Proceedings of Machine Learning Research vol 99:1–49, 2019 32nd Annual Conference on Learning Theory

Towards Testing Monotonicity of Distributions Over General Posets

Maryam Aliakbarpour MARYAMA@MIT.EDU
CSAIL, MIT

Themis Gouleakis TGOULEAK@MPI-INF.MPG.DE
Max Planck Institute for Informatics

John Peebles JPEEBLES@MIT.EDU
CSAIL, MIT

Ronitt Rubinfeld RONITT@CSAIL.MIT.EDU
CSAIL, MIT, TAU

Anak Yodpinyanee ANAK@MIT.EDU
CSAIL, MIT

Editors: Alina Beygelzimer and Daniel Hsu

Abstract

In this work, we consider the sample complexity required for testing the monotonicity of distributions over partial orders. A distribution p over a poset is *monotone* if, for any pair of domain elements x and y such that $x \preceq y$, $p(x) \leq p(y)$.

To understand the sample complexity of this problem, we introduce a new property called *bigness* over a finite domain, where the distribution is T-big if the minimum probability for any domain element is at least T. We establish a lower bound of $\Omega(n/\log n)$ for testing bigness of distributions on domains of size n. We then build on these lower bounds to give $\Omega(n/\log n)$ lower bounds for testing monotonicity over a matching poset of size n and significantly improved lower bounds over the hypercube poset.

We give sublinear sample complexity bounds for testing bigness and for testing monotonicity over the matching poset. We then give a number of tools for analyzing upper bounds on the sample complexity of the monotonicity testing problem.

The previous lower bound for testing Monotonicity of

Keywords: Property Testing; Monotone Distributions; Partially Ordered Sets;

1. Introduction

We consider the problem of testing whether a distribution is monotone: an essential property that captures many observed phenomena of real-world probability distributions. For instance, monotone distributions over *totally ordered sets* might be used to describe distributions on diseases for which the probability of being affected by the disease increases with age. More generally, an important class of distributions are characterized by being monotone over a *partially ordered set* (poset). For these distributions, if a domain element u lower bounds v in the partial ordering (denoted $u \preceq v$), then $p(u) \leq p(v)$ (whereas if u and v are unrelated in the poset, then p needs not satisfy any particular requirement on the relative probabilities of u and v). Such distributions might include distributions on diseases for which the probability of being affected increases by some combination of several risk factors. Many commonly studied distributions, e.g. exponential distributions or multivariate exponential distributions, are or can be approximated by piecewise monotone functions. As monotone distributions are a fundamental class of distributions, the problem of testing whether a distribution is monotone is a key building block for distribution testing algorithms.

Given an unknown distribution, over a poset domain, the goal is to distinguish whether the distribution is monotone or far from any monotone distribution, using as few samples as possible. This problem has been considered in the literature: the problem of testing whether a distribution is monotone was first considered in the work of Batu et al. (2004), where testing the monotonicity of distributions over totally ordered domains and partially ordered domains that corresponded to two-dimensional grids were considered. The work of Bhattacharyya et al. (2010) introduced the study of testing the monotonicity of distributions over general partially ordered domains, and in particular, considered the Boolean hypercube ($\{0,1\}^d$). Several other works considered these questions Daskalakis et al. (2013); Acharya et al. (2015); Canonne et al. (2018) under various different domains and achieved improved sample complexity bounds.

The sample complexity of the testing problem varies greatly with the structure of the poset: On the one hand, for domains of size n that are total orders, $\Theta(\sqrt{n})$ samples suffice for distinguishing monotone distributions, from those that are ϵ-far in total variation distance from any monotone distribution Batu et al. (2004); Acharya et al. (2015); Canonne et al. (2018). On the other hand, testing distributions defined over the matching poset requires nearly linear in n, specifically $\Omega(n^{1-o(1)})$, samples Bhattacharyya et al. (2010). Furthermore, for a large class of familiar posets, such as the Boolean hypercubes, little is understood about the sample complexity of the testing problem.

Our results and approaches: We first define a new property called the *bigness* property, which we use as our main building block for establishing sample complexity lower bounds for monotonicity testing. A distribution is *T-big* if every domain element is assigned probability mass at least T.

Though the bigness property is a *symmetric* property (i.e., permuting the labels of the elements does not change whether the distribution has the property or not), we use lower bounds for testing the bigness property in order to prove lower bounds on testing monotonicity, which is not a symmetric property. In addition, the bigness property is a natural property, and thus of interest in its own right.

We show that the sample complexity of the bigness testing problem is $\Theta(n/\log n)$ when $T = \Theta(1/n)$. The upper bound follows from applying the algorithm of Valiant and Valiant (2017) that learns the underlying distribution up to a permutation of the domain elements. Our lower bound approach is inspired by the framework of Wu and Yang (2016b), used to lower bound the number of samples needed to estimate support sizes. Our lower bound is established by showing that the distribution of samples, one generated from T-big distributions (p's) and the other generated from distributions that are ϵ-far from T-big (p''s), are statistically close. In contrast with the standard lower bound framework, p and p' are not picked from two sets of distributions. Instead, the distribution p (resp. p') is constructed by having each domain element i choose its probability $p(i)$, in an i.i.d. fashion, from the distribution P_V (resp. $P_{V'}$) over possible probabilities in $[0, 1]$. To design P_V and $P_{V'}$, we introduce a new optimization problem that maximizes ϵ while keeping the distribution of samples statistically close. This constraint is established via the *moments matching* technique, which allows us to show that the distributions are indistinguishable with $o(n/\log n)$ samples, but also plays a crucial role in many other settings Raskhodnikova et al. (2009); Valiant (2008); Bhattacharyya et al. (2010); Valiant and Valiant (2016, 2017); Wu and Yang (2016b,a).

By reducing from the bigness testing problem, we next give a lower bound of $\Omega(n/\log n)$ on the sample complexity of the monotonicity testing problem over the matching poset, improving on the $\Omega\left(n/2^{\Theta(\sqrt{\log n})}\right)$ lower bound in Bhattacharyya et al. (2010). In addition to improving the sample complexity lower bound, one particularly useful byproduct of our approach is that the maximum probability of an element in the constructed lower bound distribution families can be made small, which assists us in proving lower bounds for other posets in the following.

Finally, we leverage the lower bound for the monotonicity testing problem over the matching poset to prove a lower bound of $N^{1-\delta}$ for $\delta = \Theta(\sqrt{\epsilon}) + o(1)$ for monotonicity testing over the Boolean hypercube of size $N = 2^d$, greatly improving upon the standard "Birthday Paradox" lower bound of $\Omega(\sqrt{N})$. Our reduction follows from finding a large embedding of the matching poset in the hypercube, and its efficiency follows from the previously mentioned upper bound on the maximum element probability from the bigness lower bound construction above.

We then give a number of new tools for analyzing upper bounds on the sample complexity of the monotonicity testing problem:

1. We prove that the distance of a distribution to monotonicity can be characterized approximately as the weight of a *maximum weighted matching* in the *transitive closure* of the poset, where the weight of the edge (u, v) is the amount of violation from being monotone: $\max(0, p(u) - p(v))$. This characterization gives a structural result about distributions that are ϵ-far from monotone. Moreover, this results extends the work of Fischer et al. (2002) to non-boolean valued functions. The work of Fischer et al. (2002) shows that the distance of a boolean function f to monotonicity is related to the number of "violating edges" in the transitive closure of the underlying poset.

2. Via the characterization above, we show that the monotonicity testing problem over *bipartite* posets (where all edges are directed in the same direction) captures the monotonicity testing problem in its full generality. That is, we give a reduction from monotonicity testing over any poset to monotonicity testing over a bipartite poset. Our reduction preserves the number of vertices and the distance parameter up to a constant multiplicative factor. As in the previous, this result extends the work of Fischer et al. (2002) to non-boolean valued functions.

3. Leveraging the learning algorithms for *symmetric* distributions in Valiant and Valiant (2017), we propose algorithms with sample complexity $O(n/(\epsilon^2 \log n))$ for testing bigness of a distribution, and for testing monotonicity on matching posets. The proof of our latter result requires certain subtle details: (1) an additional reduction that

allows us to scale our distribution for "each side" of the matching, in order to generate sufficient samples from each side, as required by the algorithm of Valiant and Valiant (2017), and (2) technical lemmas establishing bounds between the total variation distance and the distance notion in Valiant and Valiant (2017), under the scaling mentioned earlier.

4. We give a reduction from monotonicity testing on a bipartite poset, to monotonicity testing on the matching (for which the testing algorithm is constructed above). This reduction gives an algorithm for monotonicity testing on any bipartite poset (which is the most general problem, as argued earlier), in which the overhead in the sample complexity depends only on the maximum degree of the bipartite graph.

5. We give another upper bound for testing monotonicity on bipartite posets: $O((\log M)/\epsilon^2)$ where M is the number of "endpoint sets" of all possible matchings contained in the given bipartite graph (or equivalently, the number of induced subgraphs that admit a perfect matching over their respective vertex sets). Note that for the matching poset, $M = 2^n$ yields an $O(n/\epsilon^2)$ upper bound, and therefore for matching posets our previous algorithm is preferable. However, this bound yields an upper bound of $O(n/\epsilon^2)$ for *all* posets, and could potentially be even smaller for certain classes of graphs, such as collections of large stars.

6. Finally, we give an upper bound of $O(\frac{n^{2/3}}{\epsilon} + \frac{1}{\epsilon^2})$ samples for monotonicity testing on bipartite posets, under the guarantee that the distribution being tested is a uniform distribution on some subset of known size of the domain. This special case is of interest in that it relates to the well studied problem of testing monotonicity of *Boolean functions*, in a somewhat different setting where instead of getting query access to the function, we are given uniform "positive" samples of domain elements x for which $f(x) = 1$.

Other related work Batu, Kumar, and Rubinfeld Batu et al. (2004) initiated the study of testing monotonicity of distributions. For the case where the domain is totally ordered, the sample complexity is known to be $\Theta(\sqrt{n})$ Batu et al. (2004); Acharya et al. (2015); Canonne et al. (2018). Several works have considered distributions over higher dimensional domains. In Batu et al. (2004); Bhattacharyya et al. (2010), it is shown that testing monotonicity of a distribution on the two dimensional grid $[m] \times [m]$ (here $N = m^2$) can be performed using $\widetilde{O}(N^{3/4})$ samples. For higher dimensional grids $[m]^d$ (where $N = m^d$), Bhattacharyya et al. provided an algorithm that uses $\widetilde{O}(m^{d-1/2}) = \widetilde{O}(N/\sqrt[2d]{N})$ samples Bhattacharyya et al. (2010). Acharya et al. gave an upper bound of $O(\frac{\sqrt{N}}{\epsilon^2} + (\frac{d \log m}{\epsilon^2})^d \cdot \frac{1}{\epsilon^2})$ and a lower bound of $\Omega(\sqrt{N}/\epsilon^2)$ Acharya et al. (2015). While their result gives a tight bound of $\Theta(\sqrt{N}/\epsilon^2)$ when d is relatively small compared to m, it does not yield a tester for Boolean hypercubes using a sublinear number of samples.

Bhattacharyya et al. considered the problem of monotonicity testing over general posets Bhattacharyya et al. (2010). In particular, they proposed an algorithm for testing the monotonicity of distributions over hypercubes (where $N = 2^d$) using $\tilde{O}(N/(\log N/\log\log N)^{1/4})$ samples. They provide a lower bound of $\Omega(n^{1-o(1)})$ for testing monotonicity of distributions over a matching of size n, and a lower bound of $\Omega(\sqrt{n})$ when the poset contains a linear-sized matching in the transitive closure of its Hasse digraph.

In addition to the above, testing monotonicity of distributions has been considered in various settings Adamaszek et al. (2010); Daskalakis et al. (2012); Canonne (2015). There are several works on testing various properties, e.g. uniformity, closeness, and independence when the underlying distribution is monotone Batu et al. (2005, 2004); Rubinfeld and Servedio (2005); Daskalakis et al. (2013); Acharya et al. (2013).

Testing monotonicity of *boolean functions* is also well studied (e.g., Goldreich et al. (1998); Dodis et al. (1999); Lehman and Ron (2001); Fischer et al. (2002); Chakrabarty and Seshadhri (2013, 2014); Belovs and Blais (2016); Black et al. (2018)). In the general regime, the algorithm can query the value of the function at any element in the poset. This ability is in sharp contrast with our model, in which the algorithm only receive samples according to the distribution, which do not directly reveal the probability of the elements. It is known that one can test monotonicity of functions over hypergrids, and hypercubes using as few as polylogarithmic queries in the size of the domain. This query complexity is exponentially smaller than the sample complexity of testing monotonicity of distributions, demonstrating that there are inherent differences between the two problems.

2. Preliminaries

We use $[n]$ to indicate the set $\{1, 2, \ldots, n\}$. Throughout this paper we use the total variation distance denoted by d_{TV} unless otherwise stated. We also denote the ℓ_1-distance by d_{ℓ_1}. For a distribution p, we denote the probability of the domain element x by $p(x)$. Given a multiset of samples from a distribution on $[n]$, the *histogram* of the samples is an n-dimensional vector, $h = (h_1, h_2, \ldots, h_n)$, where h_i is the frequency of the i-th element in the sample set.

A poset $G = ([n], E)$ is called a *line* if and only if E contains all the edges $(i, i+1)$ for $1 \leq i \leq n$. We say a poset is a *matching* if all of the edges in the poset are vertex-disjoint. We say a poset $G = (V, E)$ is an *n-dimensional hypercube* when V is $\{0,1\}^n$ and E contains all edges (u, v) where there exists a coordinate i such that $u_i = 0$ and $v_i = 1$ and $u_j = v_j$ for all $i \neq j$.

Monotonicity. A partially-ordered set (poset) is described as a directed graph $G = (V, E)$, where each edge (u, v) indicates the relationship $u \preceq v$ on the poset. A *matching poset* is a poset where the underlying graph G is a matching. A distribution p over a poset domain $V = \{v_1, \ldots, v_n\}$ is a distribution over the vertex set V. A distribution p is *monotone* (with respect to a poset G) if for every edge $(u, v) \in E$ (i.e., every ordered pair $u \preceq v$), $p(u) \leq p(v)$. Let $\mathsf{Mon}(G)$ be the set of all monotone distributions over the poset G. We say that p is *ϵ-far from monotone* if its *distance to monotonicity*, $d_{TV}(p, \mathsf{Mon}(G)) := \min_{q \in \mathsf{Mon}(G)} d_{TV}(p, q)$, is at least ϵ.

Definition 1 *Let p be a distribution on poset G and ϵ be the proximity parameter. Suppose an algorithm, $\mathcal{A}$, has sample access to p and the full description of poset G. $\mathcal{A}$ is called a* monotonicty tester for distributions *if the following is true with probability at least 2/3 when the tester has sample access to the distribution.*

- *If p is monotone, then $\mathcal{A}$ outputs* accept.
- *If p is ϵ-far from monotone, then $\mathcal{A}$ outputs* reject.

Bigness. A probability distribution p over a domain $[n] = \{1, \ldots, n\}$ is T*-big* if, for every domain element $i \in [n]$, $p(i) \geq T$. Related notions for distance to T-bigness are defined analogously. The parameter T is called the *bigness threshold*, and may be omitted if it is clear from the context. Let $\mathsf{Big}(n, T)$ indicate the set of all distributions over $[n]$ that are T-big. We define the distance to T-bigness as $d_{TV}\left(p, \mathsf{Big}(n, T)\right) = \min_{q \in \mathsf{Big}(n,T)} d_{TV}(p, q)$. If this distance is at least ϵ, we say the distribution is ϵ-far from being T-big.

Definition 2 *Let p be a distribution on $[n]$. Suppose Algorithm $\mathcal{A}$ receives threshold T and bigness parameter ϵ, and has sample access to p. $\mathcal{A}$ is a* T-bigness tester *if the following is true with probability at least* $2/3$*.*

- *If p is T-big, then $\mathcal{A}$ outputs* accept*.*
- *If p is ϵ-far from T-big, then $\mathcal{A}$ outputs* reject*.*

Also, T-bigness testing problem *refers to the task of distinguishing the above cases with high probability.*

Remark 1 *Note that the probability $2/3$ is arbitrary in the above definitions. One can amplify the probability of outputting the correct answer to $1 - \delta$ by increasing the number of samples by an $O(\log 1/\delta)$ factor.*

3. Overview of Our Techniques

In this section, we give an overview of our results and the high-level idea of our techniques.

3.1. A lower bound for the bigness testing problem

In Section A, we provide two random processes for generating histograms of samples from two families of distributions, such that one family consists of "big" distributions, and the other family largely of "ϵ-far from big" distributions. Then, we show that unless a large number of samples have been drawn, the distributions over the histograms generated via these two random processes are statistically very close to each other, and hence appear indistinguishable to any algorithm, as specified precisely in Theorem 3. The construction yields a lower bound for the general problem of testing the bigness property in Corollary 4. Furthermore, the construction provides a useful building block for establishing further lower bounds for monotonicity testing in various scenarios in Section C.

To generate histograms from the two families of distributions, imagine the following process: We have two prior distributions P_V and $P_{V'}$, and we generate probability vectors (measures), p and p', according to the priors: Each domain element i randomly picks its probability in an i.i.d fashion from the prior distribution. More precisely, let $V_1, V_2, \ldots, V_n$ be n i.i.d. random variables from prior P_V, then p is defined to be the following:

$$p = \frac{1}{n}(V_1, V_2, \ldots, V_n)\,.$$

We generate p' similarly according to prior $P_{V'}$. While the total probability is unlikely to sum to 1, we will design the priors, P_V and $P_{V'}$, so that we can later modify p or p' into a probability distribution with only small changes. We then generate histograms of samples from (the normalization of) p by drawing n independent random variables $h_i \sim \mathbf{Poi}(s \cdot p(i))$ (namely $h_i \sim \mathbf{Poi}(sV_i/n)$) for $i = 1, \ldots, n$, and output $h = (h_1, \ldots, h_n)$ as the histogram of the samples. Note that by Poissonization method, one may view the histogram as being generated from a set of $\mathbf{Poi}(s \cdot \sum_i V_i/n)$ samples from the normalization of p. Hence, if $\sum_i V_i/n$ is close to one, the histogram serves as a set of roughly s samples. We set s more specifically in terms of the rest of the parameters later.

The goal in Section A is to find two prior distributions P_V and $P_{V'}$, then generate two probability vectors p and p', and two histograms h and h' according to them respectively, such that the following events hold with high probability.

1. The probability vectors p and p' are approximate probability distributions; that is, their total probability masses are each close to 1.

2. After scaling the probability vectors p and p' above into respective probability distributions, the normalization of p is T-big, and the normalization of p' is ϵ-far from any T-big distribution.

3. The total numbers of (Poissonized) samples in h and h' drawn from the normalization of p and p' are each $\Omega(s)$, where s is the sample complexity lower bound we are aiming to prove.

4. Given h or h', distinguishing whether it is generated from P_V or $P_{V'}$ with success probability $2/3$ requires h or h' to contain at least s samples.

5. Additionally, we will bound the largest probability mass $p_{\max}$ that the normalized distributions place on any domain element – this part is not necessary for this section, but will be useful for the reduction between monotonicity testing and bigness testing later on.

Now, if we choose P_V and $P_{V'}$ carefully such that h and h' are generated according to the above process based on P_V and $P_{V'}$ are hard to distinguish, then we can establish a lower bound for the bigness testing problem. We state this result more formally as the following theorem in Section A.

Theorem 3 *For integer $L = O(\log n)$ and sufficiently small $\epsilon = \Omega(1/n)$, there exist a parameter $\beta = \beta(L, \epsilon)$ and two distributions $\mathcal{H}^+$ and $\mathcal{H}^-$ over the set of possible histograms of size at least $s = \Omega\left(n^{1-1/L} \log^2(1/\epsilon)/L\right)$ with the following properties:*

- *The histogram generated from $\mathcal{H}^+$ is drawn from a $1/(\beta n)$-big distribution.*

- *The histogram generated from $\mathcal{H}^-$ is drawn from a distribution which is ϵ-far from any $1/(\beta n)$-big distribution.*

- $d_{TV}\left(\mathcal{H}^+, \mathcal{H}^-\right) \le 0.01$.

- *The largest probability mass among any elements in any probability distributions above (from which the histograms are drawn) is $p_{max} = O(L^2/(n \log^2(1/\epsilon)))$.*

An important case of this theorem is when $L = \Theta(\log n)$, where we establish a nearly linear sample complexity lower bound of $\Omega(n/\log n)$ for the general problem of bigness testing as follows.

