

A Distribution Testing Approach to Clustering Distributions

Gunjan Kumar

IIT Kanpur

GUNJAN@CSE.IITK.AC.IN

Yash Pote

National University of Singapore

YASHPPOTE@GMAIL.COM

Jonathan Scarlett

National University of Singapore

SCARLETT@COMP.NUS.EDU.SG

Editors: Steve Hanneke and Tor Lattimore

Abstract

We study the following distribution clustering problem: Given a hidden partition of k distributions into 2 groups, such that the distributions within each group are the same, and the distributions associated with the clusters are pairwise ε -far in total variation, the goal is to recover the partition. We establish upper and lower bounds on the sample complexity for two fundamental cases: (1) when one of the cluster’s distributions is known, and (2) when both are unknown. Our upper and lower bounds characterize the sample complexity’s dependence on the domain size n , number of distributions k , size r of one of the clusters, and distance ε . In particular, we achieve tightness with respect to (n, k, r, ε) (up to an $O(\log k)$ factor) for all regimes. In addition, we show that this result extends to the case of d -clustering for any constant number of clusters d .

Keywords: Distribution Testing, Clustering

1. Introduction

Imagine that there has been a manufacturing defect at the dice factory, and an unknown number of dice are created with a common defect. Unfortunately, all the dice look identical, so the only way to find the fair dice is to roll them and observe the outcomes. How many rolls are needed to determine the set of fair dice?

This is a distribution clustering problem, and is formalised as follows: We are given sample access to distributions D_1, D_2, \dots, D_k , and a parameter ε . It is known that there is a hidden partition $\{\mathcal{I}, [k] \setminus \mathcal{I}\}$ of $[k]$ into two clusters such that all distributions within each cluster are identical, and distributions in different clusters are ε -far in Total Variation (d_{TV}), and the goal is to find the hidden partition. We are interested in the sample complexity, i.e., the number of samples required to correctly determine the partition with probability at least $\frac{2}{3}$. This probability can be “boosted” to $1 - \delta$ by repeating the algorithm $O(\log \frac{1}{\delta})$ times.

Broadly speaking, clustering is a fundamental problem in unsupervised learning, where the objective is to group together objects that are similar to each other while separating those that are dissimilar. Clustering of *distributions* has been studied in various forms in previous work, e.g., (Zhuang et al., 2022; Yang et al., 2024; Yavas et al., 2025), as we further discuss in Section 1.1. Recent advances on this problem, particularly in the literature on bandit algorithms, have focused on designing clustering algorithms with strong asymptotic guarantees. These methods operate in the fixed-confidence setting, where the goal is to output the partition with error at most δ . Certain optimality results are established in the limit $\delta \rightarrow 0$. However, these asymptotic analyses do not

provide results for the practical scenario where δ is constant. In particular, the precise dependence on important parameters, such as the domain size n , is hidden in lower-order terms in these analyses, and remains to be fully characterized.

In contrast, our approach builds on advances in distribution testing (surveyed by [Canonne \(2020, 2022\)](#)), where the focus is on finite sample complexity for testing properties of distributions. The goal in distribution testing is to determine whether a given distribution possesses a certain property of interest (such as uniformity) or is far from having it. Over the past decade, this area has seen significant progress, with novel algorithms developed to test various distributional properties, including uniformity, identity, and equivalence. Our work takes a distribution testing approach to clustering distributions, where we assume that the distributions in different clusters are ‘far’ (in total variation distance) and seek to determine the optimal number of required samples to attain a given constant error probability.

A naive approach for our setting is to fix one distribution, say D_1 , and then test whether each of the remaining distributions D_2, \dots, D_k is identical to D_1 . Depending on whether D_1 is known or unknown, this can be done using identity or equivalence testing algorithms. This approach requires $kf(n)$ samples, where $f(n)$ denotes the sample complexity of testing one distribution (see [Table 2](#)).

Is this dependence unavoidable? Or can the whole task be accomplished with fewer than $kf(n)$ samples? We show that we can do better than the naive approach by developing a two-stage algorithm that first identifies an exemplar from each cluster and then classifies the rest.

As an initial example, consider the die testing example above, where one kind of distribution is uniform (U) and the other is unknown (P), and ε -far from uniform. The naive method requires $k\sqrt{n}$ samples, while the optimal sample complexity turns out to be $\sqrt{nk} \max\left(\frac{\sqrt{k}}{r}, 1\right)$ where r is the number of distributions of the P type. Thus, it is \sqrt{k} times harder to find a loaded die in a bag of fair dice than it is to find a fair die in a bag of loaded dice. Intuitively, the latter scenario is easier because we can more readily learn information about the unknown distribution P .

1.1. Related Work

A Solved Case: Known Distributions The problem of clustering k distributions into two groups based on known reference distributions P and Q has a simple algorithm known as Scheffé’s test ([Scheffé, 1947](#); [Devroye and Lugosi, 2001](#)). For each distribution D_i , the task of deciding whether $D_i = P$ or $D_i = Q$ is achieved by sampling from D_i and estimating the probability of them falling into the Scheffé set $S^+ = \{x : P(x) > Q(x)\}$. This reduces the problem to a hypothesis test on a Bernoulli mean: testing if the mean is $p_0 = Q(S^+)$ or $p_1 = P(S^+)$. Observing that $p_1 - p_0 = d_{\text{TV}}(P, Q) = \varepsilon$, and that the complexity to classify a single distribution D_i with error probability $\frac{1}{3k}$ is $O\left(\frac{\log(k)}{\varepsilon^2}\right)$, we can run k such tests to classify all k distributions with a total error probability of at most $1/3$. The sample complexity is thus $O\left(\frac{k \log(k)}{\varepsilon^2}\right)$. A near-matching lower bound of $\Omega\left(\frac{k}{\varepsilon^2}\right)$ is shown by [Mannor and Tsitsiklis \(2004\)](#).

