testing that distributions are close. In *IEEE Symposium on Foundations of Computer Science*. 259–269.

[5] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. 2013. Testing Closeness of Discrete Distributions. *J. ACM* 60, 1 (2013), 4.

[6] L. Le Cam. 2012. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media.

[7] C. L. Canonne. [n.d.]. A short note on Poisson tail bounds. https://github.com/ccanonne/probabilitydistributiontoolbox/blob/master/poissonconcentration.pdf

[8] C. L. Canonne. 2016. Are Few Bins Enough: Testing Histogram Distributions. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS '16)*. Association for Computing Machinery, New York, NY, USA, 455–463. https://doi.org/10.1145/2902251.2902274

[9] C. L. Canonne. 2020. *A Survey on Distribution Testing: Your Data is Big. But is it Blue?* Number 9 in Graduate Surveys. Theory of Computing Library. 1–100 pages. https://doi.org/10.4086/toc.gs.2020.009

[10] C. L. Canonne, I. Diakonikolas, T. Gouleakis, and R. Rubinfeld. 2018. Testing Shape Restrictions of Discrete Distributions. *Theory Comput. Syst.* 62, 1 (2018), 4–62. Invited issue for STACS'16.

[11] S. Chan, I. Diakonikolas, R. Servedio, and X. Sun. 2014. Near-Optimal Density Estimation in Near-Linear Time Using Variable-Width Histograms. In *NIPS*. 1844–1852.

[12] S. Chaudhuri, R. Motwani, and V. R. Narasayya. 1998. Random Sampling for Histogram Construction: How much is enough?. In *SIGMOD Conference*. 436–447.

[13] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine. 2012. Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches. *Found. Trends databases* 4 (2012), 1–294.

[14] L. Devroye and G. Lugosi. 2004. Bin width selection in multivariate histograms by the combinatorial method. *Test* 13, 1 (2004), 129–145.

[15] I. Diakonikolas and D. M. Kane. 2016. A new approach for testing properties of discrete distributions. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 685–694.

[16] I. Diakonikolas, D. M. Kane, and V. Nikishkin. 2015. Optimal Algorithms and Lower Bounds for Testing Closeness of Structured Distributions. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*. 1183–1202.

[17] I. Diakonikolas, D. M. Kane, and V. Nikishkin. 2015. Testing Identity of Structured Distributions. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*. 1841–1854.

[18] I. Diakonikolas, D. M. Kane, and V. Nikishkin. 2017. Near-Optimal Closeness Testing of Discrete Histogram Distributions. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*. 8:1–8:15.

[19] I. Diakonikolas, J. Li, and L. Schmidt. 2018. Fast and Sample Near-Optimal Algorithms for Learning Multidimensional Histograms. In *Conference On Learning Theory, COLT 2018 (Proceedings of Machine Learning Research)*, Vol. 75. PMLR, 819–842. http://proceedings.mlr.press/v75/diakonikolas18a.html

[20] D. Freedman and P. Diaconis. 1981. On the histogram as a density estimator:L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 57, 4 (1981), 453–476.

[21] P. B. Gibbons, Y. Matias, and V. Poosala. 1997. Fast Incremental Maintenance of Approximate Histograms. In *VLDB*. 466–475.

[22] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss. 2002. Fast, small-space algorithms for approximate histogram maintenance. In *STOC*. 389–398.

[23] S. Guha, N. Koudas, and K. Shim. 2006. Approximation and streaming algorithms for histogram construction problems. *ACM Trans. Database Syst.* 31, 1 (2006), 396–438.

[24] Y. Han. 2019. Mixture vs. Mixture and Moment Matching. https://theinformaticists.com/2019/08/28/lecture-7-mixture-vs-mixture-and-moment-matching/

[25] P. Indyk, R. Levi, and R. Rubinfeld. 2012. Approximating and Testing k-Histogram Distributions in Sub-linear Time. In *PODS*. 15–22.

[26] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. C. Sevcik, and T. Suel. 1998. Optimal Histograms with Quality Guarantees. In *VLDB*. 275–286.