Corollary 2 *For sufficiently small parameter $\epsilon = \Omega(1/n)$, there exists a parameter $\beta = \beta(\epsilon)$ such that any algorithm that can distinguish whether a distribution over $[n]$ is $1/(\beta n)$-big or ϵ-far from any $1/(\beta n)$-big distribution with probability $2/3$ requires $\Omega(n \log^2(1/\epsilon)/\log n)$ samples. In particular when ϵ is a constant, β is constant, then any such algorithm requires $\Omega(n/\log n)$ samples.*

We propose the following optimization problem, **OP1**, such that its optimal solution specifies P_V and $P_{V'}$, satisfying the requirements of the theorem. Intuitively speaking, as P_V aims to generate T-big distributions, we must ensure that V_i's are bounded away from $1/\beta$, so that $p(i) = V_i/n$ has expected value higher than $T = 1/(\beta n)$. At the same time, we hope to maximize the probability that $V'_i = 0$ so that p' has lots of domain elements with probability zero to make its normalization far from any T-big distribution. In addition, we find P_V and $P_{V'}$ under the constraint that the first L moments of them are exactly matched, as to ensure that the resulting distributions over the histograms, $\mathcal{H}$ and $\mathcal{H}'$, are statistically close. The objective value of this optimization problem corresponds to the expected distance of p' to the closest T-big distribution in the ℓ_1-distance.

$$\begin{aligned}
\text{Definition of } \mathbf{OP1}: \quad \sup \quad & \tfrac{1}{\beta}\Pr[V' = 0] \\
s.t. \quad & \mathbf{E}[V] = \mathbf{E}[V'] = 1 \\
& \mathbf{E}[V^j] = \mathbf{E}[V'^j] \quad \text{for } j = 1, 2, \ldots, L \\
& V \in \left[\tfrac{1+\nu}{\beta}, \tfrac{\lambda}{\beta}\right], V' \in \{0\} \cup \left[\tfrac{1+\nu}{\beta}, \tfrac{\lambda}{\beta}\right] \text{ and } \beta > 0.
\end{aligned}$$

In the above optimization problem, the unknowns are P_V, P'_V, and β. ν and λ are two parameters specified latter in the proof. That is we are looking for two distributions P_V and P'_V such that two random variables V and V' drawn from them respectively have expected value one, and their first L moments are matched. Also, β controls the range of the probabilities, $p(i)$'s and $p'(i)$'s, and the distance to the bigness property.

We relate the optimal solution for **OP1** to an LP defined by Wu and Yang (2016b), who in turn relate their LP to the error from the best polynomial approximation of the function $1/x$ over the interval $[1 + \nu, \lambda]$. By doing this, we show the existence of a solution $(P_V, P_{V'})$ where the value $\Pr[V' = 0]$, which is proportional to the distance to $1/(\beta n)$-bigness in the second family, is sufficiently large.

Our proof relies on and extends the lower bound techniques for estimating support size provided in Wu and Yang (2016b), incorporating specific conditions for the bigness problem. Firstly, unlike the support size estimation problem, we need our distributions to be fully-supported on the domain $[n]$ for the big distributions, whereas in their case, both families of distributions are allowed to be partially supported. Secondly, our optimization problem treats the threshold $1/(\beta n)$ as a variable, whereas the support size problem simply imposes the strict threshold of $1/n$. Thirdly, based on this construction, we must also give a direct upper bound for the maximum probability, which facilitates our later proofs for providing lower bounds for the matching and hypercube posets.

3.2. From bigness lower bounds to monotonicity lower bounds

In Section C, we show how to turn our lower bound results for bigness testing problem in Section A, into lower bounds for monotonicity testing in some fundamental posets, namely the matching poset and the Boolean hypercube poset.

Matching poset. To establish our lower bound for testing monotonicity of the matching poset, we construct our distribution p by assigning probability masses to the endpoints of edges (u_i, v_i) in our matching as follows: the vertices u_i's are assigned probability masses according to the T-bigness construction, whereas the vertices v_i's are uniformly assigned the threshold T as their probability masses; the assigned probabilities are then normalized into a proper probability distribution. We show that before normalization, $p(u_i) \leq T = p(v_i)$ if the original distribution is big; and otherwise, the distance to the monotonicity of the constructed distribution measures exactly the distance to the T-bigness property. We then show that the normalization step scales the entire distribution p down by only a constant factor, hence the lower bounds for the monotonicity testing over the matching poset with $2n$ vertices asymptotically preserves the parameters ϵ, s and $p_{\max}$ of the lower bound on bigness construction for n domain elements.

Hypercube poset. To achieve our results for the Boolean hypercube, we *embed* our distributions over the matching poset into two consecutive levels ℓ and $\ell-1$ of the hypercube (where ℓ denotes the number of ones in the vertices' binary representation). We pair up elements in these levels in such a way that distinct edges of the matching have incomparable endpoints: the algorithm must obtain samples of these matched vertices in order to decide whether the given distribution is monotone or not. We also place probability mass $p_{\max}$ on all other vertices on level ℓ and above, and probability mass 0 on all remaining vertices, in order to ensure that the distribution is monotone everywhere else. Lastly, we rescale the entire construction down into a proper probability distribution. Unlike the matching poset, sometimes this scaling factor is super-constant, shrinking the overall distance to monotonicity, ϵ, to sub-constant. Here, we make use of our upper bound on $p_{\max}$ of the bigness lower bound construction to determine the scaling factor.

3.3. Reduction from general posets to bipartite graphs

In Section D, we show that the problem of monotonicity testing of distributions over the *bipartite* posets is essentially the "hardest" case of monotonicity testing in general poset domains. That is, we show that for any distribution p over some poset domain of size n, represented as a directed graph G, there exists a distribution p' over a bipartite poset G' of size $2n$ such that (1) p preserves the total variation distance of p to monotonicity up to a small multiplicative constant factor, and (2) each sample for p' can be generated using one sample drawn from p. These properties together imply the following main theorem of the section.

Theorem 14 *Suppose that there exists an algorithm that tests monotonicity of a distribution over a bipartite poset domain of n elements using $s(n, \epsilon)$ samples for any total variation distance parameter $\epsilon > 0$. Then, there exists an algorithm that tests monotonicity of a distribution over any poset domain of n elements using $O(s(2n, \epsilon/4))$ samples.*

Our approach may be summarized as follows. We first show, in Theorem 15, that we may characterize (up to a constant factor) the distance of p' to monotonicity, as the size of the *maximum matching* on the *transitive closure* of G, denoted by $TC(G)$, where the weight $w(u, v) \coloneqq \max\{p(u) - p(v), 0\}$ represents the amount that (u, v) is *violating* the monotonicity condition. In particular, we have the following theorem:

Theorem 15 *Consider a poset $G = (E, V)$ and a distribution p over its vertices. Suppose every edge (u, v) in the $TC(G)$ has a weight of $\max(0, p(u) - p(v))$. Then, the total variation distance of p to any monotone distribution is within a factor of two of the weight of the maximum weighted matching in $TC(G)$.*

This crucial theorem provides a *combinatorial* way to approximate the distance to monotonicity for general posets, leading to our upcoming construction of p' for Theorem 14 as well as some algorithms in Section E. Theorem 15 is shown via LP duality: the *dual* LP for the problem of optimally "fixing" p to make it monotone, turns out to align with the maximum (fractional) matching problem on G's transitive closure. In particular, the dual constraints are of the form $\{Ay \leq b, y \geq 0\}$ where A is a totally unimodular matrix, implying that an *integral* optimal solution exists, namely the maximum matching.

To prove Theorem 14, given the original poset $G = (V, E)$, we create a bipartite poset with two copies u^- and u^+ of each original vertex $u \in V$: the vertices u^-'s and u^+'s form the bipartition of the new bipartite poset G' of size $2n$. We add (u^-, v^+) to the bipartite poset if (u, v) is in the transitive closure of G; that is, there exists a directed path from u to v in G. The new probability distribution p' on G', is created from p on G, by dividing the probability mass $p(u)$ equally among $p'(u^-)$ and $p'(u^+)$. Note that a sample from p' is obtained by drawing from p and adding the sign $-/+$ equiprobably. It follows via transitivity that p' is monotone over G' when p is monotone over G, and via Theorem 15 that if p is ϵ-far from monotone on G, then p' is also at least $\epsilon/4$-far from monotone over G'. These conditions allow us to test monotonicity of p on any general poset G by instead testing monotonicity of p' on a bipartite poset G' with parameter $\epsilon' = \epsilon/4$, as desired.

3.4. Upper bounds results

In Section E, we provide sublinear algorithms for testing bigness, and testing monotonicity of distributions over different poset domains.

Bigness testing. In Section E.1, we provide an algorithm for bigness testing. Observe that the T-bigness property is a *symmetric* property: closed under permutation of the labels of the domain elements $[n]$. Hence, we leverage the result of Valiant and Valiant (2017) that learns the counts of elements for each probability mass: $h_p(x) = |\{a : p(a) = x\}|$. Observe that the distance to T-bigness is proportional to the total "deficits" of elements with probability mass below T. Hence, this learned information suffices for constructing an algorithm for testing bigness, using a sub-linear, $O(\frac{n}{\epsilon^2 \log n})$, number of samples.

Monotonicity testing for matchings. Next, in Section E.2, we provide an algorithm for testing monotonicity of *matching* posets. We again resort to the work of Valiant and Valiant (2017) for learning the counts of elements for each *pair of probability masses*, with respect to a pair of distributions p_1, p_2 over the domain $[n]$, namely $h_{p_1,p_2}(x, y) = |\{a :$

$p_1(a) = x, p_2(a) = y\}|$, given $O(\frac{n}{\epsilon^2 \log n})$ samples each from p_1 and p_2. We hope to consider our distribution p over a matching $G = (S \cup T, E)$ with $E = \{(u_i, v_i)\}_{i \in [n]} \subset S \times T$ as a pair of distributions, namely p_S and p_T, representing probability masses p places over $u_i \in S$ and $v_i \in T$, respectively. Learning h_{p_S, p_T} would intuitively allows us to approximate p's distance to monotonicity by summing up the "violation" for pairs $x < y$. However, there are subtle challenges to this approach that do not present in the earlier case of bigness testing.

First, we must somehow rescale p_S and p_T up into distributions according to their total masses w_S, w_T placed by p. However, it is possible that, say, $p_S = o(1)$, making samples from S costly to generate by drawing i.i.d. samples from p. We resolve this issue via a reduction to a different distribution p' that approximately preserves the distance to bigness, while placing comparable total probability masses to S and T. Second, the algorithm of Valiant and Valiant (2017) learns $h_{p_1,p_2}(x, y)$ according to a certain distance function, that we must lower-bound by the total variation distance. In particular, this bound must be established under the presence of errors in the scaling factor, as w_S and w_T are not known to the algorithm. We overcome these technical issues, which yields an algorithm for testing monotonicity over matchings. We maintain the same asymptotic complexity as that of Valiant and Valiant (2017).

Monotonicity testing for bounded-degree bipartite graphs. Moving on, in Section E.3, we tackle the problem of monotonicity testing in *bipartite* posets; as shown in Section D, this bipartite problem captures the monotonicity testing problem of *any* poset. We make progress towards resolving this problem by offering our solution for the *bounded-degree* case. We turn the distribution p on a bipartite poset G of maximum degree Δ, into a distribution p' on a *matching* poset G' that approximately preserves the distance to monotonicity: applying the algorithm of Section E.2 above constitutes a monotonicity test for p with sample complexity $O(\frac{\Delta^3 n}{\epsilon^2 \log n})$.

Our reduction simply places Δ copies $v_1, \ldots, v_\Delta$ of each vertex $v \in V(G)$ into $V(G')$, then for each edge $(u, v) \in E(G)$, connects a pair of unused endpoints (u_i, v_j), as to create a matching subgraph of size $|E(G)|$ on G'. The probability distribution p' on $V(G')$ simply distributes probability mass $p(v)$ equally among all Δ copies v_i's. (Each remaining, isolated vertex is matched with a dummy 0-mass vertex, turning G' into a matching poset.) This new graph G' contains $O(\Delta n)$ vertices, and we show that $d_{TV}(p', \mathsf{Mon}(G')) \geq d_{TV}(p, \mathsf{Mon}(G))/(2\Delta)$ by explicitly creating a "low-cost" scheme for "fixing" p into a monotone distribution on G, based on the optimal scheme that turns p' monotone on G', charging at most an extra 2Δ-multiplicative factor.

Testing monotonicity of distributions that are uniform on a subset of the domain. In Section E.4, we show that for a specific broad family of distributions on directed bipartite graphs of arbitrary degree, we can test monotonicity of such distribution using $O(\frac{n^{2/3}}{\varepsilon} + \frac{1}{\varepsilon^2})$ samples. Namely, our result applies for distributions that are uniform on an arbitrary subset of the domain, given that every poset edge is directed from some vertex in the "bottom" part to some vertex in the "top" part of the graph. Our tester performs roughly the following: First, we sample a number of vertices from the graph and throw away ones that lie in the top part. For the remaining ones in the bottom part, denoted B, we identify their neighbors T in the top part, and determine whether or not they all belong to the support of the distribution. Since the distribution is uniform in its support, this condition is sufficient for the distribution to be monotone in the induced subgraph $G[B \cup T]$. The tester accepts when it cannot rule out the possibility that T has the maximum possible probability mass. Recall that if the distribution is ϵ-far from monotone, there must exist a large matching of "violated" edges. To this end, we show that the induced subgraph $G[B \cup T]$ contains many disjoint violated edges, implying that there are many vertices in T outside of the support: the probability mass on T will be noticeably small and the tester will reject.

Upper bound via trying all matchings. In Section E.5 we give another upper bound for testing monotonicity of a distribution with respect to a bipartite graph which, in this case, has a small number of induced subgraphs that contains a perfect matching of their vertices. In particular, we show that $O(\frac{\log M}{\epsilon^2})$ samples are sufficient for this task, where M is the number of such induced subgraphs. We note that this bound matches the general learning upper bound of $O(n/\epsilon^2)$ when M attains its maximum value of $2^{\Theta(n)}$, but can potentially be better when M is asymptotically smaller. The main idea of our tester is as follows: if the distribution is ϵ-far from monotone, there exists a matching of violated edges that is $\Theta(\epsilon)$-far from monotone. Hence, for each subgraph of G that admits a perfect matching, we may approximate the weight (violation amount) of this matching by simply comparing the total probability masses between the top part and the bottom part of the subgraph. We approximate these masses with error probability $O(1/M)$ for each subgraph, which allows us to apply a union bound over all subgraphs at the end. Our tester rejects if the weight of one such subgraph exceeds ϵ, or accepts otherwise.

Acknowledgments

MA is supported by funds from the MIT-IBM Watson AI Lab (Agreement No. W1771646), the NSF grants IIS-1741137, and CCF-1733808. TG is supported by the NSF grants CCF-1740751, CCF-1650733, CCF-1733808, and IIS-1741137. Part of this work was done while TG was a postdoctoral researcher at USC supported by Ilias Diakonikolas' USC startup grant. JP is supported by the NSF grants CCF-1565235, CCF-1650733, CCF-1733808, and IIS-1741137. RR is supported by by funds from the MIT-IBM Watson AI Lab (Agreement No. W1771646), the NSF grants CCF-1650733, CCF-1733808, IIS-1741137 and CCF-1740751. AY is supported by the NSF grants CCF-1650733, CCF-1733808, IIS-1741137 and the DPST scholarship, Royal Thai Government. This work was completed while AY was at CSAIL, MIT.

References

Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, and Ananda Theertha Suresh. A competitive test for uniformity of monotone distributions. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, pages 57–65, 2013.

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, pages 3591–3599, 2015.

Michal Adamaszek, Artur Czumaj, and Christian Sohler. Testing monotone continuous distributions on high-dimensional real cubes. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 56–65, 2010.

Tugkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 381–390, New York, NY, USA, 2004. ACM. ISBN 1-58113-852-0.

Tugkan Batu, Sanjoy Dasgupta, Ravi Kumar, and Ronitt Rubinfeld. The complexity of approximating the entropy. *SIAM J. Comput.*, 35(1):132–150, 2005. doi: 10.1137/S0097539702403645. URL https://doi.org/10.1137/S0097539702403645.

Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1021–1032, 2016.

Arnab Bhattacharyya, Eldar Fischer, Ronitt Rubinfeld, and Paul Valiant. Testing monotonicity of distributions over general partial orders. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:27, 2010.

Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. A $o(d) \cdot$ polylog n monotonicity tester for boolean functions over the hypergrid $[n]^d$. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2133–2151, 2018.

Clément L. Canonne. Big data on the rise: Testing monotonicity of distributions. In *ICALP*, 2015.

Clément L. Canonne. A short note on Poisson tail bounds. 2017. URL http://www.cs.columbia.edu/~ccanonne/files/misc/2017-poissonconcentration.pdf. Available online at http://www.cs.columbia.edu/ ccanonne/files/misc/2017-poissonconcentration.pdf.

Clément L. Canonne, Ilias Diakonikolas, Themis Gouleakis, and Ronitt Rubinfeld. Testing shape restrictions of discrete distributions. *Theory Comput. Syst.*, 62(1):4–62, 2018. doi: 10.1007/s00224-017-9785-6. URL https://doi.org/10.1007/s00224-017-9785-6.

Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and lipschitz testing over hypercubes and hypergrids. In *Symposium on Theory of Computing Conference (STOC)*, pages 419–428, 2013.

Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014.

Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. Learning k-modal distributions via testing. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1371–1385, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics. URL http://dl.acm.org/citation.cfm?id=2095116.2095224.

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 '13, pages 1833–1852, Philadelphia, PA, USA, 2013. Society for Industrial and Applied Mathematics. ISBN 978-1-611972-51-1. URL http://dl.acm.org/citation.cfm?id=2627817.2627948.

Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Randomization, Approximation, and Combinatorial Algorithms and Techniques, Third International Workshop on Randomization and Approximation Techniques in Computer Science, and Second International Workshop on Approximation (RANDOM-APPROX)*, pages 97–108, 1999.

Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 474–483, 2002.

Alain Ghouila-Houri. Caractérisation des matrices totalement unimodulaires. *C. R. Acad. Sci. Paris*, 254:1192–1194, 1962.

Oded Goldreich, Shafi Goldwasser, Eric Lehman, and Dana Ron. Testing monotonicity. In *39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 426–435, 1998.

Johannes Kraus, Panayot S. Vassilevski, and Ludmil Zikatanov. Polynomial of best uniform approximation to 1/x and smoothing in two-level methods. *Comput. Meth. in Appl. Math.*, 12(4):448–468, 2012. URL http://www.degruyter.com/view/j/cmam.2012.12.issue-4/cmam-2012-0026/cmam-2012-0026.xml.

Eric Lehman and Dana Ron. On disjoint chains of subsets. *J. Comb. Theory, Ser. A*, 94 (2):399–404, 2001.

Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam D. Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM J. Comput.*, 39(3):813–842, 2009.

Ronitt Rubinfeld and Rocco A. Servedio. Testing monotone high-dimensional distributions. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 147–156, 2005. doi: 10.1145/1060590.1060613. URL http://doi.acm.org/10.1145/1060590.1060613.

A.F. Timan. *International Series of Monographs in Pure and Applied Mathematics*. Number v. 34 in International Series of Monographs in Pure and Applied Mathematics. Pergamon

Press; [distributed in the Western Hemisphere by Macmillan, New York], 1963. URL https://books.google.com/books?id=2R-4AAAAIAAJ.

Gregory Valiant and Paul Valiant. Instance optimal learning of discrete distributions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, (STOC)*, pages 142–155, 2016.

Gregory Valiant and Paul Valiant. Estimating the unseen: Improved estimators for entropy and other properties. *J. ACM*, 64(6):37:1–37:41, 2017. doi: 10.1145/3125643. URL http://doi.acm.org/10.1145/3125643.

Paul Valiant. Testing symmetric properties of distributions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 383–392, 2008.

Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6): 3702–3720, 2016a.

Yihong Wu and Pengkun Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *arXiv preprint arXiv:1504.01227v2*, 2016b.

Appendix A. A Lower Bound for the Bigness Testing Problem

In this section, we give a lower bound for the bigness testing problems. As described in the overview in Subsection 3.1, we provide two random processes for generating samples from two families of distributions, such that one family consists of "big" distributions, and the other family largely of "ϵ-far from big" distributions, and then show that they are hard to distinguish.