Distribution Testing Introduced by [Goldreich et al. \(1998\)](#) and [Batu et al. \(2000\)](#), distribution testing is a topic in property testing concerned with deciding whether a distribution has a certain property. For instance, in *uniformity testing*, one must decide if an unknown distribution Q is uniform or ε -far from uniform in total variation distance ([Paninski, 2008](#)). More generally, *identity testing* involves distinguishing between $d_{\text{TV}}(P, Q) = 0$ and $d_{\text{TV}}(P, Q) \geq \varepsilon$ for a known distribution P and an unknown distribution Q . If both P and Q are unknown, the problem is called

Bound Type	Sample Complexity	Conditions
Case 1: One distribution unknown		
Upper	$O\left(\frac{k \log(k) \sqrt{n}}{r \varepsilon^2} + \frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right)$	$n \gtrsim k \log(k)$
	$O\left(\frac{k \log(k) \sqrt{n}}{r \varepsilon^2} + \frac{k \log(k)}{\varepsilon^2}\right)$	$n \lesssim k \log(k)$
Lower	$\Omega\left(\frac{k \sqrt{n}}{r \varepsilon^2}\right)$	$r \leq \frac{k}{60}$
	$\Omega\left(\frac{k + \sqrt{nk}}{\varepsilon^2}\right)$	–
Case 2: Both distributions unknown		
Upper	$O\left(\max\left(\left(\frac{nk}{r \varepsilon^2}\right)^{2/3}, \frac{k \sqrt{n}}{r \varepsilon^2}\right) \log(k) + \left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}\right)$	$n \gtrsim \frac{k \log(k)}{\varepsilon^4}$
	$O\left(\max\left(\left(\frac{nk}{r \varepsilon^2}\right)^{2/3}, \frac{k \sqrt{n}}{r \varepsilon^2}\right) \log(k) + \frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right)$	$k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4}$
	$O\left(\max\left(\left(\frac{nk}{r \varepsilon^2}\right)^{2/3}, \frac{k \sqrt{n}}{r \varepsilon^2}\right) \log(k) + \frac{k \log(k)}{\varepsilon^2}\right)$	$n \lesssim k \log(k)$
Lower	$\Omega\left(\max\left(\left(\frac{nk}{r \varepsilon^2}\right)^{2/3}, \frac{k \sqrt{n}}{r \varepsilon^2}\right)\right)$	$r \leq \frac{k}{60}$
	$\Omega\left(\left(\frac{n^2 k}{\varepsilon^4}\right)^{1/3} + \frac{k + \sqrt{nk}}{\varepsilon^2}\right)$	–

Table 1: A summary of our results. The notation \lesssim and \gtrsim represents inequality to within suitable constant factors. The upper and lower bounds are shown to coincide to within at most a $\log k$ factor in Appendix F.

equivalence testing (Chan et al., 2014). We use results from these areas as building blocks for our algorithms, and we list them in Table 2. Goldreich (2020) showed that every identity testing in-

Problem	Complexity
Uniformity/Identity	$\Theta(\sqrt{n}/\varepsilon^2)$
Equivalence	$\Theta(\max(n^{2/3}/\varepsilon^{4/3}, \sqrt{n}/\varepsilon^2))$
Learning in TV	$\Theta(n/\varepsilon^2)$

Table 2: A summary of standard distribution testing results, e.g., see Canonne (2022).

stance can be converted to a uniformity testing problem over a slightly ($6\times$) larger domain. This connection will be used frequently in our analysis.

Likelihood-Free Hypothesis Testing (LFHT) is a related problem that addresses hypothesis testing problems in which the likelihood function is unknown or intractable; this problem is rooted in early works (Ziv, 2002; Gutman, 2002) and was recently studied in (Gerber et al., 2023; Gerber and Polyanskiy, 2024). The task in LFHT is to determine the optimal trade-off between the number of

samples s_1 drawn from two candidate distributions, P and Q , and the number of samples s_2 drawn from a third distribution D , to decide whether $D = P$ or $D = Q$.

We note that *clusterability* has been studied under a distribution testing framework by [Levi et al. \(2013\)](#). A list of distributions is said to be (c, β) -clusterable if there exists a c -clustering where each distribution in a cluster is β -close in TV distance from every other distribution in the same cluster. A list of distributions is ε -far from being clusterable if the *average* of distances of all distributions from the list from the closest clusterable list is at least ε . Their setting is significantly different from ours, as clusterability is 1) a decision (yes/no) problem, and 2) does not require samples from all distributions, as it is a bulk property. In contrast, clustering requires us to output a full partition, where each distribution is correctly placed.

Bandits and Clustering Multi-Armed Bandit techniques have been widely adopted to glean insights from stochastic data in fields ranging from quality control to user behavior analysis. While the classical goal in this field is maximizing the mean reward, several works have also studied the clustering objective ([Zhuang et al., 2022](#); [Yang et al., 2024](#); [Yavas et al., 2025](#)). Among these, perhaps the most relevant is [Yavas et al. \(2025\)](#), as they focus on clustering discrete distributions. However, there are several major differences that make their results incomparable to ours. Perhaps most importantly, they focus on obtaining tight sample complexity bounds as the error probability $\delta \rightarrow 0$ (via the so-called track-and-stop method), but the lower-order asymptotic terms hide the dependence on key parameters such as the domain size and cluster sizes. We elaborate on this in [Appendix A](#), where we highlight the fundamental differences between fixed- δ and vanishing- δ , and discuss how the techniques of [Yavas et al. \(2025\)](#) are unsuitable for the former.