[27] J. Klemela. 2009. Multivariate histograms with data-dependent partitions. *Statistica Sinica* 19, 1 (2009), 159–176.

[28] R. P. Kooi. 1980. *The Optimization of Queries in Relational Databases*. Ph.D. Dissertation.

[29] G. Lugosi and A. Nobel. 1996. Consistency of data-driven histogram methods for density estimation and classification. *Ann. Statist.* 24, 2 (04 1996), 687–706.

[30] L. Paninski. 2008. A Coincidence-Based Test for Uniformity Given Very Sparsely Sampled Discrete Data. *IEEE Transactions on Information Theory* 54, 10 (2008), 4750–4755.

[31] R. Rubinfeld. 2006. Sublinear time algorithms. (2006). Proceedings of the International Congress of Mathematicians (ICM).

[32] D. W. Scott. 1979. On optimal and data-based histograms. *Biometrika* 66, 3 (1979), 605–610.

[33] N. Thaper, S. Guha, P. Indyk, and N. Koudas. 2002. Dynamic multidimensional histograms. In *SIGMOD Conference*. 428–439.

[34] Y. Wu, P. Yang, et al. 2019. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *Annals of Statistics* 47, 2 (2019), 857–883.

A OMITTED PROOFS FROM SECTIONS 2 AND 3

We provide in this appendix the proofs of some technical lemmas, which were omitted from the main paper due to space constraints.

Lemma A.1 (Lemma 2.1, restated). *There exists an algorithm* ***Approx-Part*** *that, given parameters $\delta \in (0,1]$ and integer $B > 1$, takes $O(B\log(B/\delta))$ samples from a distribution $\mathbf{p}$ on $[n]$ and, with probability at least $1-\delta$, outputs a partition of $[n]$ into $K \le 8B$ intervals $I_1, I_2, \ldots, I_K$ such that the following holds:*

(1) For each $i \in [n]$ such that $\mathbf{p}(i) > 1/B$, there exists $\ell \in [K]$ such that $I_\ell = \{i\}$. (All "heavy" elements are left as singletons)

(2) Every other interval is such that $\mathbf{p}(I) \le 16/B$.

Proof of Lemma 2.1. Let $B > 1$, and consider an arbitrary (unknown) probability distribution $\mathbf{p}$ over $[n]$. Let $\hat{\mathbf{p}}$ be the empirical distribution obtained by taking $m = \left\lceil 18B \ln \frac{12B}{\delta} \right\rceil = O(B\log(B/\delta))$ i.i.d. samples from $\mathbf{p}$, where $C > 0$ is an absolute constant to be chosen later in the proof.

Denote by $i_1 < \cdots < i_T$ the elements i such that $\mathbf{p}(i) > 1/B$; note that $0 \le T \le B$ since there can be at most B elements with probability mass greater than $1/B$.

Consider the following (deterministic, and unknown to the algorithm) partition of the domain into consecutive intervals:

- each of $i_1 < \cdots < i_T$ is a singleton interval $J_{\ell_j} = \{i_j\}$;
- setting for convenience $i_0 = -1$ and $i_{T+1} = n+1$, define the remaining intervals greedily as follows. For each $1 \le j \le T$, to partition $\{i_j + 1, i_j + 2, \ldots, i_{j+1} - 1\}$, do

(1) set $J \leftarrow \emptyset$, $t \leftarrow i_j + 1$

(2) while $t < i_{j+1}$

  - if $\mathbf{p}(J \cup \{t\}) < 3/(2B)$, set $J \leftarrow J \cup \{t\}$
  - else add J as an interval to the partition, then start a new interval: $J \leftarrow \emptyset$
  - consider the next element: $t \leftarrow t+1$

That is, every remaining interval ("in-between" two heavy elements i_j, i_{j+1}) is greedily divided from left to right into maximal subintervals of probability mass at most $3/(2B)$. Note that since every element $i \in \{i_j+1, \ldots, i_{j+1}-1\}$ has mass at most $1/B$, this is indeed possible and leads to a partition of $\{i_j+1, \ldots, i_{j+1}-1\}$ in intervals, where all but at most one (the rightmost one) has probability mass $\mathbf{p}(J) \in [1/(2B), 3/(2B)]$.