First, we define a random process that, given a prior distribution, P_V, over non-negative numbers, generates a random probability distribution over the domain elements $[n]$, and then draws samples from it. More specifically, let V be a random variable drawn from P_V, and we also use P_V to denote the probability density function (PDF) over V; for now we require $\mathbf{E}[V] = 1$, and will specify further desired properties momentarily. We generate an *approximate* probability distribution p according to P_V. The distribution p is constructed by having each domain element i choose its probability $p(i)$, in an i.i.d. fashion, from the prior distribution, P_V, over possible probabilities. Then, we construct a histogram of roughly s samples from p according to the following steps:

- **Step 1:** Generate n i.i.d. random variables $V_1, V_2, \ldots, V_n$ according to P_V, then form the following *probability vector* over $[n]$:

$$p = \frac{1}{n}(V_1, V_2, \ldots, V_n)\,.$$

  Remark that, while p is not necessarily a probability distribution under this notion, the condition $\mathbf{E}[V] = 1$ suggests that the total probability masses of p is likely to be centered around 1. So, p is likely to be approximately a probability distribution, and can be normalized into one while modifying individual entries $p(i)$'s by only a small multiplicative factor.

- **Step 2:** Draw n independent random variables $h_i \sim \mathbf{Poi}(s \cdot p(i))$ (namely $h_i \sim \mathbf{Poi}(sV_i/n)$) for $i = 1, 2, \ldots, n$, and output $h = (h_1, h_2, \ldots, h_n)$ as the histogram of the samples. While we do not explicitly normalize p, since p is an approximate probability distribution, this histogram still captures (with high probability) $\Omega(s)$ Poissonized samples drawn from the normalization of p.

The goal in this section is to find two prior distributions P_V and $P_{V'}$, to generate two probability vectors p and p' according to the above process such that after the normalization, p and p' have the desire properties: p is big (every $p(i)$ is at least the threshold T), and p' is ϵ-far from any big distribution (p' contains a significant number of entries i with $p'(i) = 0$). Then, we generate two histograms h and h' according to p and p' respectively. If the histograms h and h' are hard to distinguish, then we can establish a lower bound for the bigness property. This requirement will show up as constraints for designing two prior distributions, P_V and $P_{V'}$, to achieve these families of distributions with high probability. Below, we summarize the conditions that we need the prior distributions to hold (with high probability):

1. The probability vectors p and p' are approximate probability distributions; that is, all of their coordinates are non-negative and their total probability masses are each close to one.

2. After scaling the probability vectors p and p' above into respective probability distributions, the normalization of p is T-big, and the normalization of p' is ϵ-far from any T-big distribution.

3. The total numbers of (Poissonized) samples in h and h' drawn from the normalization of p and p' are each $\Omega(s)$.

4. Given h or h', distinguishing whether it is generated from P_V or $P_{V'}$ with success probability $2/3$ requires h or h' to contain a large number of samples.

5. Additionally, we will bound the largest probability mass $p_{\max}$ that the normalized distributions place on any domain element – this part is not necessary for this section, but will be useful for the reduction between monotonicity testing and bigness testing later on.

We state this result as the following theorem.

Theorem 3 *For integer $L = O(\log n)$ and sufficiently small $\epsilon = \Omega(1/n)$, there exist a parameter $\beta = \beta(L, \epsilon)$ and two distributions $\mathcal{H}^+$ and $\mathcal{H}^-$ over the set of possible histograms of size at least $s = \Omega\left(n^{1-1/L} \log^2(1/\epsilon)/L\right)$ with the following properties:*

- *The histogram generated from $\mathcal{H}^+$ is drawn from a $1/(\beta n)$-big distribution.*

- *The histogram generated from $\mathcal{H}^-$ is drawn from a distribution which is ϵ-far from any $1/(\beta n)$-big distribution.*

- $d_{TV}\left(\mathcal{H}^+, \mathcal{H}^-\right) \leq 0.01$.

- *The largest probability mass among any elements in any probability distributions above (from which the histograms are drawn) is $p_{max} = O(L^2/(n \log^2(1/\epsilon)))$.*

An important case of this theorem is when $L = \Theta(\log n)$, where we establish a near-linear sample complexity lower bound of $\Omega(n/\log n)$ for the general problem of bigness testing as follows.

Corollary 4 *For sufficiently small parameter $\epsilon = \Omega(1/n)$, there exists a parameter $\beta = \beta(\epsilon)$ such that any algorithm that can distinguish whether a distribution over $[n]$ is $1/(\beta n)$-big or ϵ-far from any $1/(\beta n)$-big distribution with probability $2/3$ requires $\Omega(n \log^2(1/\epsilon)/\log n)$ samples. In particular when ϵ is a constant, β is constant, then any such algorithm requires $\Omega(n/\log n)$ samples.*

Proof By Theorem 3, there exist $\mathcal{H}^+$ and $\mathcal{H}^-$ with the aforementioned properties. Any $1/(\beta n)$-bigness tester has to distinguish between $\mathcal{H}^+$ and $\mathcal{H}^-$ with probability at least $2/3$. On the other hand, the total variation distance between $\mathcal{H}^+$ and $\mathcal{H}^-$ is at most 0.01. Therefore, no algorithm can distinguish between them while receiving $s/4 = \Theta(n \log^2(1/\epsilon)/\log n)$ samples with probability more than $(1+0.01)/2$. Therefore, testing $1/(\beta n)$-bigness requires $\Omega(n \log^2(1/\epsilon)/\log n)$ samples.

Note that in the proof of Theorem 3, β is determined by Lemma 5, and it is bounded by $1/\epsilon$. Thus, if ϵ is a constant then β is also a constant. Thus, the required sample complexity becomes $\Omega(n/\log n)$. ■

Appendix B. Proof of the lemmas in Section A

B.1. Proof of Theorem 3

Proof [proof of Theorem 3] Let positive values ν, λ, β, and a positive integer L be a set of parameters with the following property that we determine more precisely later:

$$0 < \nu \leq \frac{1}{2}, \quad \lambda > 1 + \nu, \quad 1 \leq \beta \leq \min\left\{\frac{1}{\epsilon}, \lambda\right\} \quad \text{and} \quad L = O(\log n).$$

Throughout this section, we consider the bigness threshold $T = 1/(\beta n)$, and note that the value β itself may depend on the error parameter ϵ, an the number of matched moments L. Note also that ν is a constant.

We propose the following optimization problem, **OP1**, such that its optimal solution, specifying P_V and $P_{V'}$ satisfies the requirements of the theorem. Recall that p and p' are generated by drawing n i.i.d samples, V_i's and V_i''s, from P_V and $P_{V'}$ respectively:

$$p = \frac{1}{n}(V_1, V_2, \ldots, V_n) \qquad p' = \frac{1}{n}(V_1', V_2', \ldots, V_n')$$

Intuitively speaking, as P_V aims to generate T-big distributions, we must ensure that the V_i's are bounded away from $1/\beta$, so that $p(i) \sim V_i/n$ has expected value higher than $T = 1/(\beta n)$. At the same time, we hope to maximize the probability that $V_i' = 0$ so that p' is far from any T-big distribution, under the constraint that the first L moments of P_V and

$P_{V'}$ are exactly matched, as to ensure that the resulting distributions of histograms $\mathcal{H}$ and $\mathcal{H}'$ are statistically close. The objective value of this optimization problem corresponds to the expected distance of p' to the closest T-big distribution in total variation distance. To clarify the notation, λ and ν are given to us. The unknown variables in **OP1** are the PDFs P_V and $P_{V'}$ of two random variables V and V', respectively, as well as the scaling variable $\beta > 0$. The parameter λ roughly specifies the ratio between the largest and the smallest non-zero probabilities that p and p' can take.[1]

$$\begin{aligned} \text{Definition of } \mathbf{OP1}: \quad \sup \quad & \tfrac{1}{\beta}\Pr[V'=0] \\ s.t. \quad & \mathbf{E}[V] = \mathbf{E}[V'] = 1 \\ & \mathbf{E}[V^j] = \mathbf{E}[V'^j] \quad \text{for } j = 1, 2, \ldots, L \\ & V \in \left[\tfrac{1+\nu}{\beta}, \tfrac{\lambda}{\beta}\right], V' \in \{0\} \cup \left[\tfrac{1+\nu}{\beta}, \tfrac{\lambda}{\beta}\right] \;\text{ and } \beta > 0. \end{aligned} \tag{1}$$

In the following lemma, we find the optimal value of **OP1**. We use $\mathbf{OPT}(A)$ to refer to the optimal value of optimization problem A.

Lemma 5 *For any ν and λ such that $0 < 1+\nu < \lambda$, there exists a scaling parameter, β, in $[1+\nu, \min(\lambda, 1/\mathbf{OPT}(\mathbf{OP1}))]$ such that*

$$\mathbf{OPT}(\mathbf{OP1}) = \left(\frac{1}{\sqrt{1+\nu}} - \frac{1}{\sqrt{\lambda}}\right)^2 \left(\frac{\sqrt{\frac{\lambda}{1+\nu}} - 1}{\sqrt{\frac{\lambda}{1+\nu}} + 1}\right)^{L-2}.$$

The proof of Lemma 5 is postponed to Section B.2.

Let the value of β be determined by the above lemma, and set d to be $\mathbf{OPT}(\mathbf{OP1})$.

Recall our wish list of five properties for the priors, P_V and P'_V, that we propose in the introduction of Section A. We define the following "good" events , which hold with high probability, to formalize the properties of the generated vectors p and p'.

$$E = \left\{ \left| \sum_{i=1}^{n} \frac{V_i}{n} - 1 \right| \le \nu, \text{ and } \sum_{i=1}^{n} N_i > s(1-\nu)/2 \right\}.$$

and

$$E' = \left\{ \left| \sum_{i=1}^{n} \frac{V'_i}{n} - 1 \right| \le \nu \,, \, r \ge \frac{\beta n d}{2}, \text{ and } \sum_{i=1}^{n} N'_i > s(1-\nu)/2 \right\}$$

where r is the number of elements i such that V'_i is zero. Roughly speaking, these events state that $p = \frac{1}{n}(V_i)_{i\in[n]}$ and $p' = \frac{1}{n}(V'_i)_{i\in[n]}$, generated in step 1, are approximate probability distributions (having total masses in $[1-\nu, 1+\nu] = \Theta(1)$), and step 2 generates sufficient numbers of samples in the histogram (at least $s(1-\nu)/2 = \Omega(s)$ each). Further,

[1] Note that P_V and $P_{V'}$ are on a continuous domain. However, $P_{V'}$ will additionally have a non-negligible probability *mass* placed at value 0. In fact, it turns out that in the optimal solution, P_V and $P_{V'}$ are only supported on a few distinct values ($\Theta(L) = O(\log n)$ of them), so the optimal P_V and $P_{V'}$ assume the role of probability mass functions rather than PDFs.

p' consists of as many as $r \geq \beta nd/2$ elements with probability mass 0, thus is at distance at least $rT \geq d/2$ from any T-big distribution – we will set $d \geq 2\epsilon$ to reach the desired result.

In the following lemma, we show that conditioning on E and E', after running the process using the priors P_V and $P_{V'}$, the generated histogram h is a sufficiently large set of samples from a $1/(\beta n)$-big distribution, and histogram h' is a sufficiently large set of samples from a distribution which is ϵ-far from any $1/(\beta n)$-big distribution. In addition, the total variation distance between the distribution over h's and h''s is bounded when P_V, $P_{V'}$ form a solution of **OP1**. More precisely, let $\mathcal{H}$ denote the distribution over histograms h generated by the process when the prior is P_V, and let $\mathcal{H}_E$ be the distribution over histograms h conditioning on E. We define $\mathcal{H}'$ and $\mathcal{H}_{E'}$ similarly. In the following lemma, we bound the total variation distance between $\mathcal{H}_E$ and $\mathcal{H}'_{E'}$ as well.

Lemma 6 *Let $P_V, P_{V'}$, and $\beta \in [1, 1/d]$ form a solution of* **OP1** *with objective value $d \geq 2\epsilon$. Suppose P_V and $P_{V'}$ are the prior distributions to generate histograms h and h' according to the process. Then, h given event E is a histogram of a set of at least $s(1-\nu)/2$ samples from a $1/(\beta n)$-big distribution, whereas h' given E' is a histogram of a set of at least $s(1-\nu)/2$ samples that are drawn from a distribution which is ϵ-far from any $1/(\beta n)$-big distribution. Moreover,*

$$d_{TV}\left(\mathcal{H}_E, \mathcal{H}'_{E'}\right) \leq \frac{2\lambda}{\beta n \nu^2} + \exp\left(-\frac{\beta nd}{8}\right) + 2\exp\left(-\frac{s(1-\nu)}{6}\right) + n\left(\frac{es\lambda}{2nL}\right)^L .$$

Lastly, the largest probability mass among any elements in any probability distributions (from which the samples are drawn) is $\lambda/(n(1-\nu))$.

The proof of Lemma 6 is given in Section B.3.

Now, we assign the parameters, ν, λ, and s, as follows:

$$\nu := 1/2, \quad \lambda := (1+\nu)\cdot\left(\frac{4(L-2)}{\ln\left(1/(27\epsilon)\right)} - 1\right)^2, \text{ and } s := \left\lfloor \frac{Ln}{2e\lambda} \right\rfloor$$

Recall that we set d to be the optimal value of **OP1**, and Lemma 5 tells us its value. We show that in this setting d is at least 2ϵ. Let ρ be $\sqrt{\lambda/(1+\nu)}$. Then, we have:

$$\begin{aligned} d := \left(\frac{1}{\sqrt{1+\nu}} - \frac{1}{\sqrt{\lambda}}\right)^2 \left(\frac{\sqrt{\frac{\lambda}{1+\nu}} - 1}{\sqrt{\frac{\lambda}{1+\nu}} + 1}\right)^{L-2} &\geq \frac{1}{1+\nu}\left(1 - \frac{1}{\sqrt{\frac{\lambda}{1+\nu}}}\right)^2 \left(\frac{\sqrt{\frac{\lambda}{1+\nu}} - 1}{\sqrt{\frac{\lambda}{1+\nu}} + 1}\right)^{L-2} \\ = \frac{2}{3}\left(1-\frac{1}{\rho}\right)^2\left(1 - \frac{2}{\rho+1}\right)^{L-2} > \frac{2}{27}\left(\frac{1}{e^2}\right)^{\frac{2(L-2)}{\rho+1}} &\geq \frac{2}{27}\exp\left(-\frac{4(L-2)}{\rho+1}\right) \geq 2\epsilon\,. \end{aligned}$$

as long as $\rho \geq 1.5$. It is not hard to see that, for sufficiently large n and $\epsilon \geq c/n$ for sufficiently large constant c, then $\rho \geq 1.5$ holds, yielding $d \geq 2\epsilon$, for every $\epsilon \leq c_0$, where $c_0 < 1/2$ is a constant.

Let $\mathcal{H}^+$ and $\mathcal{H}^-$ be $\mathcal{H}_E$ and $\mathcal{H}'_{E'}$ respectively. By Lemma 6, the total variation distance between $\mathcal{N}^+$ and $\mathcal{N}^-$ is at most 0.01, while s and $p_{\max}$ behave according to the claimed respective asymptotic bounds. Hence, the proof is complete. ■

B.2. Proof of Lemma 5

Lemma 7 *For any ν and λ such that $0 < 1 + \nu < \lambda$, there exists a scaling parameter, β, in $[1 + \nu, \min(\lambda, 1/\mathbf{OPT}(\mathbf{OP1}))]$ such that*

$$\mathbf{OPT}(\mathbf{OP1}) = \left(\frac{1}{\sqrt{1+\nu}} - \frac{1}{\sqrt{\lambda}}\right)^2 \left(\frac{\sqrt{\frac{\lambda}{1+\nu}} - 1}{\sqrt{\frac{\lambda}{1+\nu}} + 1}\right)^{L-2} .$$

Proof To prove the lemma, we introduce an auxiliary linear program (**LP2**) that is known to have an optimal value of the right hand side of the above equation. We prove the **LP2** has the same optimal objective value as **OP1** to prove the lemma. For two given parameters ν and λ, we define the following LP over two random variables X, X'.

$$\begin{aligned} \text{Definition of } \mathbf{LP2}: \quad \sup \quad & \mathbf{E}\left[\tfrac{1}{X}\right] - \mathbf{E}\left[\tfrac{1}{X'}\right] \\ s.t. \quad & \mathbf{E}[X^j] = \mathbf{E}[X'^j] \quad \text{for } j = 1, 2, \ldots, L-1 \\ & X, X' \in [1+\nu, \lambda] \end{aligned} \tag{2}$$

To interpret this LP, assume the unknown variable is the PDF's of the random variables X and X'. Thus, for any number x in $[1+\nu, \lambda]$, we want to find $P_X(x)$ and $P_{X'}(x)$. Note that this optimization problem is linear since all the expectations above are a linear function of P_X and $P_{X'}$. Moreover, there is an implicit constraint here that the integral of P_X and $P_{X'}$ should be one since they are probability distributions.

Observe that there exists a trivial solution where X and X' are two identically-distributed random variables, so **LP2** is feasible and its optimal objective value is at least zero. Let $\mathcal{X}^*$ and $\mathcal{X}'^*$ be a pair of random variables forming an optimal solution for **LP2**, and let $\beta^* = 1/\mathbf{E}[1/\mathcal{X}^*]$. Since all X and X' are in $[1+\nu, \lambda]$, then β^* is also in $[1+\nu, \lambda]$. On the other hand, since $\mathcal{X}'^*$ is positive and bounded, then $\mathbf{E}[1/\mathcal{X}'^*] > 0$ and thus $\mathbf{E}[1/\mathcal{X}^*] > \mathbf{OPT}(\mathbf{LP2})$; hence β^* is at most $1/\mathbf{OPT}(\mathbf{LP2})$.

Now, we argue that **LP2** and **OP1** have the same optimal value. We introduce two new random variables $\mathcal{V}^*$ and $\mathcal{V}'^*$ with the following PDFs, and later we show they form an optimal solution for **OP1**.

$$\begin{aligned} P_{\mathcal{V}^*}(v) &\coloneqq \frac{\beta^*}{v} P_{\mathcal{X}^*}(\beta^* v) + \left(1 - \beta^* \mathbf{E}\left[\frac{1}{\mathcal{X}^*}\right]\right) \delta_0(v), \quad \text{and} \\ P_{\mathcal{V}'^*}(v) &\coloneqq \frac{\beta^*}{v} P_{\mathcal{X}'^*}(\beta^* v) + \left(1 - \beta^* \mathbf{E}\left[\frac{1}{\mathcal{X}'^*}\right]\right) \delta_0(v) \end{aligned}$$

In the above equations, with a slight abuse of notation we say that $1/v$ is zero for $v = 0$; that is, the probability *mass* for $v = 0$ is given by the respective second terms. Since β^* is defined to be $1/\mathbf{E}[1/\mathcal{X}^*]$, the second term in $P_{\mathcal{V}^*}$ is zero for all v in particular for $v = 0$. We define our notation in this fashion in order to make the calculations for $\mathcal{V}^*$ and $\mathcal{V}'^*$ analogous, so we may write our proof compactly.

Now, we show that the proposed variables $\mathcal{V}^*$, $\mathcal{V}'^*$ and β^* form a feasible solution for **OP1**. First, we show that the domain of $\mathcal{V}^*$ and $\mathcal{V}'^*$ are as stated in the definition of **OP1** in Equation 1. Then, we show $P_{\mathcal{V}^*}$ and $P_{\mathcal{V}'^*}$ are probability distribution, and we prove the constraints of **OP1** hold as well.

First, consider the domain of the random variables. Clearly the domain does not include the numbers where the PDF is zero, so we prove that the $P_{\mathcal{V}^*}$ and $P_{\mathcal{V}'^*}$ are (potentially) non-zero only when when $\mathcal{V}^*$ and $\mathcal{V}'^*$ are in the rage specified by the domain constraints of the **OP1**. Recall that the second term in $P_{\mathcal{V}^*}$ is always zero. Thus, $P_{\mathcal{V}^*}$ could be potentially non-zero only if x equal to βv has a non-zero error probability according to $P_{\mathcal{X}^*}$. Therefore, $\mathcal{V}^*$ is always in $[(1+\nu)/\beta^*, \lambda/\beta^*]$. For $\mathcal{V}'^*$, in addition to the value $v \in [(1+\nu)/\beta^*, \lambda/\beta^*]$, v could be zero as well since the second term in the definition of $P_{\mathcal{V}'^*}$ may be non-zero at $v = 0$. Thus, $\mathcal{V}'^*$ is always in $\{0\} \cup [(1+\nu)/\beta^*, \lambda/\beta^*]$.