1.2. Summary of Our Contributions

Our main contributions are outlined as follows:

- We propose a novel approach to the 2-clustering problem based on distribution testing, providing algorithms that leverage powerful existing learning/testing techniques.
- We analyze the sample complexity of our algorithms and provide matching lower bounds (up to logarithmic factors) in all parameter regimes (in terms of n, r, k , and ε).
- We show that our techniques naturally generalize to the problem of clustering into d groups for any constant d , preserving the same dependencies in the sample complexity.

A summary of our contributions is presented in [Table 1](#), with discussion in the next subsection.

Organisation The remainder of this paper is organized as follows. In [Section 2](#), we formalize the problem setting. In [Section 3](#), we present our clustering algorithms for the two variants and determine their sample complexity, and provide upper bounds on their sample complexity. We then establish our lower bounds in [Section 4](#), and outline the extension to more than two clusters in [Section 5](#). Finally, we conclude with directions for future work in [Section 6](#). Deferred proofs and discussions are provided in the appendices.

1.3. Discussion of Our Results

Our results, summarized in [Table 1](#), establish nearly tight upper and lower bounds on the sample complexity. The gap is at most a $\log(k)$ factor in the upper bounds, which comes from applying

a union bound over the k distributions. See Appendix F for a detailed description of the tightness of our results. The complexity of our algorithms is the sum of two costs, $T_1 + T_2$, corresponding to two stages: finding an exemplar from each cluster (T_1) and classifying the rest (T_2). We provide two distinct lower bounds that separately match the complexity of each stage, confirming that this two-stage approach is optimal.

The cost T_1 depends on the minority cluster size r , with smaller r being hardest (“needle-in-a-haystack”), and this term matches our r -dependent lower bounds (Theorem 4). In contrast, the classification cost T_2 is independent of r and can dominate when finding an exemplar is easy (i.e., for large r). This is matched by our r -independent lower bounds derived from an LFHT framework (Theorem 6). The different regimes for the domain size n arise from using the optimal underlying testing subroutine for the given parameters.

While our upper and lower bounds are written in a different form in Table 1, we will show in Appendix F that they coincide to within at most a $\log k$ factor in all regimes.

2. Problem Setup

We consider discrete distributions over the domain $[n] = \{1, 2, \dots, n\}$. We typically adopt the symbol D for a generic distribution over $[n]$, and let $x \sim D$ denote a single sample x drawn from D . We define the shorthand $D(S) := \sum_{i \in S} D(i)$ for any set S of elements from the domain of D . The total variation distance between two distributions is given as $d_{\text{TV}}(D_1, D_2) = \frac{1}{2} \sum_{i \in [n]} |D_1(i) - D_2(i)|$. We write $A \gtrsim B$ to mean $A \geq cB$, and write $A \asymp B$ to mean both $A \lesssim B$ and $B \lesssim A$.

Our problem setup is summarized as follows.

The Distribution 2-Clustering Problem

Input: Sample access to k distributions $\{D_i\}_{i=1}^k$ over domain $[n]$, parameter $\varepsilon \in (0, 1]$. The distributions are allowed to be queried adaptively (given knowledge of the previous samples).

Promise: There exists a hidden partition $\{\mathcal{I}_j\}_{j=1}^2$ of $[k]$ satisfying two conditions:

- $\forall (a, b \in \mathcal{I}_j), D_a = D_b$.
- $\forall (a \in \mathcal{I}_1, b \in \mathcal{I}_2), d_{\text{TV}}(D_a, D_b) \geq \varepsilon$.

Goal: With probability at least $2/3$, output the true partition $\{\mathcal{I}_j\}_{j=1}^2$.

Variants: We study two settings based on prior knowledge of the two clusters’ distributions:

- I.** One is known and the other is unknown, **II.** Both are unknown.

Although we state the problem for a success probability of $\frac{2}{3}$, it is known that repetition and a majority vote can boost the success probability from $\frac{2}{3}$ to $1 - \delta$ for any $\delta > 0$ at a multiplicative $O(\log(\frac{1}{\delta}))$ cost to the sample complexity. Since the output is a partition rather than a binary decision, we describe the boosting explicitly: run the algorithm $m = O(\log(\frac{1}{\delta}))$ times independently, among the m returned partitions we are guaranteed to have $\lceil \frac{m}{2} \rceil + 1$ correct clusterings with probability at least $1 - \delta$. Since the correct clusterings are all identical, we know the true partition is the unique clustering that appears at least $\lceil \frac{m}{2} \rceil + 1$ times.

Given k distributions, the sample complexity of 2-clustering depends on the sizes of the two clusters, which may be known or unknown. We will derive the stronger form of each result accordingly, i.e., our lower bounds are for the case where the cluster size is known, while our upper bounds do not require such knowledge (though we consider the known case as a stepping stone).

For the first variant **(I)**, the relevant factor is the size of the cluster for which the distribution is unknown, and for variant **(II)**, it is the size of the smaller cluster. We denote this factor as r , and our bounds will have tight dependences on r . Throughout the paper, we assume that the two clusters are non-empty i.e., $r \in [1, k - 1]$

2.1. Adaptivity, Cluster Size Knowledge, and Unknown ε

Our algorithms are adaptive, i.e., we allow the samples to be drawn sequentially, and the decision of which distribution to sample from at each step can be informed by all past outcomes. However, our algorithms will only require taking samples in two sequential batches if r is known, or $O(\log \frac{k}{r})$ sequential batches if r is unknown. On the other hand, our lower bounds are constructed to hold even for fully adaptive procedures, thus making the near-optimality results stronger.