Overall, this process (which the algorithm cannot directly run, not knowing $\mathbf{p}$) guarantees the existence of a partition $J_1, \ldots J_S$ of the domain in at most

$$S \le T + T + 2B \le 4B$$

consecutive, disjoint intervals (the T singleton intervals, the at most T "rightmost, in-between" intervals which may have probability

mass at most $1/(2B)$, and the remaining "in-between" intervals which all have mass at least $1/(2B)$, and of which there can thus be at most $2B$ in total).

Now, consider the following partition $\hat{J}_1, \dots \hat{J}_K$ defined by the algorithm from $\hat{\mathbf{p}}$:

- every element $i \in [n]$ such that $\hat{\mathbf{p}}(i) > 1/(2B)$ is added as a singleton interval; let their number be T';
- then the algorithm proceeds as in the above process (which defined $J_1, \dots J_S$), but using $\hat{\mathbf{p}}$ instead of $\mathbf{p}$.

The same analysis (but with $T' \le 2B$ instead of $T \le B$, since the threshold for "singletons" is now $1/(2B)$ shows that this will result in a partition of the domain into K consecutive disjoint intervals $\hat{J}_1, \dots \hat{J}_K$, with

$$K \le 8B\,.$$

It remains to argue about the properties of $\hat{J}_1, \dots \hat{J}_K$, by using those of (the unknown) $J_1, \dots J_S$. First, we have that, for every $1 \le j \le T$, by a Chernoff bound,

$$\Pr\left[\,\hat{\mathbf{p}}(i_j) \le \frac{1}{2B}\,\right] \le e^{-(1/2)^2 \frac{m}{2B}} \le \frac{\delta}{3B}$$

from our setting our m. Now, for any interval J_j (where $1 \le j \le S$) such that $1/(2B) \le \mathbf{p}(J_j) \le 3/(2B)$ (there are at most $2B$ of them), we also have, all by a Chernoff bound,

$$\Pr\left[\,\hat{\mathbf{p}}(i_j) \notin \left[\frac{1}{4B}, \frac{2}{B}\right]\,\right] \le 2e^{-\frac{m}{18B}} \le \frac{\delta}{6B}$$

while for any J_j such that $\mathbf{p}(J_j) < 1/(2B)$ (there are at most T of them)

$$\Pr\left[\,\hat{\mathbf{p}}(i_j) \ge 1/B\,\right] \le e^{-\frac{m}{6B}} \le \frac{\delta}{3B}\,.$$

This means, by a union bound, that with probability at least $1 - \left(T \cdot \frac{\delta}{3B} + 2B \cdot \frac{\delta}{6B} + T \cdot \frac{\delta}{3B}\right) \ge 1 - \delta$:

- all of the $(1/B)$-heavy elements for $\mathbf{p}$ will also be $(1/(2B))$-heavy elements for $\hat{\mathbf{p}}$, and thus constitute their own singleton interval, as desired;
- all of the intervals J_j such that $1/(2B) \le \mathbf{p}(J_j) \le 3/(2B)$ (call those "heavy") satisfy $1/(4B) \le \hat{p}(J_j) \le 2/B$;
- all of the intervals J_j such that $\mathbf{p}(J_j) < 1/(2B)$ (call those "light") satisfy $\hat{p}(J_j) \le 2/B$.

Conditioning on this happening, we can analyze the properties of $\hat{J}_1, \dots \hat{J}_K$. Specifically, fix any $\hat{J}_j$ which is not a singleton of the form $\{i_\ell\}$ for one of the T $(1/B)$-heavy elements for $\mathbf{p}$. We want to argue that $\mathbf{p}(\hat{J}_j) \le 16/B$.