In addition, $P_{\mathcal{V}^*}$ (and similarly $P_{\mathcal{V}'^*}$) is a probability distribution since the integral of the PDF is one:

$$\begin{aligned}\int_{-\infty}^{\infty} P_{\mathcal{V}^*}(v)dv &= \int_{(1+\nu)/\beta^*}^{\lambda/\beta^*} \frac{\beta^*}{v}\, P_{\mathcal{X}^*}(\beta^* v)dv + \left(1 - \beta^*\,\mathbf{E}\left[\frac{1}{\mathcal{X}^*}\right]\right)\int_{-\infty}^{\infty} \delta_0(v)dv \\ &= \int_{1+\nu}^{\lambda} \frac{\beta^{*2}}{x}\, P_{\mathcal{X}^*}(x)\cdot\frac{1}{\beta^*}\,dx + \left(1 - \beta^*\,\mathbf{E}\left[\frac{1}{\mathcal{X}^*}\right]\right) \\ &= \beta^*\,\mathbf{E}\left[\frac{1}{\mathcal{X}^*}\right] + \left(1 - \beta^*\,\mathbf{E}\left[\frac{1}{\mathcal{X}^*}\right]\right) = 1,\end{aligned}$$

where the second equality is derived by substituting v with x/β^*.

Now, we focus on the constraints of **OP1**. The first constraint is $\mathbf{E}[\mathcal{V}^*] = \mathbf{E}[\mathcal{V}'^*] = 1$. Below we show that the expected value of $\mathcal{V}^*$ is 1.

$$\begin{aligned}\mathbf{E}[\mathcal{V}^*] = \int_{-\infty}^{\infty} v\, P_{\mathcal{V}^*}(v)dv &= \int_{(1+\nu)/\beta^*}^{\lambda/\beta^*} \beta^*\, P_{\mathcal{X}^*}(\beta^* v)dv + \left(1 - \beta^*\,\mathbf{E}\left[\frac{1}{\mathcal{X}^*}\right]\right)\int_{-\infty}^{\infty} v\,\delta_0(v)dv \\ &= \int_{1+\nu}^{\lambda} \beta^*\, P_{\mathcal{X}^*}(x)\cdot\frac{1}{\beta^*}\,dx = 1\end{aligned}$$

One can similarly show that $\mathbf{E}[\mathcal{V}'^*] = 1$, and the constraint holds.

The second constraint is that the first L moments of $\mathcal{V}^*$ and $\mathcal{V}'^*$ are matched: $\mathbf{E}[\mathcal{V}^{*j}] = \mathbf{E}[\mathcal{V}'^{*j}]$ for j in $[L]$. The previous constraint implies that the first moments, $\mathbf{E}[\mathcal{V}^*]$ and $\mathbf{E}[\mathcal{V}'^*]$, are equal, so here we focus on the second and higher moments. Fix j in $\{2, \ldots, L\}$. For the j-th moment of $\mathcal{V}^*$, we have:

$$\begin{aligned}\mathbf{E}[\mathcal{V}^{*j}] = \int_{-\infty}^{\infty} v^j\, P_{\mathcal{V}^*}(v)dv &= \int_{(1+\nu)/\beta^*}^{\lambda/\beta^*} \beta^*\, v^{j-1}\, P_{\mathcal{X}^*}(\beta^* v)dv + \left(1 - \beta^*\,\mathbf{E}\left[\frac{1}{\mathcal{X}^*}\right]\right)\int_{-\infty}^{\infty} v^j\,\delta_0(v)dv \\ &= \int_{1+\nu}^{\lambda} \frac{x^{j-1}}{\beta^{*j-2}}\, P_{\mathcal{X}^*}(x)\cdot\frac{1}{\beta^*}\,dx = \frac{1}{\beta^{*j-1}}\,\mathbf{E}[\mathcal{X}^{*j-1}].\end{aligned}$$

We can similarly show the same condition for $\mathbf{E}[\mathcal{V}'^{*j}]$. Since $\mathcal{X}^*$ and $\mathcal{X}'^*$ satisfies the moment matching constraints of **LP2**, we derive the moment matching constraints of **OP1** as follows:

$$\mathbf{E}[\mathcal{V}'^{*j}] = \frac{1}{\beta^{*j-1}}\,\mathbf{E}[\mathcal{X}^{*j-1}] = \frac{1}{\beta^{*j-1}}\,\mathbf{E}[\mathcal{X}'^{*j-1}] = \mathbf{E}[\mathcal{V}'^{*j}]\,.$$

Therefore, $\mathcal{V}^*$, $\mathcal{V}'^*$ and β^* form a feasible solution for **OP1**. Thus, the objective function according to $\mathcal{V}^*$, $\mathcal{V}'^*$ is at most the optimal value of **OP1**:

$$\mathbf{OPT(OP1)} \ge \frac{1}{\beta^*}\,\Pr[\mathcal{V}'^* = 0]$$

On the other hand, the objective value of **OP1** and **LP2** are the same on the two solutions we discussed:

$$\frac{1}{\beta^*}\Pr[\mathcal{V}'^* = 0] = \frac{1}{\beta^*}\left(1 - \beta^*\,\mathbf{E}\left[\frac{1}{\mathcal{X}'^*}\right]\right) = \mathbf{E}\left[\frac{1}{\mathcal{X}^*}\right] - \mathbf{E}\left[\frac{1}{\mathcal{X}'^*}\right] = \mathbf{OPT}(\mathbf{LP2})$$

where the last equality is true, since we chose X and X' to be the optimal solution of **LP2** at the beginning.

$$\mathbf{OPT}(\mathbf{OP1}) \geq \frac{1}{\beta^*}\Pr[\mathcal{V}'^* = 0] = \mathbf{OPT}(\mathbf{LP2})\,. \tag{3}$$

We continue the proof by showing that the above inequality is true in the other direction, i.e, **OPT**(**OP1**) is at most **OPT**(**LP2**). Let $P_{\mathcal{V}}$, $P_{\mathcal{V}'}$ and β form a feasible solution for **OP1**. We define random variables $\mathcal{X}$ and $\mathcal{X}'$ with the following PDFs, and show that they form a feasible solution for **LP2** in Equation 2 with the same objective value as $\mathcal{V}$ and $\mathcal{V}'$ in the **OP1**:

$$P_{\mathcal{X}}(x) \coloneqq \frac{x}{\beta^2}P_{\mathcal{V}}\left(\frac{x}{\beta}\right), \text{ and } \quad P_{\mathcal{X}'}(x) \coloneqq \frac{x}{\beta^2}P_{\mathcal{V}'}\left(\frac{x}{\beta}\right).$$

First, we show that the domain of $\mathcal{X}$ and $\mathcal{X}'$ matches with the domain constraint in **LP2**. Similar to the previous part, we prove that the PDF's are zero outside the interval specified by the domain constraint $[1+\nu, \lambda]$. Observe that $P_{\mathcal{X}}(x)$ is non-zero if and only if x and $P_{\mathcal{V}}(x/\beta)$ are both non-zero, so x/β has to be in $[(1+\nu)/\beta, \lambda/\beta]$. Thus, the domain of the random variable $\mathcal{X}$ (and similarly $\mathcal{X}'$) is $[1+\nu, \lambda]$.

Moreover, note that $P_{\mathcal{X}}$ (and similarly $P_{\mathcal{X}'}$) is a probability distribution:

$$\int_{-\infty}^{+\infty} P_{\mathcal{X}}(x)dx = \int_{1+\nu}^{\lambda} \frac{x}{\beta^2}\cdot P_{\mathcal{V}}\left(\frac{x}{\beta}\right)dx = \int_{(1+\nu)/\beta}^{\lambda/\beta} \frac{v}{\beta}\cdot P_{\mathcal{V}}(v)\cdot \beta dv = \mathbf{E}[\mathcal{V}] = 1$$

where the equation is derived by replacing x/β with a new variable v. Now, we show that the constraints of **LP2** are satisfied for $\mathcal{X}$ and $\mathcal{X}'$. Fix $j \in [L-1]$. We show the j-th moment of $\mathcal{X}$ and $\mathcal{X}'$ are equal:

$$\mathbf{E}[\mathcal{X}^j] = \int_{-\infty}^{+\infty} x^j P_{\mathcal{X}}(x)dx = \int_{1+\nu}^{\lambda} \frac{x^{j+1}}{\beta^2}\cdot P_{\mathcal{V}}\left(\frac{x}{\beta}\right)dx = \int_{(1+\nu)/\beta}^{\lambda/\beta} \frac{\beta^j\, v^{j+1}}{\beta}\cdot P_{\mathcal{V}}(v)\cdot \beta dv = \beta^j\,\mathbf{E}[\mathcal{V}^{j+1}]\,.$$

Similarly, one can show $\mathbf{E}[\mathcal{X}'^j]$ is equal to $\beta^j\mathbf{E}[\mathcal{V}'^{j+1}]$. Since the pair $\mathcal{V}$ and $\mathcal{V}'$ satisfies the moment matching constraints of **OP1**, then $\mathbf{E}[\mathcal{V}^{j+1}]$ is equal to $\mathbf{E}[\mathcal{V}'^{j+1}]$. Therefore, $\mathbf{E}[\mathcal{X}^j]$ is equal to $\mathbf{E}[\mathcal{X}'^j]$.

Now, we focus on the objective functions of the **OP1** and **LP2**. We have:

$$\begin{aligned}
\mathbf{E}\left[\frac{1}{\mathcal{X}}\right] - \mathbf{E}\left[\frac{1}{\mathcal{X}'}\right] &= \int_{1+\nu}^{\lambda} \frac{1}{x} \cdot P_{\mathcal{X}}(x) dx - \int_{1+\nu}^{\lambda} \frac{1}{x'} \cdot P_{\mathcal{X}'}(x') dx' \\
&= \int_{1+\nu}^{\lambda} \frac{1}{\beta^2} \cdot P_{\mathcal{V}}\left(\frac{x}{\beta}\right) dx - \int_{1+\nu}^{\lambda} \frac{1}{\beta^2} \cdot P_{\mathcal{V}'}\left(\frac{x'}{\beta}\right) dx' \\
&= \int_{(1+\nu)/\beta}^{\lambda/\beta} \frac{1}{\beta^2} \cdot P_{\mathcal{V}}(v) \cdot \beta dv - \int_{(1+\nu)/\beta}^{\lambda/\beta} \frac{1}{\beta^2} \cdot P_{\mathcal{V}'}(v') \cdot \beta dv' \\
&= \frac{1}{\beta} \cdot \left(\int_{(1+\nu)/\beta}^{\lambda/\beta} P_{\mathcal{V}}(v) \cdot dv - \int_{(1+\nu)/\beta}^{\lambda/\beta} P_{\mathcal{V}'}(v') \cdot dv'\right) \\
&= \frac{1}{\beta} \cdot \left(1 - \left(1 - \Pr[\mathcal{V}' = 0]\right)\right) = \frac{1}{\beta} \Pr[\mathcal{V}' = 0] \,.
\end{aligned}$$

Now that for any feasible solution of **OP1**, there exists a feasible solution for **LP2** with the same objective value, one can conclude that the optimal value of **OP1**, **OPT**(**OP1**), is at most **OPT**(**LP2**). Thus by Equation 3, we have:

$$\mathbf{OPT}(\mathbf{OP1}) \geq \frac{1}{\beta^*} \Pr[\mathcal{V}'^* = 0] = \mathbf{OPT}(\mathbf{LP2}) \geq \mathbf{OPT}(\mathbf{OP1})$$

which implies that $\mathcal{V}^*$, $\mathcal{V}'^*$ and β^* also form an *optimal* solution for **OP1**, and hence **OPT**(**OP1**) and **OPT**(**LP2**) are equal. This also implies that β^* is at most $1/\mathbf{OPT}(\mathbf{OP1})$.

In Appendix E of Wu and Yang (2016a), Wu and Yang proved that an optimal solution of **LP2** can be obtained through the best polynomial approximation of the function $1/x$. More formally, they showed that there exists a solution for **LP2** with the following optimal value:

$$\mathbf{OPT}(\mathbf{LP2}) = 2 \inf_{f \in \mathcal{P}_{L-1}} \sup_{x \in [1+\nu, \lambda]} \left|\frac{1}{x} - f(x)\right|$$

where $\mathcal{P}_{L-1}$ is the set of all degree $L-1$ polynomials. The optimal polynomial approximation error have been studied in Kraus et al. (2012) and in Sec. 2.11.1 of Timan (1963). They computed the maximum error of the best degree $L-1$ polynomial approximation. More precisely, we have:

$$\mathbf{OPT}(\mathbf{OP1}) = \mathbf{OPT}(\mathbf{LP2}) = 2 \inf_{f \in \mathcal{P}_{L-1}} \sup_{x \in [1+\nu, \lambda]} \left|\frac{1}{x} - f(x)\right| = \left(\frac{1}{\sqrt{1+\nu}} - \frac{1}{\sqrt{\lambda}}\right)^2 \left(\frac{\sqrt{\frac{\lambda}{1+\nu}} - 1}{\sqrt{\frac{\lambda}{1+\nu}} + 1}\right)^{L-2} .$$

Hence, the proof is complete. ■

B.3. Proof of Lemma 6

Before stating the lemma, we review the definitions we used so far. Recall that p and p' are generated by drawing n i.i.d samples, V_i's and V_i''s, from P_V and $P_{V'}$ respectively:

$$p = \frac{1}{n}(V_1, V_2, \ldots, V_n) \qquad p' = \frac{1}{n}(V_1', V_2', \ldots, V_n')$$

and E and E' where the desired events:

$$E = \left\{ \left| \sum_{i=1}^{n} \frac{V_i}{n} - 1 \right| \leq \nu, \text{ and } \sum_{i=1}^{n} N_i > s(1-\nu)/2 \right\} .$$

and

$$E' = \left\{ \left| \sum_{i=1}^{n} \frac{V'_i}{n} - 1 \right| \leq \nu \, , \, r \geq \frac{\beta n d}{2}, \text{ and } \sum_{i=1}^{n} N'_i > s(1-\nu)/2 \right\}$$

where r was the number of elements i for which V'_i is zero. We generate histograms h and h' according to p and p' respectively. let $\mathcal{H}$ denote the distribution over histograms h generated by the process when the prior is P_V, and let $\mathcal{H}_E$ be the distribution over histograms h conditioning on E. We define $\mathcal{H}'$ and $\mathcal{H}_{E'}$ similarly. In the following lemma, we prove "good properties" for p and p' after normalization and also bound the total variation distance between $\mathcal{H}_E$ and $\mathcal{H}'_{E'}$.

Lemma 8 *Let $P_V, P_{V'}$, and $\beta \in [1, 1/d]$ form a solution of* **OP1** *with objective value $d \geq 2\epsilon$. Suppose P_V and $P_{V'}$ are the prior distributions to generate histograms h and h' according to the process. Then, h given event E is a histogram of a set of at least $s(1-\nu)/2$ samples from a $1/(\beta n)$-big distribution, whereas h' given E' is a histogram of a set of at least $s(1-\nu)/2$ samples that are drawn from a distribution which is ϵ-far from any $1/(\beta n)$-big distribution. Moreover,*

$$d_{TV}\left(\mathcal{H}_E, \mathcal{H}'_{E'}\right) \leq \frac{2\lambda}{\beta n \nu^2} + \exp\left(-\frac{\beta n d}{8}\right) + 2\exp\left(-\frac{s(1-\nu)}{6}\right) + n\left(\frac{es\lambda}{2nL}\right)^L .$$

Lastly, the largest probability mass among any elements in any probability distributions (from which the samples are drawn) is $\lambda/(n(1-\nu))$.

Proof First, we show given event E, the normalization of p is $1/(\beta n)$-big distribution. From **OP1**, we know that the V_i's are in $[(1+\nu)/\beta, \lambda/\beta]$, and the V'_i's are in $\{0\} \cup [(1+\nu)/\beta, \lambda/\beta]$. Observe that $p(i)$ after normalization is at least the following:

$$\frac{p(i)}{\sum_j p(j)} \geq \frac{V_i/n}{\sum_j V_j/n} \geq \frac{(1+\nu)/(\beta n)}{\sum_j V_j/n} \geq \frac{1}{\beta n}$$

where the last inequality is due to the fact that $\sum_j V_j/n$ is at most $1+\nu$. Thus, the normalization of p is $1/(\beta n)$-big. On the other hand, we can achieve the same lower bound for the normalized value of $p'(i)$ when V'_i is not zero, so the normalization of p' places either probability mass zero, or at least $1/(\beta n)$, on each element. Similarly, the maximum probability mass among the normalization of p's and p''s is at most

$$\frac{p(i)}{\sum_j p(j)} \leq \frac{V_i/n}{\sum_j V_j/n} \leq \frac{\lambda/(\beta n)}{1-\nu} \leq \frac{\lambda}{n(1-\nu)}$$

because $\beta \geq 1$, yielding the desired bound on the maximum probability mass.

Next, we show that given E', the normalization p' is ϵ-far from any big distribution. Note that if V'_i is zero, then probability $p'(i)$ even after normalization remains zero. So, there are exactly r elements that have probability mass zero and the rest (based on above argument) each have probability mass at least $1/(\beta n)$. Thus, the total variation distance to $1/(\beta n)$-bigness is at least $r/(\beta n)$, and given E' it is at least $d/2 \geq \epsilon$.

Now, we show the distance between $\mathcal{H}_E$ and $\mathcal{H}'_{E'}$ is bounded. By the triangle inequality we have:

$$\begin{aligned} d_{TV}\left(\mathcal{H}_E, \mathcal{H}'_{E'}\right) &\leq d_{TV}\left(\mathcal{H}_E, \mathcal{H}\right) + d_{TV}\left(\mathcal{H}, \mathcal{H}'\right) + d_{TV}\left(\mathcal{H}', \mathcal{H}'_{E'}\right) \\ &\leq \Pr[E^c] + d_{TV}\left(\mathcal{H}, \mathcal{H}'\right) + \Pr[E'^c]\,, \end{aligned}$$

where the superscript c for the events, E and E' indicates the complimentary event. Now, we start with bounding the probability of the complementary events of E and E' from above to show that they happen with small probability. Since the V_i's (and similarly the V_i''s) are independently drawn from P_V with expected value 1, and they are in the range $[0, \lambda/\beta]$, then by the Chebyshev inequality, we have:

$$\Pr\left[\left|\sum_{i=1}^{n} \frac{V_i}{n} - 1\right| > \nu\right] \leq \frac{\sum_i \mathbf{Var}[V_i]}{n^2\nu^2} \leq \frac{\mathbf{E}[V^2]}{n\nu^2} \leq \frac{\lambda \mathbf{E}[V]}{\beta n \nu^2} \leq \frac{\lambda}{\beta n \nu^2}.$$

Recall that the d was the optimal value of **OP1**. Thus, $\Pr[V_i' = 0]$ is βd. Moreover, r, the number of the V_i''s that are zero, is a Binomial random variable with $\mathbf{E}[r] = n \cdot \Pr[V_i' = 0]$ which is $\beta n d$. Thus, by the Chernoff bound, we have:

$$\Pr\left[r < \frac{\beta n d}{2}\right] = \Pr\left[\frac{r}{n} < \beta d\left(1 - \frac{1}{2}\right)\right] \leq \exp\left(-\frac{\beta n d}{8}\right).$$

Finally, we show that the total number of samples is high with high probability. Assume we already have $\sum_{i=1}^{n} V_i/n$ is at least $1-\nu$. Then the total number of samples $\sum_{i=1}^{n} h_i$ is a Poisson random variable with mean $t := s\sum_{i=1}^{n} V_i \geq s(1-\nu)$. By the tail bound for Poisson distributions proved in Canonne (2017)[2], we have

$$\begin{aligned} \Pr\left[\sum_{i=1}^{n} h_i \leq \frac{s(1-\nu)}{2}\right] &\leq \Pr\left[\sum_{i=1}^{n} h_i \leq t - t/2\right] \leq \exp\left(-\frac{(t/2)^2}{t + t/2}\right) \\ &\leq \exp\left(-\frac{t}{6}\right) \leq \exp\left(-\frac{s(1-\nu)}{6}\right) \end{aligned}$$

One can achieve a similar result for $\sum_{i=1}^{n} h_i'$.