For clarity of exposition, our algorithms are presented assuming the cluster size r is known. We describe the variation for unknown r in Appendix H, and show that it only requires minor modifications and at most a constant factor increase in samples, albeit with more adaptive rounds as mentioned above. Similarly, while our main presentation assumes the separation parameter ε is known, we show in Appendix I that this assumption can be relaxed. Specifically, we provide an adaptive algorithm that does not require knowledge of ε and achieves the same sample complexity up to $O(\log \log(1/\varepsilon))$ factors and an additive $O(\frac{n}{\varepsilon^2})$ term (which does not depend on k and is thus non-dominant in several large- k regimes including $k = \Omega(n)$).

3. Algorithms for 2-Clustering

3.1. One Unknown, One Known

Here we assume that we are given knowledge of one distribution Q , whereas the other distribution P is unknown. Using the Identity-to-Uniformity reduction mentioned in Section 1.1, we can assume without loss of generality that the known distribution Q is the uniform distribution U , and accordingly, we will henceforth use U instead of Q . A formal statement of this reduction is given in Appendix B. Thus, the task in this section is to find the hidden partition $\{\mathcal{I}_P, \mathcal{I}_U\}$.

Naive Upper Bound A naive approach is to test each distribution D_1, \dots, D_k for uniformity, ensuring a success probability of at least $1 - \frac{1}{k}$ for each test. This requires $O\left(\frac{\sqrt{n} \log(k)}{\varepsilon^2}\right)$ samples per distribution (see Table 2, with an extra $\log k$ term coming from the union bound), for a total sample complexity of $O\left(\frac{k \log(k) \sqrt{n}}{\varepsilon^2}\right)$.

A Better Upper Bound Our algorithm improves on the naive approach by proceeding in two stages. First, we find *one* distribution in \mathcal{I}_P , which gives us sample access to P . Given this sample access, we can use distribution testing techniques to efficiently classify *all* of the other distributions.

Stage 1: Find One Unknown We select a subset of $\lceil \frac{3k}{r} \rceil$ distributions uniformly at random, and test each of them for uniformity. Let E_1 be the event that this subset contains no non-uniform distributions. If we upper bound the failure probability by sampling with replacement,

then $\Pr[E_1] \leq \exp\left(-\frac{r}{k} \cdot \frac{3k}{r}\right) = e^{-3} \leq \frac{1}{9}$. Note that if $\frac{3k}{r} \geq k$, then the subset contains all k distributions, and hence $\Pr[E_1] = 0$ trivially. To ensure the uniformity tests are reliable, we use enough samples so that the probability of any test failing, E_2 , is bounded by $\Pr[E_2] \leq \frac{1}{9}$ via a union bound. Assuming the complement event $\overline{E_1} \wedge \overline{E_2}$ occurs, we successfully identify one distribution of the unknown type. By Table 2 along with an $O\left(\log \frac{k}{r}\right)$ factor for success amplification and the union bound, the sample complexity of this stage is $O\left(\frac{k}{r} \cdot \frac{\sqrt{n \log(k)}}{\varepsilon^2}\right)$.

Stage 2: Find All Unknowns Supposing that we completed Stage 1 successfully, we have identified one of the k distributions as being the unknown distribution P . Without loss of generality we will assume D_1 is such a distribution. We then attempt to locate all the unknown distributions. To do so, we will use one of two algorithms, MultiLFHT (Theorem 1) and ESW-Tester (Algorithm 1 in Appendix C.2), depending on the problem parameters.

We start by presenting the theorem stating the complexity of the MultiLFHT algorithm.

Theorem 1 *Let P and Q be two distributions with $d_{TV}(P, Q) \geq \varepsilon$. Given sample access to P and Q , there exists a procedure (which we refer to as MultiLFHT) that, with probability at least $\frac{8}{9}$, correctly classifies distributions $\{D_i\}_{i=1}^k$ as P or Q . The sample complexity is as follows in the $O(\cdot)$ sense:*

$$\begin{cases} \left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3} & \text{if } n \gtrsim \frac{k \log(k)}{\varepsilon^4} \\ \frac{\sqrt{nk \log(k)}}{\varepsilon^2} & \text{if } k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4} \\ \frac{k \log(k)}{\varepsilon^2} & \text{if } n \lesssim k \log(k) \end{cases}$$

Proof We establish this result by invoking LFHT multiple times, and using its sample complexity bounds given in Table 2 in (Gerber et al., 2023). The full proof is deferred to Appendix C.1. \blacksquare

We now present the Empirical Subset Weighting Tester (ESW-Tester) (the pseudocode can be found in Appendix C.2). This adapts the test proposed by Acharya et al. (2022), which was designed to take in two sets of samples of size s_1 and s_2 from a distribution D to determine whether $D = U$ or $d_{TV}(D, U) \geq \varepsilon$. We adapt the test so that we can draw s_1 samples from P and s_2 samples from D to distinguish between $D = P$ and $D = U$.

Lemma 2 *ESW-Tester takes in sample access to k distributions $\{D_i\}_{i \in [k]}$ over $[n]$, parameter $\varepsilon \gtrsim 1/n^{1/4}$, and sample access to a distribution P . It is promised that for an unknown set $\chi \subseteq [k]$, $D_i = P$ for $i \in \chi$, and $D_i = U$ for $i \in [k] \setminus \chi$, with $d_{TV}(P, U) \geq \varepsilon$. ESW-Tester outputs χ with probability at least $\frac{8}{9}$, and the sample complexity is*

$$\begin{cases} O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } n \gtrsim \frac{k \log(k)}{\varepsilon^4} \\ O\left(\frac{k \log(k)}{\varepsilon^4}\right) & \text{if } n \lesssim \frac{k \log(k)}{\varepsilon^4} \end{cases}.$$

The correctness of the ESW-Tester relies on the observation that samples drawn from a non-uniform distribution P tend to collide on ‘‘heavy’’ elements. Consequently, the set of unique elements S constructed in the first step will carry significantly more probability mass under P than under the uniform distribution U .