- If $\hat{J}_j$ intersects with at most 8 $J_{j'}$'s (which must then all be consecutive, and at least 7 are "heavy", and the rightmost one is either heavy or light), we are done, since then $\mathbf{p}(\hat{J}_j) \le 8 \cdot 2/B = 16/B$.
- Moreover, $\hat{J}_j$ cannot intersect with more than 8 $J_{j'}$'s, as otherwise those are consecutive, and thus $\hat{J}_j$ must contain at least 6 "heavy" $J_{j'}$'s. But in that case $\hat{\mathbf{p}}(\hat{J}_j) \ge 6 \cdot 1/(4B) = 3/(2B)$, while by our greedy construction we ensured that $\hat{\mathbf{p}}(\hat{J}_j) < 3/(2B)$.

This concludes the proof. □

Corollary A.2 (Corollary 2.5, restated). *Fix an interval j, we can re-compute the statistics Z_j for $O(\log(1/\delta))$ many iterations with independent samples and take the median over all iterations. The median $\bar{Z}_j$ then satisfies the following properties*

- ***Markov-Based*** *It holds that $\bar{Z}_j \le 3\,\mathbf{E}[Z_j]$ with probability at least $1 - \delta$.*
- ***Chebyshev-Based*** *If $\mathbf{E}[Z_j] \ge \frac{1}{5} m\varepsilon^2$, where m is the number of samples used to compute Z_j, it holds that $\left|\bar{Z}_i - \mathbf{E}[Z_i]\right| \le \frac{1}{5} m\varepsilon^2$ with probability at least $1 - \delta$.*

Lastly, given a set of intervals $\mathcal{I} = \{I_1, I_2, \cdots\}$, the statistic $\bar{Z}$ on $\mathcal{I}$ is $\bar{Z} := \sum_{I_j \in \mathcal{I}} \bar{Z}_j$.

Proof of Corollary 2.5. We first show the Markov Based concentration result. Suppose we compute the test statistics Z_j for T times, each with $m \ge 20000\sqrt{n}/\varepsilon^2$ independent samples. Denote the results as $Z_j^{(1)}, \cdots, Z_j^{(T)}$. By Markov Inequality, we have

$$\Pr\left[Z_j^{(t)} > 3\,\mathbf{E}[Z_j]\right] < \frac{1}{3}.$$

Notice that if we take $\bar{Z}_j = \text{Median}(Z_j^{(1)}, \cdots, Z_j^{(t)})$, $\{\bar{Z}_j > 3\,\mathbf{E}[Z_j]\}$ happens only when at least half of the statistics exceed $\mathbf{E}[Z_j]$. By applying the tail bounds for binomial distribution, we have

$$\Pr\left[\bar{Z}_j > 3\,\mathbf{E}[Z_j]\right] \le \exp\left(-2T\left(\frac{2}{3} - \frac{T/2}{T}\right)\right) = \exp\left(-\frac{T}{3}\right),$$

which is bounded by δ as long as $T \ge 3 \cdot \log(1/\delta)$.

For the Chebyshev based result, we combine Proposition 2.4 with Chebyshev's Inequality, which gives

$$\Pr\left[\left|Z_j^{(t)} - \mathbf{E}[Z_j]\right| > \frac{1}{5}\,\mathbf{E}[Z_j]\right] \le \Pr\left[\left|Z_j^{(t)} - \mathbf{E}[Z_j]\right| > 2 \cdot \sigma(Z_j)\right] < \frac{1}{4}.$$

The rest of the argument is similar to the Markov based result. □

Lemma A.3 (Lemma 2.6, restated). *Assume Algorithm 2 is given a partition $\mathcal{I}$ satisfying $|\mathcal{I}| \ge 100k \log k$ and outputs the set of intervals $\mathcal{Q}$. Moreover, denote $\mathcal{G} = \mathcal{I} \setminus \mathcal{Q}$.*

- *Suppose $\mathbf{p} \in \mathcal{H}_k^n$. Then the algorithm returns $\hat{\mathbf{p}}$ and $\mathcal{Q}$ such that $\widetilde{\chi}^2_{Z,\mathcal{I}\setminus\mathcal{Q}}\left(\mathbf{p} \,\middle\|\, \hat{\mathbf{p}}\right) \le \varepsilon$ with probability at least $1 - \delta$.*
- *Suppose $\mathrm{TV}^{\mathcal{I}\setminus\mathcal{Q}}(\mathbf{p}, H) > 40\varepsilon$ for every $H \in \mathcal{H}_k^n$ and any $\mathcal{Q}$ such that $\mathcal{Q} \subseteq \mathcal{I}$ and $|\mathcal{Q}| \le 12k \log k$. Then, the algorithm outputs Reject with probability at least $1 - \delta$.*