Now, we continue bounding the distance between $\mathcal{H}_E$ and $\mathcal{H}'_{E'}$. $\mathcal{H}^{(i)}$ (and similarly $\mathcal{H}'^{(i)}$) indicates the distribution over the i-th coordinate of the histogram, h_i. By the previous inequality, we have:

$$\begin{aligned} d_{TV}\left(\mathcal{H}_E, \mathcal{H}'_{E'}\right) &\leq \Pr[E^c] + d_{TV}\left(\mathcal{H}, \mathcal{H}'\right) + \Pr[E'^c] \\ &\leq \Pr[E^c] + \Pr[E'^c] + n \cdot d_{TV}\left(\mathcal{H}^{(i)}, \mathcal{H}'^{(i)}\right) \\ &\leq \frac{2\lambda}{\beta n \nu^2} + \exp\left(-\frac{\beta n \epsilon}{8}\right) + 2\exp\left(\frac{s(1-\nu)}{6}\right) + n\left(\frac{es\lambda}{2nL}\right)^L, \end{aligned}$$

[2] If X is a Poisson random variable with mean λ, then for any $t > 0$, we have $\Pr\left[X \leq \lambda - t\right] \leq \exp\left(-\frac{t^2}{\lambda + t}\right)$

where the last inequality follows from the fact that the first L moments of P_V and $P_{V'}$ are matched, by Lemma 6 in Wu and Yang (2016b), we have:

$$d_{TV}\left(\mathcal{N}_V^{(i)}, \mathcal{N}_{V'}^{(i)}\right) \leq \left(\frac{es\lambda}{2nL}\right)^L$$

Hence, the proof is complete. ■

Appendix C. From Bigness to Monotonicity

In this section, we show how to turn our lower bound results for bigness testing problem in the previous section, into lower bounds for monotonicity testing in some fundamental posets, namely the matching poset and the Boolean hypercube poset. See Subsection 3.2 for the proof overviews.

C.1. Monotonicity testing on a matching poset

Theorem 9 *Consider the pair of distributions $\mathcal{N}^+$, $\mathcal{N}^-$ for the bigness problem as specified in Theorem 3 with bigness threshold $T = O(1/n)$, number of samples s, and maximum probability p_{max}. There exists a distribution on a matching of size n with maximum probability $p'_{max} = \Theta(p_{max})$ such that testing, with success probability $2/3$, whether a matching randomly drawn from such a distribution is monotone or $\epsilon' = \Theta(\epsilon)$-far from any monotone distribution, requires $s' = \Omega(s)$ samples.*

Proof Let $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$ form the vertex set of a directed matching M_n of size n where the edges are (v_i, u_i)'s for $i = 1, 2, \ldots, n$. Consider the distribution over the matching poset $G = (U \cup V, \{(v_i, u_i) | i \in [n]\})$; more specifically, the distribution is monotone if and only if the probabilities $p(u_i) \geq p(v_i)$ for all i. We apply the Poissonization technique, then prove our lower bound by contradiction: assume there exist an algorithm $\mathcal{A}$ which tests monotonicity of distributions over the matching of size n using $\mathbf{Poi}(s')$ samples where $s' = o(s)$ and successfully distinguishes whether the distribution is monotone or $\epsilon' := \epsilon/(2(1+nT))$-far from monotone with probability at least $2/3$. To reach the desired contradiction, we turn these samples into $s'(1+nT)$ samples for the T-bigness testing problem, and show that one can test T-bigness using $\mathcal{A}$ as a black-box tester. Note that $T = O(1/n)$, so the factor $1 + nT$ is $\Theta(1)$ in this proof.

Assume we have a distribution, p, over $[n]$ elements for which we wish to test the bigness property. We construct a distribution q_p over a matching over $U \cup V$ based on p as follows:

$$q_p(u_i) = \frac{p(i)}{1+nT}, \quad q_p(v_i) = \frac{T}{1+nT}.$$

Clearly the maximum probability of q_p is at most $p'_{\max} := p_{\max}/(1 + nT)$. Next we show the changes in distances to monotonicity. Next we show the difference in distance to monotonicity from the case that p is T-big and the case that p is ϵ-far from T-big. If p is a T-big distribution, then $q_P(u_i) \geq T/(1 + nT) \geq q_p(v_i)$ and thus q_p is monotone.

Next, if p is ϵ-far from any T-big distribution, then we show that q_p is $\epsilon/(2(1+nT))$-far from any monotone distribution. Let S be the set of elements for which $p(i) < T$. Clearly, to make p a T-big distribution, one has to increase all the $p(i)$ to T for $i \in S$ and there is no need to increase the probability of any other elements. Therefore, the total variation distance to of p to $\mathsf{Big}(n)$ is exactly $\sum_{i \in S} T - p(i)$ assuming $T \leq 1/n$. Let q' be the closest monotone distribution to q_p, and observe that $q'(u_i) \geq q'(v_i)$. We compute:

$$\begin{aligned}
d_{TV}(q_p, \mathsf{Mon}(M_n)) = d_{TV}(q_p, q') &= \frac{1}{2}\sum_{i=1}^{n} |q_p(u_i) - q'(u_i)| + |q_p(v_i) - q'(v_i)| \\
&= \frac{1}{2}\sum_{i=1}^{n} \left|\frac{p(i)}{1+nT} - q'(u_i)\right| + \left|\frac{T}{1+nT} - q'(v_i)\right| \\
&\geq \frac{1}{2}\sum_{i=1}^{n} \max\left(0, q'(u_i) - \frac{p(i)}{1+nT} + \frac{T}{1+nT} - q'(v_i)\right) \\
&\geq \frac{1}{2}\sum_{i=1}^{n} \max\left(0, \frac{T-p(i)}{1+nT}\right) \geq \frac{1}{2(1+nT)}\sum_{i \in S} T - p(i) \\
&= \frac{d_{TV}(p, \mathsf{Big}(n))}{2(1+nT)} = \frac{\epsilon}{2(1+nT)} .
\end{aligned}$$

Finally we show that the assumed algorithm $\mathcal{A}$ may be used to test the T-bigness property of p. Suppose we are given access to $\mathbf{Poi}(s')$ independent samples from the distribution p for which we want to test T-bigness property. We construct a distribution q_p as described above: to obtain $\mathbf{Poi}(s'(1+nT))$ samples from q_p, for each $i \in [n]$, we create $\mathbf{Poi}(s' \cdot p(i))$ and $\mathbf{Poi}(s' \cdot T)$ samples of u_i and v_i respectively. The $\mathbf{Poi}(s' \cdot p(i))$ samples for each i of the u_i's may be obtained by substituting each element i from p with u_i in $\mathbf{Poi}(s')$ samples from p, whereas $\mathbf{Poi}(s' \cdot T)$ samples for v_i's may be generated directly by drawing v_i's uniformly at random. Thus, using $\mathbf{Poi}(s')$ samples from p, one can construct $\mathbf{Poi}(\Omega(s'))$ samples from q_p and use $\mathcal{A}$ for testing the monotonicity of the matching poset q_p, which corresponds to testing the T-bigness of p, yielding a contradiction by the fact that bigness testing requires $\Omega(s)$ samples by Theorem 3. ■

This result, applied with Theorem 3 using $L = \Theta(\log n)$ (where $s = \Omega\left(n \ln^2(1/\epsilon)/\log n\right)$, $p_{\max} = O((\log^2 n)/(n \ln^2(1/\epsilon)))$ and $T = 1/(\beta n) \in [\epsilon/n, 1/n]$), immediately yields the following lower bound for the testing monotonicity in a matching poset.

Corollary 10 *For sufficiently small parameter $\epsilon = \Omega(1/n)$, any algorithm that can distinguish whether a distribution over a matching poset on $2n$ vertices is monotone, or ϵ-far from any monotone distribution, with probability $2/3$ requires $\Omega((n \ln^2(1/\epsilon))/\log n)$ samples. Moreover, the maximum probability mass of the distribution in the lower bound construction can be bounded above by $O((\log^2 n)/(n \ln^2(1/\epsilon)))$.*

C.2. Monotonicity testing on a hypercube poset

Consider the Boolean hypercube poset $\{0,1\}^d$ with $N = 2^d$ vertices. For convenience, let $\mathcal{C}$ and $\mathcal{S}$ denote the distribution of distributions implicitly constructed in the lower bound of Theorem 9, where distributions in $\mathcal{C}$ are monotone, and distributions in $\mathcal{S}$ are ϵ-far from any monotone distribution, respectively. Theorem 9 shows that randomly-drawn distributions from $\mathcal{C}$ and $\mathcal{S}$ generate statistically similar histograms over the matching poset. For simplicity, we do not distinguish the parameters ϵ, s and $p_{\max}$ in Theorem 3 and Theorem 9 as they are equivalent up to a constant factor.

C.2.1. General lower bound for monotonicity testing on a hypercube poset

We first establish the theorem that describes the result of the outlined embedding approach, then later apply this result to achieve interesting special cases.

Theorem 11 *Let an integer $\ell \geq 1$ be a parameter. Suppose that there exists a pair $(\mathcal{C}, \mathcal{S})$ of distribution of distributions over a matching on $n = \binom{d-1}{\ell-1}$ pairs of vertices, forming an instance for the monotonicity problem with distance ϵ, a maximum probability p_{max}, and a lower bound of s samples. Then, testing monotonicity on the Boolean hypercube of size $N = 2^d$ with distance parameter ϵ/W requires $\Omega(sW)$ samples, where $s = \Omega((n \ln^2(1/\epsilon))/\log n)$ and $W = 1 + \Theta((\log^2 n)/(n \ln^2(1/\epsilon))) \cdot \left(\sum_{i=\ell}^{d} \binom{d}{i} - \binom{d-1}{\ell-1}\right)$.*

Proof Consider two consecutive levels ℓ and $\ell - 1$ of a hypercube, where the ℓ^{th} level consists of vertices whose coordinates contain exactly ℓ ones. Our approach is to embed our matching onto these levels in the hypercube, so that each edge of the matching has one endpoint in each of the two levels, and each endpoint is mutually incomparable to any endpoint of any other edge.

We choose our coordinates for the embedding as follows. We pick all the vertices such that there are exactly $\ell - 1$ ones among the first $d - 1$ coordinates. Let M denote the set of these vertices. There are exactly $2 \cdot \binom{d-1}{\ell-1}$ vertices in the set M. Clearly, each vertex in M is comparable with the vertex whose coordinate only differs at the last bit. Furthermore, it is incomparable with the rest of the vertices in M, as other coordinates also have $\ell - 1$ ones on the first $d - 1$ bits.

Next we describe the probabilities assigned to each vertex on the hypercube, given p, the distribution over a matching (drawn from $\mathcal{C}$ or $\mathcal{S}$). First we assign the probabilities to M according to p. Namely, the set of coordinates of M with ℓ ones corresponds to U and that with $\ell - 1$ ones corresponds V, where U and V are as defined in the previous proof. Then, for the remaining vertices in level ℓ and above, assign the probability of $c \cdot ((\log^2 n)/(n \ln^2(1/\epsilon)))$ for a sufficiently large c such that the quantity becomes at least $p_{\max}$. Let $W = 1 + \Theta((\log^2 n)/(n \ln^2(1/\epsilon))) \cdot \left(\sum_{i=\ell}^{d} \binom{d}{i} - \binom{d-1}{\ell-1}\right)$ be the total probability assigned to all these vertices so far. We divide all assigned probabilities by W to finally obtain a distribution over the hypercube. We denote the constructed distribution over the hypercube p_H.

Clearly, the proposed construction preserves the monotonicity due to the incomparability between distinct embedded matching edges. In particular, if distribution over the matching is drawn from $\mathcal{C}$, the distribution over the hypercube will still be monotone; if it is drawn from $\mathcal{S}$, then the distance to monotonicity is now ϵ/W since, at the very least, the subposet restricted to the embedded matching must be modified to a monotone distribution over this matching.

Using Corollary 10, any algorithm that can test the monotonicity of p_H requires $\Omega(s)$ samples from the matching vertices. Note that if we draw a sample from p_H with probability

$1/W$ it is from the matching. Therefore, observe that $\mathbf{Poi}(s)$ samples from the matching are required in order to obtain $\mathbf{Poi}(sW)$ samples from the hypercube with high probability. This yields the lower bound of $\Omega(sW)$ samples for testing monotonicity over the hypercube poset. ■

C.2.2. Applications of Theorem 11

We extend Theorem 11 into two following corollaries. Firstly, we consider embedding our matching to the largest possible levels of the hypercube, namely the middle ones, showing the lower bound of $\Omega(nd)$ samples for $\epsilon = \Theta(1/d^{2.5})$ (Corollary 12). To complement this first corollary that only handles sub-constant ϵ, we secondly apply our construction to higher levels of the hypercube, and readjust the construction from Theorem 3 so that $L = \Theta(1)$ moments are matched (as opposed to $\Theta(\log n)$). This approach shows the lower bound of $\Omega(N^{1-\delta})$ for testing monotonicity on the hypercube poset with distance parameter ϵ, such that $\delta \to 0$ as $\epsilon \to 0$ (Corollary 13).

Corollary 12 *For sufficiently small $\epsilon = \Theta(1/d^{2.5})$, any algorithm that can distinguish whether a distribution over a Boolean hypercube poset of size $N = 2^d$ is monotone, or ϵ-far from any monotone distribution, with success probability $2/3$ requires $\Omega(Nd)$ samples.*

Proof Let ℓ be $\lceil d/2 \rceil$. As we stated in the proof of Theorem 11, we embed a matching of size $n := \binom{d-1}{\ell-1}$ onto the middle layer of the hypercube where n is at least $\Omega(N/\sqrt{d}) = \Omega(N/\sqrt{\log N})$ by Stirling's approximation. We have

$$W = 1 + \Theta(d^{2.5}/(N \log^2(1/\epsilon'))) \cdot \Theta(N) = \Theta(d^{2.5}/\log^2(1/\epsilon'))\,.$$

Applying Theorem 11, we achieve our lower bound of $\Omega(Nd)$ for $\epsilon = \Theta(1/d^{2.5})$ by choosing a sufficiently small constant ϵ'. ■

Corollary 13 *Any algorithm that can distinguish whether a distribution over a Boolean hypercube poset of size $N = 2^d$ is monotone, or ϵ-far from any monotone distribution, with success probability $2/3$ requires $\Omega(N^{1-\delta})$ samples, where ϵ and $\delta = \Theta(\sqrt{\epsilon}) + o(1)$ are constants. In particular, $\delta \to 0$ as $\epsilon \to 0$.*

Proof Without loss of generality assume d is even. Otherwise, observe that when d is odd, we may embed a hypercube of size 2^d in a hypercube of size 2^{d+1} and achieve the same lower bound up to a constant factor. Consider $\ell \geq d/2$. Observe that

$$\binom{d}{\ell+i} = \binom{d}{\ell} \cdot \frac{d-\ell}{\ell+1} \cdot \frac{d-\ell-1}{\ell+2} \cdots \frac{d-\ell-i+1}{\ell+i} \leq \binom{d}{\ell}\left(\frac{d-\ell}{\ell+1}\right)^i .$$

This yields the inequality

$$\sum_{i=\ell}^{d} \binom{d}{i} = \binom{d}{\ell} \sum_{i=0}^{d-\ell} \left(\frac{d-\ell}{\ell+1}\right)^i \leq \binom{d}{\ell} \sum_{i=0}^{\infty} \left(\frac{d-\ell}{\ell+1}\right)^i = \binom{d}{\ell} \frac{\ell+1}{2\ell-d+1}.$$

We pick $\ell = d/2 + \alpha d$ for some constant $0.24 > \alpha > 0$ so that $\sum_{i=\ell}^{d} \binom{d}{i} = \Theta\left(\binom{d}{\ell}\right)$. The embedded matching is of size $n = \binom{d-1}{\ell-1} = \frac{d}{\ell}\binom{d}{\ell} = \Theta\left(\binom{d}{\ell}\right)$.

Next, consider the application of Theorem 9 leveraging Theorem 3 with constant parameters ϵ and L, yielding the lower bound of $s = \Omega(n^{1-1/L}/L)$ samples for $p_{\max} = O(L^2/n) = O(L^2/\binom{d}{\ell})$. We compute $W = 1 + \Theta\left(L^2/\binom{d}{\ell}\right) \cdot \Theta\left(\binom{d}{\ell}\right) = \Theta(L^2)$. Applying Theorem 11, we achieve the lower bound of $\Omega(n^{1-\frac{1}{L}}L)$ for testing monotonicity over the hypercube with $\epsilon = \Theta(1/L^2)$.

Recall that $\ell = d/2 + \alpha d$. Using a similar argument as above, we can also bound

$$\begin{aligned} n = \binom{d}{\ell} &\geq \binom{d}{d/2} \cdot \frac{d/2-1}{d/2+1} \cdots \frac{d/2-\alpha d}{d/2+\alpha d} \geq \binom{d}{d/2}\left(\frac{d/2-\alpha d}{d/2+\alpha d}\right)^{\alpha d} \\ &\geq \binom{d}{d/2}(1-4\alpha)^{\alpha d} \geq \frac{N}{\sqrt{2d}}(1-4\alpha)^{\alpha \log N} = \frac{N^{1+\alpha\log(1-4\alpha)}}{\sqrt{2d}}, \end{aligned}$$

establishing the lower bound of $\widetilde{\Omega}(N^{(1+\alpha\log(1-4\alpha))(1-\frac{1}{L})}) = \Omega(N^{1-\delta})$ for testing monotonicity over the hypercube poset, where $\delta = 1/L - \alpha\log(1-4\alpha) + o(1)$. Since $\epsilon = \Theta(1/L^2)$, for sufficiently large N, we may choose sufficiently small α and large L, so that $\delta = \Theta(\sqrt{\epsilon}) + o(1)$, as desired. ■

Appendix D. Reduction from General Posets to Bipartite Graphs

In this section, we show that the problem of monotonicity testing of distributions over the *bipartite* posets is essentially the "hardest" case of monotonicity testing in general poset domains. That is, we show that for any distribution p over some poset domain of size n, represented as a directed graph G, there exists a distribution p' over a bipartite poset G' of size $2n$ such that (1) p preserves the total variation distance of p to monotonicity up to a small multiplicative constant factor, and (2) each sample for p' can be generated using one sample drawn from p. These properties together imply the following main theorem of this section.

Theorem 14 *Suppose that there exists an algorithm that tests monotonicity of a distribution over a bipartite poset domain of n elements using $s(n, \epsilon)$ samples for any total variation distance parameter $\epsilon > 0$. Then, there exists an algorithm that tests monotonicity of a distribution over any poset domain of n elements using $O(s(2n, \epsilon/4))$ samples.*

Proof Consider an arbitrary poset described as a directed graph $G = (V, E)$, and an associated probability distribution p over V. We construct a bipartite graph $G' = (V', E')$ based on the transitive closure of G, denoted by $TC(G)$, and a distribution p' over V' such that testing the monotonicity of p over V is roughly equivalent to testing the monotonicity of p' over V.

The construction of the bipartite $G' = (V', E')$ is as follows: for each $v \in V$, we add two vertices v^+ and v^- to V', so that $S := \{v^+\}_{v \in V}$ and $T := \{v^-\}_{v \in V}$ together form the bipartition $V' := S \cup T$. Think of S and T as the set of top and bottom vertices respectively.

Next, consider two vertices u and v such that there is a path from u to v in G (i.e., (u,v) is an edge in $TC(G)$). For every such pair, we add the directed edge (u^-, v^+) to E'. Given the distribution p over V, we set $p'(v^+) = p'(v^-) = p(v)/2$. Observe that we can generate a sample from p' using a sample from p: if v is drawn from p, a sample for p' is obtained by picking either v^+ or v^-, each with probability $1/2$.

Now, we prove that testing monotonicity of p is equivalent to testing monotonicity of p'. If p is monotone, then p' is also monotone: for each $(u^-, v^+) \in E'$, $p(u) \leq p(v)$ via the transitivity of monotonicity of p along the u-v path on G. So, $p'(u^-) = p(u)/2 \leq p(v)/2 = p'(v^+)$.