The detailed analysis, provided in Appendix C.2, formalizes this intuition using two key technical results:

- A separation lemma (Lemma 11, adapted from (Canonne, 2022)) which quantifies the gap between the expected mass of S under P versus U .
- A majorization argument (Lemma 12) which ensures that the mass of S under U , remains bounded with high probability, even when S is generated from P .

Given this separation, the algorithm reduces the problem to a simple mean estimation task: for each D_j , it estimates the mass $D_j(S)$ using s_2 samples. If $D_j = P$, the estimate will be high; if $D_j = U$, it will be low. The sample complexities s_1 (for constructing S) and s_2 (for testing) are optimized to minimize the total cost $s_1 + k s_2$, yielding the two regimes stated in the lemma.

We now put the above findings together and state the resulting theorem, and its associated sample complexity for ‘Find All’. The derivation of this sample complexity is given in Appendix C.3.

Theorem 3 *Let P be a distribution such that $d_{\text{TV}}(P, U) \geq \varepsilon$. There exists an algorithm with sample access to P that, with probability at least $\frac{8}{9}$, correctly classifies all distributions $\{D_i\}_{i=1}^k$ as P or U . The total sample complexity is as follows:*

$$\begin{cases} O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } n \gtrsim k \log(k) \\ O\left(\frac{k \log(k)}{\varepsilon^2}\right) & \text{if } n \lesssim k \log(k). \end{cases}$$

Combining the Two Stages The total error of the two stages is bounded by the sum of probabilities of all the bad events: $\Pr[E_1] + \Pr[E_2] + \Pr[\text{Stage 2 Algorithm Fails}] \leq \frac{1}{9} + \frac{1}{9} + \frac{1}{9} = \frac{1}{3}$. Combining the algorithms for the two stages gives us the upper bounds stated in Table 1, and we state this formally as Theorem 14 in Appendix C.4.

3.2. Both Unknown

In the case that both distributions are unknown, the naive upper bound, by performing $k - 1$ independent equivalence tests (Table 2) is:

$$O\left(k \log(k) \cdot \max\left(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}\right)\right).$$

We improve upon this using similar ideas as our two-stage approach from Section 3.1. The first stage involves finding one distribution of each type, and the second stage partially learns both the distributions and then uses that to classify each remaining distribution.

Stage 1: Find One of Each Suppose that D_1 has the majority distribution P ; if not, a similar argument applies with the roles of P and Q swapped. Since r of the k distributions are Q , we expect to have Q appear once in a uniformly random sample of $\frac{k}{r}$ distributions. Formally, let $\mathcal{I} \subseteq [k]$ be a uniformly random subset of size $\lceil \frac{3k}{r} \rceil$, and let E_1 be the event that there is at least one Q in \mathcal{I} . By a standard tail bound, we upper bound the failure probability by $\Pr[\overline{E_1}] \leq e^{-3} \leq \frac{1}{9}$.

To actually find Q , we will adapt the *unequal samples equivalence test* from (Diakonikolas and Kane, 2016). In our version, we draw s_1 samples from P (via D_1) to create a ‘sketch’ set, which

we then use to test the remaining distributions, drawing s_2 samples from each. Since the algorithm and proof follow from the ones presented in (Diakonikolas and Kane, 2016), we defer the details to Appendix C.5. The sample complexity of this test is:

$$O\left(\max\left(\left(\frac{nk}{r\varepsilon^2}\right)^{2/3}, \frac{k\sqrt{n}}{r\varepsilon^2}\right)\log(k)\right).$$

We choose s_1 and s_2 such that each test is erroneous with probability at most $\frac{1}{9 \cdot \lceil 3k/r \rceil}$, and we denote the event that at least one test fails by E_2 . Using the union bound over $\lceil \frac{3k}{r} \rceil$ equivalence tests, we obtain $\Pr[E_2] \leq \frac{1}{9}$. We will henceforth assume $\overline{E_2}$ occurs.

Stage 2: Find All Assuming success in Stage 1, we have identified two distributions as being P and Q . We use the MultiLFHT algorithm from Theorem 1 to classify the remaining $k - 2$ distributions into their respective clusters with probability at least $\frac{8}{9}$.

Combining the Two Stages The total error probability of the two-stage algorithm bounded by the sum of all failure events: $\Pr[\overline{E_1}] + \Pr[E_2] + \Pr[\text{Stage 2 Fails}] \leq \frac{1}{9} + \frac{1}{9} + \frac{1}{9} = \frac{1}{3}$. The sum of the costs from Stage 1 and Stage 2 gives us the upper bound as stated in Case 2 of Table 1. The formal statement of these bounds is given as Theorem 17 in Appendix C.6.

4. 2-Clustering Lower Bounds

We will present two lower bounds, one in Section 4.1 and another in Section 4.2, which will be relevant for different parameter regimes. The lower bound provided in Section 4.1 can be roughly interpreted as a lower bound ‘against’ Stage 1 of our algorithm, whereas the lower bound presented in Section 4.2 is a bound ‘against’ Stage 2. These two lower bounds will imply all of our lower bounds in Table 1. We reiterate that all of our lower bounds hold even when the cluster sizes (r and $k - r$) are known by the algorithm, and even when the algorithm may be fully adaptive.