Lastly, the algorithm uses at most $O((\sqrt{n}/\varepsilon^2) \cdot \log k \cdot \log(k/\delta) + |\mathcal{I}| \log(1/\delta)/\varepsilon^2)$ samples.

Proof of Lemma 2.6. We first argue about the completeness of the test, that is, its behavior when $\mathbf{p} \in \mathcal{H}_k^n$; Then, we turn to the soundness case, i.e., its probability of outputting Reject when $\mathbf{p}$ is "far" from all k-histograms, in the sense of the second item of the lemma's statement.

Completeness. Assume $p \in \mathcal{H}_k^n$, we argue the probability that the algorithm outputs Reject at a fixed iteration is small. Since $\mathbf{p} \in \mathcal{H}_k^n$, by Lemma 2.3, it holds that $\chi^2_{\mathcal{I}\setminus\mathcal{B}}\left(\mathbf{p} \,\middle\|\, \hat{\mathbf{p}}\right) \le \varepsilon^2$ with probability at least $1 - \frac{\delta}{100}$. Then, by Proposition 2.4, we have $\sum_{i \in \mathcal{I}\setminus\mathcal{B}} \mathbf{E}[Z_i] \le m\varepsilon^2$. By Corollary 2.5, it holds for a fixed iteration $\sum_{i \in \mathcal{I}\setminus\mathcal{B}} \bar{Z}_i \le 3m\varepsilon^2$ with probability at least $1 - \frac{\delta}{100k \cdot \log k}$ if we apply the median

trick over $M := \Theta(\log(1/\delta) + \log k)$ repetitions. Conditioned on that, i.e., the statistics behave well on the non break-point intervals, the algorithm will never output Reject in line 8 as the algorithm can always choose to remove all the break-point intervals.

Next, we show that the quantity $\sum_{i\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_i]$ becomes small as the algorithm exits from the filtering stage – regardless of whether it runs for the full $12k\log k$ iterations or exits early (if $\sum_{i\in\mathcal{I}_{\text{rem}}} Z_i$ is already small in some iteration).

First, we examine the case when the algorithm runs for the full $12\log k$ iterations. Among the break-point intervals, there are some especially "heavy" ones - those whose statistics satisfy $\mathbf{E}[Z_j] \geq 100m\varepsilon^2$. We argue all these "heavy bad" intervals will be removed after the first iteration with high probability. Recall that we apply the median trick for $M := \Theta(\log(1/\delta) + \log k)$ many repetitions. Then, by Corollary 2.5 (Chebyshev Based item), for a fixed "heavy" break-point intervals j, it holds that $\bar{Z}_j \geq 99m\varepsilon^2$ with probability at least $1 - \frac{\delta}{100\cdot k\cdot\log k}$. By a union bound, all these intervals will be detected and removed in the first iteration with probability at least $1 - \frac{\delta}{100\cdot\log k}$.

Conditioning on that, we have $\sum_{j\in\mathcal{B}\cap\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_j] \leq 100km\varepsilon^2$ and $\sum_{j\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_j] \leq (100k+1)m\varepsilon^2$ from the second iteration. Denote the set of intervals removed at the current iteration by $\mathcal{R}$. By definition of the algorithm, we have $\sum_{i\in\mathcal{R}} \bar{Z}_i \geq \alpha = m\varepsilon^2$. By applying Corollary 2.5 (Markov Based item) on each individual interval and taking a union bound over all $O(k\log k)$ intervals in $\mathcal{I}$, we have

$$\sum_{i\in\mathcal{R}} \mathbf{E}[Z_i] \geq \frac{1}{3}\sum_{i\in\mathcal{R}} \bar{Z}_i \tag{11}$$

with probability at least $1 - \frac{\delta}{100\log k}$. On the other hand, we always have

$$\sum_{i\in\mathcal{R}} \bar{Z}_i \geq \sum_{i\in\mathcal{I}_{rem}} \bar{Z}_i - 3\alpha. \tag{12}$$