Next, suppose p is ϵ-far from p'. By Lemma 16 (shown below), there exists a (directed) matching M in $TC(G)$, such that

$$\sum_{(u,v)\in M} p(u) - p(v) \geq d_{TV}(\mathsf{Mon}(G), p) \geq \epsilon\,. \tag{4}$$

Then, the set of edges (u^-, v^+)'s corresponding to $(u,v) \in M$ also forms a matching, M', on G'. Let p'^* be the monotone distribution on G' closest to p'. Since p'^* is a monotone distribution, for an edge (u^-, v^+), $p'^*(v^+)$ is at least $p'^*(u^-)$. Then, by the triangle inequality, we obtain:

$$\begin{aligned}
d_{TV}(\mathsf{Mon}(G'), p') &= \tfrac{1}{2} \cdot |p' - p'^*| = \tfrac{1}{2} \sum_{v\in V} |p'(v^-) - p'^*(v^-)| + |p'(v^+) - p'^*(v^+)| \\
&\geq \tfrac{1}{2} \sum_{(u^-,v^+)\in M'} |p'(u^-) - p'^*(u^-)| + |p'(v^+) - p'^*(v^+)| \\
&\geq \tfrac{1}{2} \sum_{(u^-,v^+)\in M'} p'(u^-) - p'^*(u^-) - p'(v^+) + p'^*(v^+) \\
&= \tfrac{1}{2} \sum_{(u^-,v^+)\in M'} p'(u^-) - p'(v^+) + (p'^*(v^+) - p'^*(u^-)) \\
&\geq \tfrac{1}{2} \sum_{(u^-,v^+)\in M'} p'(u^-) - p'(v^+) \\
&\geq \tfrac{1}{2} \sum_{(u,v)\in M} (p(u) - p(v))/2 \geq \epsilon/4.
\end{aligned}$$

Note that the second to last inequality is true since p'^* is monotone, and $p'^*(v^+)$ has to be at least $p'^*(u^-)$. Therefore, if p is ϵ-far from monotone, then p' is $\epsilon/4$-far from monotone.

Thus, to distinguish whether p is monotone or ϵ-far from any monotone distribution on G, it is suffices to test if p' is monotone or $\epsilon/4$-far from any monotone distribution on the bipartite poset G'. ■

An interesting byproduct of Equation 4 is the following: If you consider the violation of each edge from monotonicity to be the weight of that edge, then the weight of the maximum weighted matching is the distance of the distribution to monotonicity. We formally explained it in the following theorem.

Theorem 15 *Consider a poset $G = (E, V)$ and a distribution p over its vertices. Suppose every edge (u, v) in the $TC(G)$ has a weight of $\max(0, p(u)-p(v))$. Then, the total variation distance of p to any monotone distribution is within a factor of two of the weight of the maximum weighted matching in $TC(G)$.*

Proof Let W indicates the weight of the maximum weighted matching. Fix a matching M of k edges (u_i, v_i). Assume p' is the closest monotone distribution to p, so $p'(u_i) \leq p'(v_i)$ for every edge (u_i, v_i). One can show the following:

$$\begin{aligned} d_{TV}(\mathsf{Mon}(G), p) &= \frac{1}{2} \cdot \|p - p'\|_1 = \frac{1}{2} \sum_{(u_i,v_i)\in M} |p(u_i) - p'(u_i)| + |p(v_i) - p'(v_i)| \\ &\geq \frac{1}{2} \sum_{(u_i,v_i)\in M} \max\big(0, p(u_i) - p(v_i) + p'(v_i) - p'(u_i)\big) \\ &\geq \frac{1}{2} \sum_{(u_i,v_i)\in M} \max\left(0, p(u_i) - p(v_i)\right) \geq \frac{1}{2} W \end{aligned}$$

where the last inequality is true, because the above is true for any matching M. On the other hand by Lemma 16, there exists a (directed) matching M_0 in $TC(G)$, such that

$$d_{TV}(\mathsf{Mon}(G), p) \leq \sum_{(u_i,v_i)\in M^*} p(u_i) - p(v_i) \leq W\,.$$

Thus, the proof is complete. ■

D.1. Proof of auxiliary lemmas

Lemma 16 *Let p be a probability distribution over the vertex set V of an unweighted directed graph $G = (V, E)$ representing a poset. Then, there exists a matching M on the transitive closure $TC(G)$ such that*

$$\sum_{(u,v)\in M} p(u) - p(v) \geq d_{TV}(p, \mathsf{Mon}(G))\,.$$

Proof Define ϵ to be the ℓ_1-distance of p to monotonicity. We need to show the following:

$$\sum_{(u,v)\in M} p(u) - p(v) \geq \epsilon/2\,.$$

Let f^* be the monotone *function* on G closest to p (in the ℓ_1-distance). Let d denote $\|f^* - p\|_1$: the ℓ_1-distance between f^* and p. Note that f^* is not necessarily a probability distribution which implies that d can be smaller than ϵ. To prove the above inequality, we will use d as an intermediate variable which is in between the left hand side and the right hand side of the above inequality. Specifically, it suffices to prove the following:

(i) $d \geq \epsilon/2$;

(ii) there exists a matching M on the transitive closure of G such that $\sum_{(u,v)\in M} p(u) - p(v) = d$.

Proof of Item (i): To show that d is at least $\epsilon/2$, we prove that the monotone distribution p_{f^*}, obtained by normalizing f^*, is at most $2d$-far from p. Since any monotone distribution is at least ϵ-far from p in ℓ_1-distance , we will have $\epsilon \leq \|p - p_{f^*}\|_1 \leq 2d$, establishing the desired claim.

First, note that if $f^*(v)$ is zero for all v, then by definition d is at least $\epsilon/2$:

$$d = \sum_{v \in V} |p(v) - f^*(v)| = \sum_{v \in V} |p(v)| = 1 \geq \epsilon/2$$

where the inequality holds since the ℓ_1-distance between two distributions is always at most 2, so ϵ is as well. Hence, assume f^* is not a zero function for the rest of the proof.

Also, note that f^* is a non-negative function. We prove the non-negativity of f^* by contradiction: assume $f^*(v)$ is negative for some v. Consider a non-negative function $f(v) = \max\{f^*(v), 0\}$. It is not hard to see that f is monotone due to monotonicity of f^*. For every v for which $f^*(v) < 0$, we have

$$|p(v) - f(v)| = p(v) - 0 < p(v) - f^*(v) = |p(v) - f^*(v)| \ .$$

Since $f^*(v) = f(v)$ everywhere else, $\|p - f\|_1 = \sum_{v \in V} |p(v) - f(v)| < \sum_{v \in V} |p(v) - f^*(v)| = \|p - f^*\|_1$ when f^* contains some negative entry. This contradicts the fact that f^* was the closest monotone function to p, hence $f^*(v)$ has to be non-negative for all v's.

Consider $p_{f^*}(v) = f^*(v)/\sum_u f^*(u)$; it follows that p_{f^*} is a well-defined monotone distribution. Then,

$$\begin{aligned}
\epsilon \leq \|p - p_{f^*}\|_1 &\leq \|p - f^*\|_1 + \|f^* - p_{f^*}\|_1 = d + \sum_{v \in V} \left| f^*(v) - \frac{f^*(v)}{\sum_{u \in V} f^*(u)} \right| \\
&= d + \sum_{v \in V} f^*(v) \cdot \left| \frac{\left(\sum_{u \in V} f^*(u)\right) - 1}{\sum_{u \in V} f^*(u)} \right| = d + \left| \sum_{u \in V} f^*(u) - 1 \right| \\
&= d + \left| \sum_{u \in V} f^*(u) - \sum_{u \in V} p(u) \right| \leq d + \sum_{u \in V} |f^*(u) - p(u)| \\
&= d + \|p - f^*\|_1 = 2d \ .
\end{aligned}$$

Thus, Item (i) is proved.

Proof of Item (ii): We leverage the duality theorem in linear programming. We write an LP that optimizes over all monotone functions f's to find the function f^* closest to p under the ℓ_1-distance. Let $x(v)$ be the variable that indicates the amount of perturbation at vertex x that is needed to make p monotone. For an edge (u, v), the monotonicity constraint requires that $f(v) = p(v) + x(v)$ is at least $f(u) = p(u) + x(u)$, or equivalently,

$$x(v) - x(u) \geq p(u) - p(v) \ .$$

Given this inequality, we can find the monotone function closest to p by solving the following linear program:

$$\begin{aligned}
\textbf{LP3}: \quad &\min \quad \sum_{v \in V} |x(v)| \\
&s.t. \quad x(v) - x(u) \geq p(u) - p(v) \quad \forall (u, v) \in E
\end{aligned}$$

We denote the optimal solution for **LP3** by $x^*(v) \coloneqq f^*(v) - p(v)$, and the corresponding optimal value of the objective function by $d \coloneqq \|p - f^*\|_1$.

To obtain the dual of **LP3**, we write down its standard form by substituting $x(v)$ by $x^+(v) - x^-(v)$ as follows:

$$\begin{aligned} \textbf{LP4}: \quad \min \quad & \sum_{v \in V} x^+(v) + x^-(v) & \\ s.t. \quad & (x^+(v) - x^-(v)) - (x^+(u) - x^-(u)) \geq p(u) - p(v) & \forall (u,v) \in E \\ & x^+(v), x^-(v) \geq 0 & \forall v \in V \ . \end{aligned}$$

Then **LP4** has the following dual:

$$\begin{aligned} \textbf{LP5}: \quad \max \quad & \sum_{(u,v) \in E} (p(u) - p(v)) \cdot y(u,v) & \\ s.t. \quad & \sum_{(u,v) \in E} y(u,v) - \sum_{(v,u) \in E} y(v,u) \leq 1 & \forall v \in V \\ & \sum_{(v,u) \in E} y(v,u) - \sum_{(u,v) \in E} y(u,v) \leq 1 & \forall v \in V \\ & y(u,v) \geq 0 & \forall (u,v) \in E \ . \end{aligned}$$

By strong duality, the optimal value of **LP5** is equal to the optimal value of **LP3**, namely d. On the other hand, the optimal solution of **LP5** can help us to find a matching that satisfies the property in Item ii. Constraints of **LP5** can be viewed in the form of $Ay \leq b$ and $y \geq 0$. Since A is a *totally unimodular matrix* by Lemma 17 (proved below), the LP admits an optimal solution that is also *integral*.

Let y^* denote an integral optimal solution of the **LP5**, and let S be a multi-set of the edges, containing $y^*(u,v)$ copies of edge (u,v). Define the weight of each edge (u,v) as $w(u,v) \coloneqq p(u) - p(v)$, and let the weight of a set S be the sum of the weight of the edges in S. Thus:

$$w(S) \coloneqq \sum_{(u,v) \in S} w(u,v) = \sum_{(u,v) \in S} p(u) - p(v) = \sum_{(u,v) \in E} (p(u) - p(v)) \cdot y^*(u,v) = d \ .$$

We construct a matching M where $w(M) = w(S)$, which completes the proof of Item ii. Based on the constraints of the **LP5**, S forms a subgraph on G (but plausibly with multi-edges) such that the absolute difference between the number of incoming edges and outgoing edges at each vertex is at most one. Hence, we can decompose S to paths and cycles.

Consider a path $P = \langle v_1, v_2, \ldots, v_k \rangle$. Observe that the weight of a path only depends on its endpoints:

$$w(P) = \sum_{i=1}^{k-1} w(v_i, v_{i+1}) = \sum_{i=1}^{k-1} p(v_i) - p(v_{i+1}) = p(v_1) - p(v_k) = w(v_1, v_k) \ .$$

Remark that the edge (v_1, v_k) does not necessarily belong to E, but since v_1 and v_k are endpoints of a path P, then (v_1, v_k) is contained in the *transitive closure* of G.

By the above equation, if we replace the edges of P in S by a single edge (v_1, v_k), then $w(S)$ remains unchanged. We can also remove all cycles without changing $w(S)$ since the weight of a cycle is always zero. Lastly, we may also join paths so that their endpoints are all distinct (since the difference between the in-degree and the out-degree of any vertex is at most one). After this process, we eventually obtain a matching M on the transitive closure of G such that

$$w(M) = \sum_{(u,v)\in M} w(u,v) = w(S) = d\,,$$

concluding the proof of Item (ii) and this lemma. ■

Lemma 17 *The matrix A, namely the coefficient matrix of* **LP5** *when the constraints are written in the form $Ay \le b$ and $y \ge 0$, is a totally unimodular matrix.*

Proof We arrange the rows of A so that the two constraints of each vertex v_i occupy two consecutive rows $2i-1$ and $2i$ for $i = 1, \dots, n$, and that each column j corresponds to the edge $e_j = (u_j, u'_j)$ for $j = 1, \dots, |E|$. Then, each entry of A can be described as follows:

$$A_{i,j} = \begin{cases} 1 & \big(i \equiv 0 \pmod 2 \text{ and } u_j = v_{i/2}\big) \text{ or } \Big(i \equiv 1 \pmod 2 \text{ and } u'_j = v_{(i+1)/2}\Big) \\ -1 & \big(i \equiv 1 \pmod 2 \text{ and } u_j = v_{(i+1)/2}\big) \text{ or } \Big(i \equiv 0 \pmod 2 \text{ and } u'_j = v_{i/2}\Big) \\ 0 & \text{otherwise}\,. \end{cases}$$

To prove that A is a totally unimodular matrix, we make use of the following theorem.

Theorem 18 (Ghouila-Houri Characterization Ghouila-Houri (1962)) *An integral $m \times n$ matrix A is a totally unimodular matrix if and only if, for any non-empty subset of rows, namely R, there exists a disjoint partition of R into R_1 and R_2, such that the following is true.*

$$\sum_{i\in R_1} A_{i,j} - \sum_{i\in R_2} A_{i,j} \in \{0, 1, -1\} \qquad \textit{for } \; j = 1, 2, \dots, n\,. \tag{5}$$

Here, for each non-empty subset $R \subseteq [2n]$, we explicitly define R_1 and R_2 according to the following three conditions. (1) If both $2i-1$ and $2i$ are in R, put both of them in R_1. (2) If only $2i-1$ is in R, then put $2i-1$ in R_1. (3) If only $2i$ is in R, then put $2i$ in R_2.

Consider column j corresponding to $e_j = (v_r, v_{r'})$. This column has four non-zero entries:

$$A_{2r-1,j} = -1, \quad A_{2r,j} = 1, \quad A_{2r'-1,j} = 1, \quad A_{2r',j} = -1\,.$$

If both $2r-1$ and $2r$ appear in R, or both of them are not in R, clearly Equation 5 holds (similarly for $2r'-1$ and $2r'$). Thus, assume that exactly one of two rows $2r-1$ and $2r$, and exactly one of the two rows $2r'-1$ and $2r'$, are in R. It is not hard to see that if the corresponding entries $A_{i,j}$'s in these rows have the same sign, then one row ends up in R_1 and the other row ends up in R_2. If the entries have different signs, then both rows end up in the same set R_1 or R_2. In both of these cases, the sum in Equation 5 becomes zero. Hence, the proof is complete. ■

Appendix E. Algorithms with Sublinear Sample Complexity

In this section, we provide sublinear sample complexity algorithms for testing bigness, and testing monotonicity of distributions over different poset domains. See Subsection 3.4 for proof overviews.

E.1. An Algorithm for Bigness Testing

We give an algorithm for the bigness testing problem that requires a sublinear number of samples. For testing bigness, all the domain elements must be at least a threshold T. The high level idea is to learn the *histogram* of the distribution use a result from Valiant and Valiant (2017). Then given the histogram, if the weight of the elements that are below the threshold is less than $\Theta(\epsilon)$, then we can accept the distribution, otherwise we reject.

First, we define the histogram of a distribution.

Definition 3 *For a distribution p, we define $h_p : (0,1] \to \mathbb{N} \cup \{0\}$ to be the histogram of p if and only if for all $x \in (0,1)$, $h(x)$ is the number of domain element i such that $p(i)$ is equal to x.*

Let $\pi : [n] \to [n]$ be a permutation of the domain elements. We define $p^{(\pi)}$ to be the permutation of p according to π such that for all domain element i, $p^{(\pi)}(i)$ is equal to $p(\pi(i))$. Based on the definition, it is not hard to see permutation does not change the number of domain element with a certain probability, so $h(p)$ and $h(p^{(\pi)})$ are the same. Hence, when we learn the histogram of p, we can claim that we learn p *up to a permutation*.

For learning, we will use a result from Valiant and Valiant (2017) for learning discrete distributions, up to a permutation of the domain elements. In Theorem 1.11 of Valiant and Valiant (2017), combined with Fact 1 of Valiant and Valiant (2016), authors provided the following theorem:

Theorem 19 (Valiant and Valiant (2017, 2016)) *There exists an algorithm that, given $O\left(\frac{n}{\epsilon^2 \log n}\right)$ i.i.d. samples from an unknown distribution p, outputs an explicit description of a distribution, namely q, such that there exists a permutation $\pi : [n] \to [n]$ where $\sum_{i\in[n]} |p(i) - q(\pi(i))| \leq \epsilon$ with success probability $2/3$.*

This theorem implies the following upper bound for bigness testing.

Algorithm 1: Algorithm for Bigness Testing.

```
Bigness-Test($\epsilon$, sample access to $p$)
$\epsilon' \leftarrow \epsilon/3$
$\mathcal{S} \leftarrow$ Draw $O(\frac{n}{\epsilon'^2 \log n})$ samples from $p$
$q \leftarrow$ Learn $p$ (up to a permutation over $[n]$) via Theorem 19 with error parameter $\epsilon'$ using samples in $\mathcal{S}$
if $d_{TV}(q, \mathsf{Big}(n,T)) \leq \epsilon'$ then
    return accept
end
return reject
```

Corollary 20 *For bigness threshold $T \leq 1/n$, there exists an algorithm that distinguishes whether a distribution p is T-big or ϵ-far from T-big with success probability $2/3$ using $O(\frac{n}{\epsilon^2 \log n})$ i.i.d. samples from p.*

Proof We refer to Algorithm E.1 for the outline of our procedure. Let q denote the distribution outputted by the "learner" as promised by Theorem 19 with distance parameter $\epsilon' = \epsilon/3$. Let π be the permutation guaranteed by Theorem 19. We define q' be the distribution obtained by permuting the elements of q according to the associated permutation such that for each domain element i, let $q'(i) = q(\pi(i))$. Hence, with probability at least $2/3$, $d_{TV}(p, q')$ is at most $\leq \epsilon'$. Note that π is not known to the algorithm, but used for the analysis.
Now, we have the following two cases: If p is T-big, then

$$d_{TV}(q', \mathsf{Big}(n,T)) \leq d_{TV}(q', p) \leq \epsilon' = \epsilon/3.$$

On the other hand, if p is ϵ-far from T-big, then

$$d_{TV}(q', \mathsf{Big}(n,T)) \geq d_{TV}(p, \mathsf{Big}(n,T)) - d_{TV}(p, q') \geq \epsilon - \epsilon' \geq 2\epsilon/3.$$

That is, q offers us a condition for T-bigness testing by simply measuring its distance to T-bigness (the **if** condition of Algorithm E.1). Therefore, Algorithm E.1 outputs the correct answer with probability at least $2/3$. Note that learning p using parameter $\epsilon' = \Theta(\epsilon)$ does not change the asymptotic sample complexity, so the proof is complete. ■

E.2. An Algorithm for Testing Monotonicity on Matchings

We give a sublinear time algorithm for testing monotonicity on matchings. Similar to the previous section, we use a result from Valiant and Valiant (2017) for learning the *distribution histogram* of a *pair of distributions*. First we employ the following definitions (see also Definition 5.2 and Definition 5.4 of Valiant and Valiant (2017)). A *distribution histogram* of a pair of distributions is a function that counts the number of elements with a given probability mass x in the distribution p_1 and y in the distribution p_2. More formally, we have the following definition:

Definition 4 (Valiant and Valiant (2017)) *For a pair of distributions p_1 and p_2, we say $h_{p_1,p_2} : [0,1]^2 \setminus \{(0,0)\} \to \mathbb{N} \cup \{0\}$ is the* distribution histogram *of p_1 and p_2 if and only if for any (x,y) in the domain: $h_{p_1,p_2}(x,y) = |\{a : p_1(a) = x, p_2(a) = y\}|$.*

We will use this two-dimensional histogram to indicate a histogram of a distribution over a matching of size n: Let p_1 and p_2 be the two distributions that p imposes on the top and the bottom vertices in the matching respectively. Without loss of generality assume the edges in the matching connects the i-th vertex in the bottom to the i-th vertex in the top. Note that $h_{p_1,p_2}(x,y)$ counts the number of domain elements $a \in [n]$ such that $p_1(a) = x$ and $p_2(a) = y$. Hence, $\int_{x=0}^{1} \int_{y=0}^{1} h_{p_1,p_2}(x,y) dy\, dx$ is the number of matched pairs of vertices with at least one non-zero probability vertex. Since the sum of probabilities according to p_1 is one, we have $\int_{x=0}^{1} \int_{y=0}^{1} x \cdot h(x,y) = 1$. This is similarly true for p_2: $\int_{x=0}^{1} \int_{y=0}^{1} y \cdot h(x,y) = 1$.