In Section 4.1, we establish a lower bound of $\Omega\left(\frac{k\sqrt{n}}{r\varepsilon^2}\right)$ for the *One-Known-One-Unknown* case, and a lower bound of $\Omega\left(\max\left\{\left(\frac{nk}{r\varepsilon^2}\right)^{2/3}, \frac{k\sqrt{n}}{r\varepsilon^2}\right\}\right)$ for the *Both-Unknown* case. Both bounds follow from a common framework: We reduce *from* uniformity/equivalence testing to the task of determining an index of a distribution belonging to a particular cluster—an easier problem than the full clustering task—thereby inheriting the known lower bounds for UST.

In Section 4.2, we establish a lower bound of $\Omega\left(\frac{\sqrt{nk}}{\varepsilon^2}\right)$ for the *One-Known-One-Unknown* case, and a lower bound of $\Omega\left(\max\left\{\frac{k+\sqrt{nk}}{\varepsilon^2}, \left(\frac{kn^2}{\varepsilon^4}\right)^{1/3}\right\}\right)$ for the *Both-Unknown* case. These bounds also arise from a reduction, but a different one from that of Section 4.1. Specifically, here we reduce from known lower bound results in likelihood-free hypothesis testing (LFHT), showing that any algorithm for our clustering problem can be converted into an algorithm for LFHT.

4.1. Clustering Lower Bounds via Unequal Sample Distribution Testing

As mentioned before, we prove our first lower bounds by showing lower bounds for an easier task: identifying the index of a *single* distribution from a specific cluster. When one distribution is known (assumed uniform), this reduces to finding the index of any non-uniform distribution. When both

distributions are unknown, it reduces to identifying the index of a distribution from the smaller cluster (here we assume $r \leq \frac{k}{60}$, so the clusters are of unequal size).

Theorem 4 *Let $k \geq 60$ and $1 \leq r \leq \frac{k}{60}$. Consider a set of k distributions $\{D_i\}_{i \in [k]}$, where for an unknown subset $\mathcal{I} \subseteq [k]$ of size r , we have $D_i = P$ for all $i \in \mathcal{I}$ and $D_i = Q$ for all $i \in [k] \setminus \mathcal{I}$, with $d_{\text{TV}}(P, Q) \geq \varepsilon$. Any algorithm (possibly adaptive and/or knowing r) that returns an index in \mathcal{I} with probability at least 0.9 requires a number of samples lower bounded as follows:*

- (a) **(One Known)** *If Q is known to be uniform, then the required number of samples is $\Omega\left(\frac{k\sqrt{n}}{r\varepsilon^2}\right)$.*
- (b) **(Both Unknown)** *If both P and Q are unknown, the required number of samples is $\Omega\left(\max\left\{\frac{k\sqrt{n}}{r\varepsilon^2}, \left(\frac{nk}{r\varepsilon^2}\right)^{2/3}\right\}\right)$.*

These bounds are obtained via reductions from unequal sample testing (UST). Formally, $\text{UST}(P, Q, s_1, s_2)$ is the problem of distinguishing with probability at least 0.9 between $H_0 : P = Q$ and $H_1 : d_{\text{TV}}(P, Q) \geq \varepsilon$ given s_1 samples from P and s_2 from Q . When Q is known to be the uniform distribution U , we denote the problem by $\text{UST}(P, U, s_1, s_2)$. These are the standard problems of equivalence and uniformity testing in the unequal-sample setting [Bhattacharya and Valiant \(2015\)](#); [Canonne \(2022\)](#). The required sample complexity lower bounds are stated as follows.

Lemma 5 *Given $d_{\text{TV}}(P, U) \geq \varepsilon$ and $d_{\text{TV}}(P, Q) \geq \varepsilon$:*

1. $\text{UST}(P, U, s_1, s_2)$ *requires that: $s_2 \gtrsim \frac{\sqrt{n}}{\varepsilon^2}$, (Thm. 4 in [Paninski \(2008\)](#)).*
2. $\text{UST}(P, Q, s_1, s_2)$ *requires that: $s_1 s_2^2 \gtrsim \left(\frac{n}{\varepsilon^2}\right)^2$ $s_2 \gtrsim \frac{\sqrt{n}}{\varepsilon^2}$, (Thm. 1 in [Bhattacharya and Valiant \(2015\)](#)).*

Using this, we prove both parts of [Theorem 4](#) in [Appendix E.1](#). To give an overview of our reduction, here we sketch the argument for the *One-Known-One-Unknown* case with $r = 1$, i.e., exactly one index corresponds to an unknown distribution that is far from uniform and the remaining $k - 1$ distributions are uniform. First, consider the null instance where all distributions are uniform. By an averaging argument, there must exist an index j whose corresponding distribution is sampled from in expectation at most $O(T/k)$ times and is output with probability at most $O(1/k)$.

Now set the distribution D_j at index j to P (P is an instance in $\text{UST}(P, U, s_1, s_2)$), so P can either be U or far from U). Correctness requires the algorithm to output j if P is non-uniform with high probability (whereas in the null case j is output with probability at most $O(1/k)$). Moreover, we will abort the algorithm as soon as D_j is sampled CT/k times for some suitably-chosen constant $C > 1$. Note that if P is far from uniform, then with high probability, either j is output or D_j gets sampled CT/k times. Moreover, if P is uniform, then with high probability, index j is not output and D_j gets sampled fewer than CT/k times. Finally, combining that the distribution D_j is always sampled at most CT/k times with the first part of [Lemma 5](#), we must have $CT/k \gtrsim \frac{\sqrt{n}}{\varepsilon^2}$ implying $T \gtrsim k\sqrt{n}/\varepsilon^2$.

4.2. Clustering Lower Bounds via LFHT Lower Bounds

Our second set of lower bounds is stated as follows.