Applying Corollary 2.5 (Chebyshev Based item) on the sum of the statistics in the remaining intervals before current filtering (namely, we still have $\mathcal{R} \in \mathcal{I}_{rem}$) as long as $\sum_{i\in\mathcal{I}_{rem}} \mathbf{E}[Z_i] \geq 10m\varepsilon^2$, it holds

$$\sum_{i\in\mathcal{I}_{rem}} \bar{Z}_i \geq \frac{9}{10}\sum_{i\in\mathcal{I}_{rem}} \mathbf{E}[Z_i] \tag{13}$$

with probability at least $1-\frac{\delta}{100\log k}$. Combining Equations (11), (12), (13), we then have, as long as $\sum_{i\in\mathcal{I}_{rem}} \mathbf{E}[Z_i] \geq 10m\varepsilon^2$,

$$\sum_{i\in\mathcal{R}} \mathbf{E}[Z_i] \geq \frac{1}{5}\sum_{i\in\mathcal{I}_{rem}} \mathbf{E}[Z_i] \tag{14}$$

with probability at least $1 - \frac{\delta}{50\log k}$. In other words, at least a $1/5$ portion of $\sum_{i\in\mathcal{I}_{rem}} \mathbf{E}[Z_i]$ is removed in every iteration. Thus, after $\log(101k)/\log(5/4) \leq 12\log k$ iterations, we have $\sum_{i\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_i] \leq 10m\varepsilon^2$.

If the algorithm exits the filtering stage early due to $Z < 4\alpha = 4m\varepsilon^2$ in some iteration, we argue $\sum_{i\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_i]$ must also be small with high probability. For the sake of contradiction, suppose $\sum_{i\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_i] \geq 5m\varepsilon^2$. Then by Corollary 2.5 (Chebyshev based item), we have $\sum_{i\in\mathcal{I}_{\text{rem}}} Z_i \geq \frac{9}{10}\sum_{i\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_i] \geq 4.5m\varepsilon^2$ with probability at least $1 - \frac{\delta}{100\log k}$. Thus, conditioning on that the filtering stage exits early, we must have $\sum_{i\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_i] \leq 5m\varepsilon^2$ with probability at least $1 - \frac{\delta}{100\log k}$.

Therefore, regardless of whether the algorithm exits early from the loop, we have $\sum_{i\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_i] \leq 10m\varepsilon^2$ conditioning on that the algorithm reaches line 15 with high probability (we will specify the fail probability later). This shows that the algorithm succeeds in find the desired $\hat{p}$ and Q. It remains to show that the algorithm will not reject on line 15. By Corollary 2.5 (Markov Based item), $\sum_{i\in\mathcal{I}_{\text{rem}}} Z_i \leq 3\cdot\sum_{i\in\mathcal{I}_{\text{rem}}} \mathbf{E}[Z_i] \leq 30m\varepsilon^2$ with probability at least $1 - \frac{\delta}{100}$. Hence, the testing stage on line 15 will pass with high probability. Hence, the algorithm gives the desired result in the completeness case.

We now revisit all cases where the algorithm may fail. First, the learning stage may fail with probability at most $\delta/100$. In each filtering iteration, the statistics may misbehave with probability at most $\delta/(50\log k)$. In the final testing stage, the statistics may misbehave with probability at most $\frac{\delta}{100}$. Overall, the algorithm succeeds with probability at least $(1-\frac{\delta}{100})\cdot(1-\frac{\delta}{50\log k})^{12\log k}\cdot(1-\frac{\delta}{100}) \geq 1-\delta$.