Now, we define the distance between two histograms of two distributions: h and g. At a high level, the distance between two histograms is the minimum cost one needs to pay to "transform" h to g. In particular, we transform one histogram to another by moving mass from one point to another: By moving mass c from (x, y) to (x', y'), we obtain another histogram h', such that $h'(x, y) = h(x, y) - c$, $h'(x', y') = h(x, y) + c$ and for all other points in $[0, 1]^2$, h and h' are equal. The cost of this move is $c \cdot (|x - x'| + |y - y'|)$. More formally, we have the following definition.

Definition 5 (Valiant and Valiant (2017)) *For a pair of functions $h, g : [0, 1]^2 \backslash \{(0, 0)\} \to \mathbb{N} \cup \{0\}$, we define the* distance notation *$W(h, g)$ as the minimum cost over all mass moving schemes with finitely many steps for turning h into g, where the cost for moving value $c > 0$ from point (x, y) to (x', y') is $c(|x - x'| + |y - y'|)$. Note that we assume that $\sum_{x,y} h(x, y) = \sum_{x,y} g(x, y)$, where extra value at point $(0, 0)$ on h or g may be added to ensure this equality.*

Let $p^{(\pi)}$ be the *permuted distribution* of p according to the permutation π of $[n]$ such that for each domain element i, ${p'}_1^{(\pi)}(i) = p'_1(\pi(i))$. Note that as long as we permute p_1 and p_2 with the *same* permutation, the distribution histogram h_{p_1,p_2} and $h_{p_1^{(\pi)},p_2^{(\pi)}}$ are the same. Moreover, given h_{p_1,p_2} one can construct q_1 and q_2 such that there exists a permutation π for which q_1 and q_2 are the permuted versions of p_1 and p_2 according to π.
We relate the distance W to the total variation distance in the following Lemma. In particular, the distance W between two distribution histograms h_{p_1,p_2}, $h_{p'_1,p'_2}$ defined according to two pairs of distributions (p_1, p_2), (p'_1, p'_2) upper bounds the ℓ_1-distance up to a permutation of the labels of the domain elements.

Lemma 21 *Let functions h_{p_1,p_2}, $h_{p'_1,p'_2}$ be defined according to two pairs of probability vectors (p_1, p_2), (p'_1, p'_2). There exists a permutation π of $[n]$ such that*

$$W(h_{p_1,p_2}, h_{p'_1,p'_2}) \geq \|p_1 - p_1'^{(\pi)}\|_1 + \|p_2 - p_2'^{(\pi)}\|_1.$$

Proof According to the definition of the distance, W, there exists a moving scheme consisting of a sequence of R steps, denoted by $\langle(c_r, (x_r, y_r), (x'_r, y'_r))\rangle_{r\in[R]}$ (with $c_r > 0$), describing the changes that eventually turn h_{p_1,p_2} into $h_{p'_1,p'_2}$ for which we move the mass of c_r from the source (x_r, y_r) to sink (x'_r, y'_r) at step r. We claim that if the scheme has minimum cost, $W(h_{p_1,p_2}, h_{p'_1,p'_2})$, without loss of generally, we may make the following assumptions about the scheme: (1) There are no two steps r_1 and r_2 such that (x'_{r_1}, y'_{r_1}) is the same as (x_{r_2}, y_{r_2}). (2) All the c_r's are positive integers.
To see why (1) is true, assume otherwise; if $r_1 = r_2$, then $(x'_{r_1}, y'_{r_1}) = (x_{r_2}, y_{r_2})$ means that the source and the sink in step r_1 is the same, so no mass is actually moved. Hence, we can just remove this step without changing the scheme. if $r_1 \neq r_2$, then $(x'_{r_1}, y'_{r_1}) = (x_{r_2}, y_{r_2})$ means that mass of quantity $\min(c_{r_1}, c_{r_2})$ is first moved from (x_{r_1}, y_{r_1}) to (x'_{r_1}, y'_{r_1}), and then moved from (x'_{r_1}, y'_{r_1}) to (x'_{r_2}, y'_{r_2}). Clearly, one can move the same quantity of mass from (x_{r_1}, y_{r_1}) to (x'_{r_2}, y'_{r_2}) directly with no larger cost, making one of the steps r_1 or r_2 vacuous.
Given (1), we now show that (2) also holds: Note that given (1), each point (x, y) may

appear in the steps as either a source or a sink, but not both. Moreover, the order of the steps does not matter, since the source always has the capacity for providing the mass. If there are several steps that move mass between the same source and the same sink, one can replace all of them with one step moving the total quantity of mass moved between them. Now, we can assume between each source and each sink there is a well defined quantity indicating how much mass we moved from the source to sink. This fact helps us to form a graph where the vertices are the sources and the sinks which appeared in the scheme. We put a directed edge from a source to a sink if we moved a non-integer mass from the source to the sink. We assign a weight to the edge which is the fractional part of the mass we moved from the source to the sink. We propose the following process for changing the steps for which each change removes at least one edge from the graph. We keep repeating the process until no edge remains to assure that all c_r's are integers.
Remove sources or sinks with no edge. Clearly, the graph is bipartite, and all the edges are from sources to sinks. Since h_{p_1,p_2} and $h_{p'_1,p'_2}$ are integer, the final mass at each source and sink will eventually be an integer. Hence, each source has an out-degree of at least two and each sink has an in-degree of at least two. Therefore, the graph has an undirected cycle with an even length. Let S and T be the sets of the sources and the sinks involved in the cycle respectively. Let E_1 and E_2 be a partition of the edges in the cycle such that every other edge is in the same set. Clearly, each source (and sink) has exactly one edge in E_1 and one edge in E_2. As we define before the cost of moving one unit of mass via an edge from (x, y) to (x', y') is $|x - x'| + |y - y'|$. We define the cost of E_1 (and E_2) to be the total cost of edges in E_1 (and E_2). Without loss of generality assume cost of E_1 is not greater than the cost of E_2. Let c^* be the minimum weight of edges in E_2. We modify the steps such that each step with a corresponding edge is E_2 moves c^* less mass, and each steps with a corresponding edge in E_2 moves c^* more mass. Clearly, this process does not increase the total cost of the scheme. However, it makes the fractional part of at least one step equal to zero. We repeat this process until no such step exists which concludes the proof for claiming (2).

Let $h^{(0)}, h^{(1)}, \ldots, h^{(R)}$ be the series of the distribution histograms which is generated during the mass moving scheme after each step. $h^{(0)}$ is the distribution histogram we start with, h_{p_1,p_2}, and $h^{(R)}$ is the final distribution histogram $h_{p'_1,p'_2}$. Now, we create a sequence of pairs of *vectors* $p_1^{(r)}, p_2^{(r)} : [n] \to [0, 1]$ such that $h^{(r)} = h_{p_1^{(r)},p_2^{(r)}}$ (under the same definition of distribution histogram, relaxed to allow non-distributions $p_1^{(r)}, p_2^{(r)}$). We start off with $p_1^{(0)}$ and $p_2^{(0)}$ being p_1 and p_2. Given $p_1^{(r-1)}, p_2^{(r-1)}$, we obtain $p_1^{(r)}, p_2^{(r)}$ as follows.

Consider step r described as $(c_r, (x_r, y_r), (x'_r, y'_r))$ with an integer c_r. Inductively, assume $h^{(r-1)} = h_{p_1^{(r-1)},p_2^{(r-1)}}$ which implies that $p_1^{(r-1)}$ and $p_2^{(r-1)}$ contain at least $c_r \le h^{(r-1)}(x_r, y_r)$ entries i with $p_1^{(r-1)}(i) = x_r$ and $p_2^{(r-1)}(i) = y_r$. To apply step r, we pick an arbitrary set I_r of c_r many such entries, then modify the entries $p_1^{(r-1)}(i)$ and $p_2^{(r-1)}(i)$ from x_r and y_r to x'_r and y'_r respectively for each $i \in I_r$. That is, $p_1^{(r)}(i) = x'_r$ and $p_2^{(r)}(i) = y'_r$ for $i \in I_r$, and $p_1^{(r)}(i) = p_1^{(r-1)}(i)$ and $p_2^{(r)}(i) = p_2^{(r-1)}(i)$ for $i \notin I_r$. Hence, the ℓ_1-*distance* incurred by step r becomes:

$$\begin{aligned}\|p_1^{(r-1)} - p_1^r\|_1 + \|p_2^{(r-1)} - p_2^r\|_1 &= \sum_{i\in[n]} |p_1^{(r-1)}(i) - p_1^{(r)}(i)| + \sum_{i\in[n]} |p_2^{(r-1)}(i) - p_2^{(r)}(i)| \\ &= \sum_{i\in I_r} |p_1^{(r-1)}(i) - p_1^{(r)}(i)| + \sum_{i\in I_r} |p_2^{(r-1)}(i) - p_2^{(r)}(i)| \\ &= c_r|x_r - x'_r| + c_r|y_r - y'_r|\,.\end{aligned}$$

By summing over all R steps, and applying the triangle inequality, we have:

$$\begin{aligned}\|p_1 - p_1^R\|_1 + \|p_2 - p_2^R\|_1 &\leq \sum_{r\in[R]} \|p_1^{(r-1)} - p_1^r\|_1 + \sum_{r\in[R]} \|p_2^{(r-1)} - p_2^r\|_1 \\ &= \sum_{r\in[R]} c_r|x_r - x'_r| + c_r|y_r - y'_r| \\ &= W(h^{(0)}, h^{(R)}) = W(h_{p_1,p_2}, h_{p_1^{(R)},p_2^{(R)}})\,.\end{aligned}$$

Now it remains to show that there exists a permutation π that maps the labels of the given distribution p'_1, p'_2 to our constructed vectors $p_1^{(R)}, p_2^{(R)}$; namely, $p_1'^{(\pi)} = p_1^{(R)}$ and $p_2'^{(\pi)} = p_2^{(R)}$. Indeed, $h_{p'_1,p'_2}$ is the distribution histogram that counts the number of indices i with $p'_1(i) = x$ and $p'_2(i) = y$, so $h_{p'_1,p'_2} = h_{p_1^R,p_2^R}$ implies that for every (x, y), there are also equally many indices i' with $p_1^R(i') = x$ and $p_2^R(i') = y$. Hence, there exists a bijection between their indices that maps i''s to i's and vice versa, concluding the lemma. ∎

Next, we state the the result of Valiant and Valiant (2017) to learn the distribution histogram of a pair of distributions.

Theorem 22 (Theorem 5.6 of Valiant and Valiant (2017)) *There exists an algorithm that, given $O\left(\frac{n}{\epsilon^2 \log n}\right)$ i.i.d. samples each from a pair of unknown distributions p_1 and p_2, outputs a function g such that $W(h_{p_1,p_2}, g) \leq \epsilon$ with success probability $2/3$.*

We now prove the upper bound for the monotonicity testing problem over the matching poset.

Theorem 23 *For sufficiently small positive constant ϵ, there exists an algorithm that distinguishes whether a distribution p over the vertex set $V = S \cup T$ of a directed matching M_n on $2n$ vertices is monotone or ϵ-far from monotone with success probability $2/3$ using $O(\frac{n}{\epsilon^2 \log n})$ i.i.d. samples from p.*

Proof For clarity, denote the edge set of the graph $G = (V, E)$ with the set of edges $E = \{(u_i, v_i)\}_{i\in[n]}$, and the set of vertices $V = S \cup T$ where $S = \{u_i\}_{i\in[n]}$ and $T = \{v_i\}_{i\in[n]}$. For a distribution p over $V = S \cup T$, let p_S and p_T denote the probability mass p places on elements of S and T; note that p_S and p_T are functions on domain S and T, but generally not probability distributions.

Algorithm 2: Algorithm for Testing Monotonicity over a Matching poset.

```
Sample-from-$p'$($s$, sample access to $p$)
Comment: $p'$ consists of half $p$ and half uniform.
$\mathcal{S} \leftarrow \emptyset$
for $i = 1, \ldots, s$ do
    if a (fresh) fair coin-toss comes up head then
        Draw a sample from $p$ and add to $\mathcal{S}$
    end
    else
        Draw a uniform random vertex $x_i$ and add to $\mathcal{S}$ (where $x \in \{u, v\}$ and $i \in [n]$)
    end
end
return $\mathcal{S}$ Monotonicity-Testing-over-$M_n$($\epsilon$, sample access to $p$) $\epsilon' \leftarrow \epsilon/14$
$\mathcal{S} \leftarrow$ Sample-from-$p'$($O(\frac{n}{\epsilon'^2 \log n})$, sample access to $p$)
$\widetilde{g} \leftarrow$ Apply Theorem 22 for $p'$ with error parameter $\epsilon'$ using samples $\mathcal{S}$
$\hat{w}_S \leftarrow$ Approximate total probability mass that $p'$ places on $S$ using $O(1/\epsilon')$ samples
$\hat{w}_T \leftarrow 1 - \hat{w}_S$
$\hat{g} \leftarrow$ Rescale $\widetilde{g}$ to satisfy $\hat{g}(\hat{w}_S \cdot x, \hat{w}_T \cdot y) = \widetilde{g}(x, y)$
$g^* \leftarrow$ Compute a function minimizing $W(\hat{g}, g^*)$ defined according to some monotone $q^*$
if $W(\hat{g}, g^*) \leq 3\epsilon'$ then
    return accept.
end
else
    return reject.
end
```

The outline of our algorithm is given as Procedure Monotonicity-Testing-over-M_n in Algorithm 2. In our algorithm, we hope to invoke Theorem 22 by considering the (normalized) p_S and p_T as our p_1 and p_2, respectively. However, Theorem 22 requires roughly the same number of samples from both p_1 and p_2, while p_S and p_T may have vastly different total probability masses; for instance, it may be costly to try to obtain many samples from S.

Before we proceed, by Theorem 15, it is straightforward to see:

$$\sum_{i \in [n]} \max\{p(u_i) - p(v_i), 0\} \geq d_{TV}(p, \mathsf{Mon}(M_n)) \geq \frac{1}{2} \sum_{i \in [n]} \max\{p(u_i) - p(v_i), 0\}.$$

In order to make the probability of the top and the bottom vertices at least a constant, we define an auxiliary probability distribution p' obtained by averaging p with a monotone distribution: $p'(w) = p(w)/2 + 1/(4n)$ where $w \in V$. Clearly, if p is monotone, then p' is monotone too. Also, if p is ϵ-far from monotone, then observe that the distance of p' to monotone is

$$\begin{aligned} d_{TV}(p', \mathsf{Mon}(M_n)) &\geq \frac{1}{2} \sum_{i \in [n]} \max\{p'(u_i) - p'(v_i), 0\} \\ &\geq \frac{1}{2} \sum_{i \in [n]} \max \left\{ \left(\frac{p(u_i)}{2} + \frac{1}{4n} \right) - \left(\frac{p(v_i)}{2} + \frac{1}{4n} \right), 0 \right\} \\ &\geq \frac{1}{2} \sum_{i \in [n]} \max \left\{ \frac{p(u_i) - p(v_i)}{2}, 0 \right\} \geq \frac{1}{4} d_{TV}(p, \mathsf{Mon}(M_n)) \geq \frac{\epsilon}{4} \,, \end{aligned}$$

which preserves the distance to monotone to a factor of 4. We can generate samples for p' using asymptotically the same number of samples from p: A sample from p' is obtained by drawing a sample from p or drawing a uniform random vertex with probability 1/2 each (Procedure Sample-from-p' in Algorithm 2); henceforth, we consider the problem of testing p' for monotonicity with distance $\epsilon/4$ instead.

The main benefit for considering the monotonicity testing on p' instead of p is that the total amount of probability masses placed on S and on T are at least $1/4 = \Omega(1)$ each. Hence, it takes $\Theta(s)$ samples from p according to the procedure above to obtain at least s samples from each of S and T with good constant probability; that is, we can create our input for the algorithm in Theorem 22 using $\Theta(s)$ i.i.d. samples from p.

Denote by w_S, w_T the total probability masses that p' places on S and T, respectively. Let p'_S and p'_T be the probability function that p assigns to vertices of S and T, respectively. Let $\widetilde{p'_S}$ and $\widetilde{p'_T}$ be the distributions over S and T that are obtained by normalizing p'_S and p'_T (separately). More precisely, we have

$$\widetilde{p'_S}(i) = \frac{p'_S(i)}{w_S} = \frac{p'(u_i)}{w_S}, \text{ and } \widetilde{p'_T}(i) = \frac{p'_T(i)}{w_T} = \frac{p'(v_i)}{w_T} \qquad \text{for } i \in [n] \,.$$

Let $\epsilon' = \Theta(\epsilon)$ (to be determined exactly later). Invoking Theorem 22 with this parameter, we obtain a function $\widetilde{g}$ where $W(h_{\widetilde{p'_T}, \widetilde{p'_S}}, \widetilde{g}) \leq \epsilon'$ using $O(\frac{n}{\epsilon^2 \log n})$ samples from p.

Next, we rescale each dimension of $\widetilde{g}$ back by w_S and w_T, thereby obtaining our estimate of $h_{p'_S, p'_T}$. If we knew w_S and w_T exactly, we would define $g(w_S \cdot x, w_T \cdot y) = \widetilde{g}(x, y)$, and we would have $W(h_{p'_S, p'_T}, g) \leq \epsilon'$. However, we can only estimate w_S and w_T up to an additive error ϵ' with high constant probability using $O(1/\epsilon^2)$ samples. To this end, let $\hat{w}_S$ be the estimate of w_S, and let $\hat{w}_T = 1 - \hat{w}_S$. We define $\hat{g}$ for which $\hat{g}(\hat{w}_S \cdot x, \hat{w}_T \cdot y) = \widetilde{g}(x, y)$. Below, we show that $\hat{g}$ is a good estimation of $h_{p'_S, p'_T}$.