Theorem 6 *Let $n \geq 1$, $k \geq 100$, and $\varepsilon \in (0, 1]$. Consider a set of k distributions $\{D_i\}_{i \in [k]}$ where for an unknown subset $\mathcal{I} \subseteq [k]$ of size r , we have $D_i = P$ for all $i \in \mathcal{I}$ and $D_i = Q$ for all $i \in [k] \setminus \mathcal{I}$ with $d_{\text{TV}}(P, Q) \geq \varepsilon$. Any algorithm (possibly adaptive and/or knowing r) that returns the partition $\{\mathcal{I}, [k] \setminus \mathcal{I}\}$ with probability at least 0.9 must have a total sample complexity of:*

- (a) $\Omega\left(\frac{k+\sqrt{nk}}{\varepsilon^2}\right)$, when one of the underlying distributions is uniform.
- (b) $\Omega\left(\left(\frac{kn^2}{\varepsilon^4}\right)^{1/3} + \frac{k+\sqrt{nk}}{\varepsilon^2}\right)$ when both underlying distributions are unknown.

We briefly outline the proof here, and provide the details in Appendix E.2. As mentioned earlier, these bounds are obtained via reductions from *likelihood-free hypothesis testing* (LFHT). Recall that $\text{LFHT}(P, Q, s_1, s_2)$ is the problem of distinguishing, with probability at least 0.9, between $H_0 = P^{\otimes s_1} \otimes Q^{\otimes s_1} \otimes P^{\otimes s_2}$ and $H_1 = P^{\otimes s_1} \otimes Q^{\otimes s_1} \otimes Q^{\otimes s_2}$ where P, Q are distributions with $d_{\text{TV}}(P, Q) \geq \varepsilon$. Equivalently, we get s_1 samples from each of P and Q , and s_2 samples from a distribution $D \in \{P, Q\}$, and must decide (w.p. ≥ 0.9) whether $D = P$ or $D = Q$. When Q is known to be the uniform distribution U , $\text{LFHT}(P, Q, s_1, s_2)$ reduces to distinguishing $H_0 = Q^{\otimes s_1} \otimes U^{\otimes s_2}$ from $H_1 = Q^{\otimes s_1} \otimes Q^{\otimes s_2}$; we denote this case by $\text{LFHT}(P, U, s_1, s_2)$.

Gerber and Polyanskiy (2024) present tight bounds for these two problems, and we state the relevant lower bounds in the following lemma, again using the notation $A \gtrsim B$ for $A \geq cB$.

Lemma 7 *Given $d_{\text{TV}}(P, U) \geq \varepsilon$ and $d_{\text{TV}}(P, Q) \geq \varepsilon$, and assuming the restriction $s_1 \gtrsim s_2$:*

1. $\text{LFHT}(P, U, s_1, s_2)$ requires the following constraints: $s_1 \gtrsim \frac{\sqrt{n}}{\varepsilon^2}$, $s_1 s_2 \gtrsim \frac{n}{\varepsilon^4}$, and $s_2 \gtrsim \frac{1}{\varepsilon^2}$.
2. $\text{LFHT}(P, Q, s_1, s_2)$ requires

$$\begin{cases} s_1^2 s_2 \gtrsim \left(\frac{n}{\varepsilon^2}\right)^2 & s_1 s_2 \gtrsim \frac{n}{\varepsilon^4} & s_2 \gtrsim \frac{1}{\varepsilon^2} & \text{if } n \gtrsim \frac{1}{\varepsilon^4} \\ s_1 \gtrsim \frac{\sqrt{n}}{\varepsilon^2} & s_1 s_2 \gtrsim \frac{n}{\varepsilon^4} & s_2 \gtrsim \frac{1}{\varepsilon^2} & \text{if } n \lesssim \frac{1}{\varepsilon^4} \end{cases}$$

Remark 8 *The first bound is inferred from the construction of P and Q in Section B.3 of Gerber and Polyanskiy (2024), where Q is taken to be the uniform distribution, and the second is from Table 2 of Gerber et al. (2023), where the symbols n, m , and k map to s_1, s_2 and n in our paper. We state the inequalities in Lemma 7 under the additional constraint $s_1 \gtrsim s_2$, as this will turn out to be automatically satisfied in our reduction argument.*

Our proof of Theorem 6 is based on a switching argument. Note that the algorithm returns a partition of size $(r, k - r)$. Let \mathcal{I} denote the subset of size $k - r$.

We first define a null instance and identify an index j whose corresponding distribution is sampled only a small number of times (in expectation) and also has a small probability of appearing in the output set \mathcal{I} . We then replace the distribution at this index with D , from an $\text{LFHT}(P, Q, s_1, s_2)$ instance, where D is either P or Q .

Depending on whether $D = P$ or $D = Q$, the algorithm behaviour differs, either in the number of times the distribution at index j is sampled from, or in whether j is included in the output set \mathcal{I} . This difference is used to detect whether D is P or Q . Moreover, distribution at index j (note that this is unknown D) is sampled from at most $80T/k$ times, while every other distribution is sampled from at most a total of T times. Substituting $s_1 = T$ (the total number of samples drawn from P and Q) and $s_2 = 80T/k$ (the total number of samples drawn from unknown D) into Lemma 7 yields the desired lower bound on the sample complexity T .

5. Extension to d -Clustering

Our results naturally generalize to the problem of clustering into d groups for any constant $d > 2$. We consider a setting with d total clusters, comprising d_{kn} known centers and d_{un} unknown centers (where $d = d_{\text{un}} + d_{\text{kn}}$).

The algorithmic strategy generalizes our two-stage approach: first, finding d_{un} unknown exemplars using a “coupon collector” strategy combined with identity and unequal-samples equivalence testing; and second, running a tournament of pairwise tests to classify the remaining distributions.