Soundness. For the sake of contradiciton, assume the algorithm reaches and passes the last testing stage: this implies that $\text{TV}^{\mathcal{I}\setminus Q}(\mathbf{p},\hat{\mathbf{p}}) \leq \varepsilon$ where $|Q| \leq 12k\log k$ with probability at least $1-\delta$. However, since we passed line 3, we have $\text{TV}(\mathbf{p}^*,\hat{\mathbf{p}}) \leq \varepsilon$ where $\mathbf{p}^* \in \mathcal{H}_k^n$. Combining the two, we get a contradiction, and thus the algorithm must output Reject in some stage. □

Lemma A.4 (Lemma 3.3, restated). *Let $V := \frac{m}{k}(1+\varepsilon U)$, $V' := \frac{m}{k}(1+\varepsilon U')$, where U, U' are the random variables solution of the Linear Program from Section 3. Then, if $\frac{k}{\varepsilon^2}\cdot\frac{L}{\lambda^2} > m$, we have*

$$\text{TV}\left(\mathbf{E}[\text{Poi}(V)], \mathbf{E}\left[\text{Poi}\left(V'\right)\right]\right) \leq e^{-L/4}.$$

Proof of Lemma 3.3. We will use Theorem 4 from [24], restated below:

Theorem A.5 (Theorem 4 from [24]). *For any $\Lambda > 0$ and random variables X, X' supported on $[-\Lambda,\infty)$, we have*

$$\text{TV}\left(\mathbf{E}[\text{Poi}(\Lambda+X)], \mathbf{E}[\text{Poi}(\Lambda+X')]\right) \leq \frac{1}{2}\left(\sum_{\ell=0}^{\infty}\frac{\left|\mathbf{E}[X^\ell]-\mathbf{E}[X'^\ell]\right|^2}{\ell!\Lambda^\ell}\right)^{1/2}.$$

We apply this theorem with $X := \frac{\varepsilon m}{k}U$, $X' := \frac{\varepsilon m}{k}U'$, and $\Lambda := \frac{m}{k}$, to get

$$\begin{aligned}
&4\,\text{TV}\left(\mathbf{E}[\text{Poi}(V)], \mathbf{E}\left[\text{Poi}\left(V'\right)\right]\right)^2 \\
&\leq \sum_{\ell=0}^{\infty}\left(\frac{\varepsilon m}{k}\right)^{2\ell}\frac{\left|\mathbf{E}\left[U^\ell\right]-\mathbf{E}\left[U'^\ell\right]\right|^2}{\ell!(m/k)^\ell} = \sum_{\ell=L/2+1}^{\infty}\left(\frac{\varepsilon^2 m}{k}\right)^{\ell}\frac{\left|\mathbf{E}\left[U^\ell\right]-\mathbf{E}\left[U'^\ell\right]\right|^2}{\ell!} \\
&\qquad\qquad \text{(the first } L \text{ moments match)} \\
&\leq \sum_{\ell=L/2+1}^{\infty}\left(\frac{\varepsilon^2 m}{k}\right)^{\ell}\frac{4\lambda^{2\ell}}{\ell!} = 4\sum_{\ell=L/2+1}^{\infty}\left(\frac{\varepsilon^2\lambda^2 m}{k}\right)^{\ell}\frac{1}{\ell!}. \quad (|U|,|U'|\leq\lambda)
\end{aligned}$$

Setting for convenience $\kappa := \frac{\varepsilon^2\lambda^2 m}{k}$. and denoting by Y a $\mathrm{Poi}(\kappa)$ random variable, this leads to (since $L > \kappa$)

$$\mathrm{TV}\left(\mathbf{E}[\mathrm{Poi}(V)], \mathbf{E}\left[\mathrm{Poi}\left(V'\right)\right]\right)^2 \leq \sum_{\ell=L+1}^{\infty} \frac{\kappa^\ell}{\ell!} = e^\kappa \Pr[\, Y \geq L+1 \,]$$
$$\leq e^\kappa e^{-\frac{(L+1-\kappa)^2}{2(L+1)}} = e^{-\frac{1}{2}\left(L+1+\frac{\kappa^2}{L+1}\right)} \leq e^{-\frac{L}{2}},$$

where the second inequality is by standard Poisson concentration (see, e.g., the note [7]). This proves the lemma. □