Recall that $W(h_{\widetilde{p'_S}, \widetilde{p'_T}}, \widetilde{g}) \leq \epsilon'$. By definition, there exists a minimum-cost sequence of steps $\langle (c_r, (x_r, y_r), (x'_r, y'_r)) \rangle_{r \in [R]}$ for turning $\widetilde{g}$ to $h_{\widetilde{p'_T}, \widetilde{p'_S}}$:

$$W(h_{\widetilde{p'_T}, \widetilde{p'_S}}, \widetilde{g}) = \sum_{r \in [R]} c_r \left(|x_r - x'_r| + |y_r - y'_r| \right) \leq \epsilon.$$

Observe that under the cost function in Definition 5, we may assume without loss of generality that there are no r, r' such that $(x, y) = (x'_r, y'_r) = (x_{r'}, y_{r'})$ in the moving scheme. Namely, we can instead "shortcut" this scheme by moving the value $\min\{c_r, c_{r'}\}$ from (x_r, y_r)

to $(x'_{r'}, y'_{r'})$ directly without leaving any extra amount at (x, y) (during step r) to pick up later (during step r'). In this moving scheme, the value of $h_{p^r_S, p^r_T}$ on any (x, y) must be non-increasing or non-decreasing throughout the steps $r \in [R]$ (since values are only being moved *in*, or only being moved *out*, but not a mixture of both). In particular, this condition implies that the total value of c_r's moving into (x', y') never exceeds the value of $h_{\widetilde{p'_S}, \widetilde{p'_T}}(x', y')$. More formally,

$$\sum_{r \text{ s.t. } x'_r = x', y'_r = y'} c_r \le h_{\widetilde{p'_S}, \widetilde{p'_T}}(x', y').$$

Now, we are ready to bound $W(\hat{g}, h_{p'_S, p'_T})$. By definition, we have $h_{p'_S, p'_T}(w_S \cdot x, w_T \cdot y) = h_{\widetilde{p'_S}, \widetilde{p'_T}}$ and $\hat{g}(\hat{w}_S \cdot x, \hat{w}_T \cdot y) = \tilde{g}(x, y)$. Thus, any moving scheme that turns $\tilde{g}$ into $h_{\widetilde{p'_S}, \widetilde{p'_T}}$, will also turn $\hat{g}$ into $h_{p'_S, p'_T}$. Hence, we can use the same sequence (up to scaling) for moving the mass from $h_{\widetilde{p'_S}, \widetilde{p'_T}}$ to $\tilde{g}$ to show a bound for $W(\hat{g}, h_{p'_S, p'_T})$: at step $r \in [R]$, we move the value c_r from $g(\hat{w}_S \cdot x, \hat{w}_T \cdot y)$ to $h(w_S \cdot x', w_T \cdot y')$. We establish our bound as follows.

$$\begin{aligned}
W(\hat{g}, h_{p'_S, p'_T}) &\le \sum_{r \in [R]} c_r \left(|\hat{w}_S \cdot x_r - w_S \cdot x'_r| + |\hat{w}_T \cdot y_r - w_T \cdot y'_r| \right) \\
&= \sum_{r \in [R]} c_r \left(|\hat{w}_S \cdot x_r - \hat{w}_S \cdot x'_r + \hat{w}_S \cdot x'_r - w_S \cdot x'_r| + |\hat{w}_T \cdot y_r - \hat{w}_T \cdot y'_r + \hat{w}_T \cdot y'_r - w_T \cdot y'_r| \right) \\
&\le \sum_{r \in [R]} c_r \left(\hat{w}_S |x_r - x'_r| + w_T |y_r - y'_r| \right) + \sum_{r \in [R]} c_r \left(|w_S - \hat{w}_S| \cdot x'_r + |w_T - \hat{w}_T| \cdot y'_r \right) \\
&\le \left(\sum_{r \in [R]} c_r \left(|x_r - x'_r| + |y_r - y'_r| \right) \right) + \epsilon' \cdot \left(\sum_{r \in [R]} c_r \left(x'_r + y'_r \right) \right) \\
&\le W(\tilde{g}, h_{\widetilde{p'_S}, \widetilde{p'_T}}) + \epsilon' \cdot \left(\int_{x=0}^{\infty} \int_{y=0}^{\infty} h_{\widetilde{p'_S}, \widetilde{p'_T}}(x, y) \cdot (x + y) \, dy \, dx \right) \\
&\le \epsilon' + \epsilon' \cdot \left(\sum_i \widetilde{p'_S}(i) + \widetilde{p'_T}(i) \right) = 3\, \epsilon'.
\end{aligned}$$

Going back to our algorithm, we compute g^*: the function minimizing $W(\hat{g}, g^*)$ that is also defined according to an actual *monotone* probability distribution q^* over V. Observe that if p' is monotone, then

$$W(\hat{g}, g^*) \le W(\hat{g}, h_{p'_S, p'_T}) \le 3\epsilon'$$

due to the optimality assumption above. On the other hand, if p' is $\epsilon/4$-far from monotone, then by choosing $\epsilon' = \epsilon/14$,

$$\begin{aligned}
W(\hat{g}, g^*) &\ge W(h_{p'_S, p'_T}, g^*) - W(h_{p'_S, p'_T}, \hat{g}) \\
&\ge \|p', q^{*(\pi)}\|_1 - W(h_{p'_S, p'_T}, \hat{g}) \\
&= 2\, d_{TV}(p', q^{*(\pi)}) - W(h_{p'_S, p'_T}, \hat{g}) \ge 2(\epsilon/4) - 3\epsilon' = 4\epsilon',
\end{aligned}$$

for some permutation π over $[n]$, where $q^{*(\pi)}(u_i) = q^*(u_{\pi(i)})$ and $q^{*(\pi)}(v_i) = q^*(v_{\pi(i)})$, making use of Lemma 21 above. Hence, g provides us with a condition for testing monotonicity over the matching poset M_n, as desired. ■

E.3. An Algorithm for Testing Monotonicity on Bounded Degree Bipartite Graphs with Sub-linear Sample Complexity

We give an algorithm which tests monotonicity of a distribution p on a *bipartite* poset G with sample complexity $O\left(\frac{\Delta^3 n}{\epsilon^2 \log n}\right)$ where Δ denotes an upper bound for the maximum degree over all vertices in G. Given sample access to the distribution p, we implement a sampling oracle for a certain distribution p' on a *matching* poset G' with $O(\Delta n)$ vertices. This distribution p' is monotone on G' if p is monotone on G, and p' is $\epsilon/(2\Delta)$-far from monotone on G' if p is ϵ-far on G. Hence, we apply the algorithm for testing monotonicity on the matching poset G' to test the monotonicity of p', immediately obtaining the desired sample complexity. We describe the construction of G' and the distribution p' below and show the correctness of our approach in Theorem 25.

More formally, let p be a distribution over a directed bipartite poset $G = (V = V_b \cup V_t, E)$ where $V_b = \{u_i\}_{i \in [n]}$ and $V_t = \{v_i\}_{i \in [n]}$ are the sets of the bottom and the top vertices, and $E \subseteq V_b \times V_t$ is the set of edges. Let Δ be an upper bound on the degree of G.

The matching poset G'. Based on G, we create a matching $G' = (V' = V'_b \cup V'_t, E')$ over $n' = O(\Delta n)$ vertices according the following procedure. Similar to G, V'_b is the set of bottom vertices, V'_t is the set of top vertices, and E' is the set of edges.

- Create Δ *copy vertices* $w^1, \ldots, w^\Delta$ for each vertex $w \in V$.
- For each edge $e = (u, v) \in E$, match an unmatched pair of vertices u^i, v^j via the *copy edge* $e' = (u^i, v^j)$; place $u^i \in V'_b$, $v^j \in V'_t$ and $e' \in E'$.
- For all remaining unmatched vertices w^i, create a *dummy vertex* $\bar{w}^i$, then match it to w^i via the *dummy edge* $\bar{e}_{w^i} = (\bar{w}^i, w^i)$; place $\bar{w}^i \in V'_b$, $w^i \in V'_t$ and $\bar{e}_{w^i} \in E'$. Note that the dummy vertex is always put in the bottom set.

Note that the second step above is always possible since there are at most Δ edges incident to a vertex.

Distribution p' over G'. The distribution p' over the poset G' is defined as follows. For each copy vertex w^i, set $p'(w^i) = p(w)/\Delta$. For each dummy vertex $\bar{w}^i$, set $p'(\bar{w}^i) = 0$. One can generate a sample from p', by drawing a sample w in V according to p, and drawing i uniformly at random from $[\Delta]$: The i-th copy of w, w^i, is a sample drawn from p'.

In the following lemma, we show that the distance of p' to being monotone is closely related to the distance of p to monotonicity.

Lemma 24 *Let p and p' be two distributions over G and G' as described above. If p is monotone, then p' is monotone. If p is ϵ-far from being monotone, then p' is $(\epsilon/2\Delta)$-far from being monotone.*

Proof Observe that for each copy edge $e' = (u^i, v^j)$, the probabilities at the endpoints are $p'(u^i) = p(u)/\Delta$ and $p'(v^j) = p(v)/\Delta$, respectively. Thus, if $p(u)$ is at most $p(v)$, then $p'(u^i)$ will remain at most $p'(v^j)$. Furthermore, for each dummy edge $\bar{e}_{w^i} = (\bar{w}^i, w^i)$, the probability of the bottom vertex, $p'(\bar{w}^i)$, is zero, so this edge never violates the monotonicity of G'. Hence it follows immediately that if p is monotone on G, then p' is monotone on G' as well.

On the other hand, assume p is ϵ-far from being monotone. We define a weighted graph on the transitive closure of G, $TC(G)$, where the weight of each edge (u, v) is $\max(p(u) - p(v), 0)$. By the proof of Theorem 15, $TC(G)$ has a weighted matching, namely M, of weight W such that

$$\frac{W}{2} \leq d_{TV}(\mathsf{Mon}(G), p) \leq W\,. \tag{6}$$

Since G is a bipartite poset, and the edges are all from V_b to V_t, $TC(G)$ is the same as G. Hence, each edge $e = (u, v)$ in M is in E as well. Also, by the construction of G', there exists a copy edge $e' = (u^i, v^j) \in E'$ that corresponds to e. Let M' be the set of copy edge $e' = (u^i, v^j)$ where $e = (u, v)$ is in M. M' is a matching in G' as well.

Observe that by the above construction, the weight of $e' = (u^i, v^j)$ is $\max(p'(u^i) - p'(v^j), 0) = \max(p(u) - p(v), 0)/\Delta$. Hence, G' contains a matching, M', of weight $W' := W/\Delta$ which is at most the weight of the maximum matching in G'. Let W' be the weight of the maximum matching in G'. By Theorem 15 and Equation 6, we obtain:

$$\frac{d_{TV}(\mathsf{Mon}(G), p)}{2\Delta} \leq \frac{W}{2\Delta} \leq \frac{W'}{2} \leq d_{TV}(\mathsf{Mon}(G'), p')\,.$$

Thus, if p is ϵ-far from being monotone, then p' is $\epsilon/(2\Delta)$-far from monotone as well, concluding the lemma. ■

Given the above lemma, it is sufficient to test monotonicity of p' with proximity parameter $\epsilon' = \epsilon/(2\Delta)$. See Algorithm 3 for the steps. Below, we show the correctness of the algorithm.

Algorithm 3: Reduction from testing monotonicity bipartite to Matching .

```
Reduction($G, n, \Delta, \epsilon$, sample access to $p$) $\epsilon' \leftarrow \epsilon/2\Delta$
$G' \leftarrow$Construct the matching poset from $G$ as described.
$\mathcal{S} \leftarrow$ Generate $O(\frac{\Delta^3 n}{\epsilon'^2 \log n})$ samples from $p'$
Test if $p'$ is monotone or $\epsilon'$-far from it via Algorithm 2 using the samples in $\mathcal{S}$.
Output the result of the test.
```

Corollary 25 *There exists an algorithm that tests whether a distribution p over a bipartite poset G of n vertices and maximum degree Δ, is monotone or ϵ-far from monotone with success probability $2/3$, using $O(\frac{\Delta^3 n}{\epsilon^2 \log n})$ i.i.d. samples from p.*

Proof Given Lemma 24, it suffices to test the monotonicity of G' with parameter $\epsilon' = \epsilon/\Delta$. Using Theorem 23 and since G' is a matching of size $n' = O(\Delta n)$, one can test monotonicity of p' with high probability using $O(n'/(\epsilon'^2 \log n')) = O(\Delta^3 n/(\epsilon^2 \log n))$ samples as desired. Therefore, the proof is complete. ■

E.4. Testing monotonicity of distributions that are uniform on a subset of the domain

In this section, we give an algorithm for testing monotonicity on a specific yet broad class of instances. More specifically, suppose that we are given a directed bipartite graph $G(V = V_T \cup V_B, E \subseteq V_T \times V_B)$, along with a probability distribution on the set V. Note that all the directed edges go from a vertex in the "bottom" set V_B, to a vertex in the "top" set V_T. We additionally assume that all distributions which we sample from are *uniform on a subset* of V whose size is known to the algorithm. That is, for every vertex $u \in V$ either $p_u = 0$ or $p_u = 1/|R|$, where R is the support of the distribution p.

We will show the following result:

Theorem 26 *Let G be a directed bipartite graph as described above and p be a probability distribution on V which is uniform on a subset of V, namely R. Given the size of R, there exists an algorithm with sample complexity $O(\frac{n^{2/3}}{\epsilon} + \frac{1}{\epsilon^2})$ that can test, with success probability $2/3$, whether p is monotone on G, or p is ϵ-far from any monotone function on G,*

At a high level, our tester works as follows: We draw an initial set $\mathcal{S}_1$ of s_1 samples from p. We define $B = \mathcal{S}_1 \cap V_B$ to be the set of vertices from the bottom, V_B, that we see in the sample set. Then, we look at the set $T \subseteq V_T$ containing all out-neighbors of the vertices in B. We show the following structural property of distributions that are ϵ-far from being monotone: in expectation, the constructed set T contains ϵ/s_1 endpoints of violating edges, so $|T|$ cannot be too small. Thus, if $|T|$ is much smaller than ϵ/s_1, we can immediately conclude that the distribution is close in total variation distance to some monotone distribution. However, if T is sufficiently large in cardinality, we draw more samples in order to estimate the amount of probability mass on T. Note that if p is monotone, then we expect that all the elements in T be in the support of the distribution, namely R, so every single element of T should have probability mass $\frac{1}{|R|}$ for the distribution to be monotone. The tester rejects if there is sufficient evidence that this is not the case. More specifically, the proposed tester is given in Algorithm E.4.

Proof of Theorem 26: As given in the algorithm, let $s_1 = O(\frac{n^{2/3}}{\epsilon})$ and $s_2 = O(n^{2/3})$ denote the sample sizes of the two steps described earlier. We consider the following two cases.

Completeness case: Assume p is a monotone distribution. Clearly, each sample we draw has a non-zero probability. Since we pick T to be the neighbor set of the samples we draw, we know that every element in T has a non-zero probability. By the uniformity assumption, this probability is $|T|/|R|$. Thus, when we draw s_2 samples from the distribution we expect $|T|/|R|$ fraction of them fall into T. So, the expected value of $|Y|$ is $s_2 \cdot |T|/|R|$. We defer the asymptotic complexity analysis of this case to the end of our proof.

Soundness case: Assume p is ϵ-far from being a monotone distribution. Consider all the violating edges (u, v) in E for which $p(u)$ is greater than $p(v)$. By Lemma 16, there exists a set of edges, namely M, that form a matching, and we have:

$$\sum_{(u,v)\in M} p(u) - p(v) \geq \epsilon\,.$$

Algorithm 4: Algorithm for Testing Monotonicity of a Uniform Distribution over a subset of the domain.

```
Monotonicity-Test($G, \epsilon, |R|$, and sample access to $p$)
$\mathcal{S}_1 \leftarrow$ Draw $s_1 = O(\frac{n^{2/3}}{\epsilon})$ samples from $p$.
$B \leftarrow \mathcal{S}_1 \cap V_B$      $\sharp$ where $V_B$ is the set of bottom vertices
$T \leftarrow N(B)$      $\sharp N(B)$ is the neighbor set of the set $B$
if $|T| \leq \frac{\epsilon s_1}{2}$ then
    return accept
end
$\mathcal{S}_2 \leftarrow$ Draw $s_2 = O(n^{2/3})$ samples from $p$.
$Y \leftarrow T \cap \mathcal{S}_2$
$\epsilon' \leftarrow \frac{\epsilon \cdot s_1}{2|T|}$
if $|Y| \geq s_2 \cdot (1 - \frac{\epsilon'}{2}) \cdot \frac{|T|}{|R|}$ then
    return accept
end
else
    return reject
end
```

Note that without loss of generality one can assume M only has violating edges, since removing non-violating edges only makes the left hand side larger. By the uniformity assumption for p, $p(u) - p(v)$ is exactly $1/|R|$. Thus, by the above inequality, we have $|M|/|R|$ is at least ϵ.

Since there are $|M|$ vertices in V_B that belong to the matching, $|B \cap M|$ is a random variable distributed according to the binomial distribution $\mathbf{Bin}(s_1, |M|/|R|)$, we have that

$$\mathbf{E}\left[|B \cap M|\right] = \frac{s_1 \cdot |M|}{|R|} \geq \epsilon s_1 \,.$$

Using Chebyshev's inequality and the fact that $|B \cap M|$ is a binomial distribution, we have

$$\begin{aligned}\Pr\left[|B \cap M| \leq \frac{\epsilon s_1}{2}\right] &\leq \Pr\left[|B \cap M| \leq \frac{\mathbf{E}[|B \cap M|]}{2}\right] \leq \frac{4\mathbf{Var}[|B \cap M|]}{\mathbf{E}[|B \cap M|]^2} \\ &\leq \frac{4 s_1 \cdot (|M|/|R|) \cdot (1 - |M|/|R|)}{(s_1 \cdot |M|/|R|)^2} \leq \frac{4}{\epsilon s_1} = O(n^{-2/3}) \,.\end{aligned}$$

Thus, with high probability, B contains at least $\epsilon s_1/2$ endpoints in M. Note that the neighbor set of B contains the other endpoints of the edges in the matching M. Thus, T contains at least $|B \cap M|$ vertices of zero probability, which implies that the size of T has to be at least $\epsilon s_1/2$. Hence, for sufficiency large n, the probability that p gets rejected due to the condition $|T| \leq \epsilon s_1/2$ is negligible.

Consider the second set of samples we draw in the algorithm S_2. Clearly, the size of $Y \coloneqq T \cap S_2$ is a binomial random variable drawn from $\mathbf{Bin}(s_2, |T \cap R|/|R|)$. However, we show that $\epsilon \coloneqq \epsilon s_1/(2|T|)$ fraction of the elements in T have zero probability. Thus, $|T \cap R|/|R|$ is at most $(1 - \epsilon')|T|/|R|$ while in the completeness case it is $|T|/|R|$. So, we only need to estimate the bias of a Bernoulli random variable up to an additive error of $\epsilon'' := \epsilon'|T|/(2|R|)$. By Hoeffding bound, we only need to draw $O(1/\epsilon''^2)$ samples to distinguish the two cases with high probability which implies:

$$s_2 = \Theta\left(\frac{1}{\epsilon''^2}\right) = \Theta\left(\frac{|R|^2}{\epsilon'^2|T|^2}\right) \leq O\left(\frac{n^2}{\epsilon^2 s_1^2}\right) = O(n^{2/3})$$

Thus, with high probability, we distinguish them correctly.

E.5. Upper bound via trying all matchings

In this section we present a simple upper bound for the problem of monotonicity testing on bipartite graphs. Let $\mathcal{M}$ be the number of pairs of subsets (S_t, S_b) of top and bottom elements respectively for which there exists a perfect matching between them. The algorithm is the following:

Algorithm 5: Algorithm for Testing Monotonicity on a bipartite graph.

```
MONOTONICITY-TEST($G, \epsilon$, and sample access to $p$)
$s \leftarrow$ draw $O(\log \mathcal{M}/\epsilon^2)$ samples from $p$.
for each pair of equal size subsets $(S_t, S_b)$ of top and bottom elements respectively do
    if there exists a perfect matching between $S_t$ and $S_b$ then
        $\hat{w}_t \leftarrow$ Estimate the total probability mass of $S_t$
        $\hat{w}_b \leftarrow$ Estimate the total probability mass of $S_b$
        if $\hat{w}_t$ is less than $\hat{w}_b - \epsilon/2$ then
            return reject
        end
    end
end
return accept
```

Theorem 27 *We can test whether a distribution p over a bipartite graph G with n vertices is monotone or ϵ-far from any monotone distribution with success probability $2/3$, using $O((\log M)/\epsilon^2)$ samples, where M is the number of pairs of subsets of top and bottom elements respectively for which there exists a perfect matching between them. That is, $O(n/\epsilon^2)$ samples for a worst case graph G.*

Proof Let w_t and w_b denote the probability mass of S_t and S_b respectively. Note that if we use $O(1/\epsilon^2)$ samples, we can estimate w_t and w_b within an additive error of $\epsilon/8$. Thus, we can estimate the difference of the two with error of $\epsilon/4$ with a constant probability. We can amplify the probability of the correctness, by repeating the estimation and taking the median of them. Therefore, for each pair of subsets, the probability that the algorithm fails to estimate the difference of w_b and w_t within an error of $\epsilon/4$ is at most $O(\frac{1}{M})$. By union bound, we distinguish whether $w_b - w_t$ is at least ϵ or at most zero by comparing the $\hat{w}_b - \hat{w}_t$ with $\epsilon/2$, with a constant success probability.

Now, if p is ϵ-far from being monotone with respect to the graph G, there exists a matching such that the total difference between the probabilities of the bottom and the top elements, $w_b - w_t$ is at least ϵ by Lemma 16. Thus, in one of the iteration, we will consider this matching, and output reject. Also, if p is monotone with respect to the graph G, there is no violating edge. Therefore, for each pair S_t and S_b, we have $w_b - w_t \leq 0$. Thus, in no iteration we output reject, and the distribution will be accepted at the end.

Lastly, since there are at most $2^{n_t} \cdot 2^{n_b} = 2^{n_t + n_b} = 2^n$ pairs of subsets where n_t, n_b is the total number of top and bottom elements respectively, we conclude that the sample complexity is $O(n/\epsilon^2)$. ■

Remark: Note that in order to execute the above algorithm, it is not required to know the quantity M in advance. We can instead draw more samples and update all our estimates at the same time to sufficiently reduce the error probability for each estimate for the union bound to work.