Theorem 9 (Sample Complexity of d -Clustering) *There exists an algorithm that solves the d -clustering problem with probability at least $2/3$, and such that for $d_{\text{un}} = 1$ (resp., $d_{\text{un}} \geq 2$), the sample complexity is as given in Case 1 (resp., Case 2) of Table 1, up to constant factors that may depend on d , where r is the size of the smallest cluster among those with unknown distributions.*

We now argue that this sample complexity is tight (up to logarithmic factors). This is because the 2-clustering problem is a special case of d -clustering, so the lower bounds established in Section 4 apply immediately. Consequently, for any constant d , the sample complexity of d -clustering is governed by the same tight dependencies on n , k , r , and ε as the 2-clustering case; recall that this tightness is verified for 2-clustering in Appendix F.

The formal algorithms and proofs for this extension are provided in Appendix D.

6. Conclusion and Future Work

We have provided upper and lower bounds for d -clustering of distributions, with near-matching scaling in all parameter regimes. Among other things, our results imply the following non-obvious insight: *it is \sqrt{k} times harder to find a loaded die in a bag of fair dice than it is to find a fair die in a bag of loaded dice.*

We conclude by listing some open problems that arise naturally:

- **Dependence on the failure probability.** As discussed in Appendix A, it remains open to establish the optimal sample complexity when the dependence on the failure probability δ is explicitly incorporated.
- **Optimal bounds for d -clustering for $d = \omega(1)$.** While we have provided an optimality result for d -clustering with $d = O(1)$, determining the optimal dependence on the number of clusters d remains an open question for $d \in \omega(1)$.
- **Adaptivity gap.** Our algorithm crucially relies on adaptive sampling in order to find a “representative” from one cluster (or both) and then use it to classify the remaining distributions. It would be of interest to determine whether adaptivity is essential, i.e., whether non-adaptive sampling requires strictly more samples, thus creating an “adaptivity gap”.

Acknowledgments

This work is supported by the Singapore National Research Foundation (NRF) under its AI Visiting Professorship programme.

References

- Jayadev Acharya, Clément L. Canonne, Yuhan Liu, Ziteng Sun, and Himanshu Tyagi. Interactive inference under information constraints. *IEEE Transactions on Information Theory*, 68:502–516, 2022.
- Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Symposium on Foundations of Computer Science (FOCS)*, pages 259–269. IEEE, 2000.
- Bhaswar Bhattacharya and Gregory Valiant. Testing closeness with unequal sized samples. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015.
- Clément L. Canonne. *A Survey on Distribution Testing: Your Data is Big. But is it Blue?* Graduate Surveys. Theory of Computing Library, 2020.
- Clément L. Canonne. Topics and techniques in distribution testing: A biased but representative sample. *Foundations and Trends® in Communications and Information Theory*, 19, 2022.
- Siu-On Chan, Ilias Diakonikolas, Paul Valiant, and Gregory Valiant. Optimal algorithms for testing closeness of discrete distributions. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1193–1203. SIAM, 2014.
- Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*. Springer Science & Business Media, 2001.
- Ilias Diakonikolas and Daniel M Kane. A new approach for testing properties of discrete distributions. In *Symposium on Foundations of Computer Science (FOCS)*, 2016.
- Patrik R Gerber, Yanjun Han, and Yury Polyanskiy. Minimax optimal testing by classification. In *Conference on Learning Theory (COLT)*, pages 5395–5432. PMLR, 2023.
- Patrik Róbert Gerber and Yury Polyanskiy. Likelihood-free hypothesis testing. *IEEE Transactions on Information Theory*, 2024.
- Oded Goldreich. The uniform distribution is complete with respect to testing identity to a fixed distribution. In *Computational Complexity and Property Testing: On the Interplay Between Randomness and Computation*, pages 152–172. Springer, 2020.
- Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45:653–750, 1998.
- Michael Gutman. Asymptotically optimal classification for multiple tests with empirically observed statistics. *IEEE Transactions on Information Theory*, 35(2):401–408, 2002.
- Reut Levi, Dana Ron, and Ronitt Rubinfeld. Testing properties of collections of distributions. *Theory of Computing*, 9:295–347, 2013.
- Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

- Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. *Inequalities: Theory of Majorization and its Applications*. Springer, second edition, 2011.
- Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 2008.
- Henry Scheffé. A useful convergence theorem for probability distributions. *The Annals of Mathematical Statistics*, 18, 1947.
- Gregory Valiant and Paul Valiant. The power of linear estimators. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2011.
- Junwen Yang, Zixin Zhong, and Vincent YF Tan. Optimal clustering with bandit feedback. *Journal of Machine Learning Research (JMLR)*, 2024.
- Recep Can Yavas, Yuqi Huang, Vincent YF Tan, and Jonathan Scarlett. A general framework for clustering and distribution matching with bandit feedback. *IEEE Transactions on Information Theory*, 2025.
- Yubo Zhuang, Xiaohui Chen, and Yun Yang. Wasserstein k -means for clustering probability distributions. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.
- Jacob Ziv. On classification with empirically observed statistics and universal data compression. *IEEE Transactions on Information Theory*, 34(2):278–286, 2002.

Appendix A. Comparison to Bandit Clustering Results

In this section, we provide a more detailed discussion of how our results differ from those of (Yavas et al., 2025), which gave a general matching/clustering framework that includes 2-clustering as a special case.

We first highlight that (Yavas et al., 2025) provides tight results in the asymptotic limit $\delta \rightarrow 0$ (with all other parameters held fixed), which is fundamentally different from our attention on constant error probability. To appreciate this distinction, we note that in the simpler uniformity testing problem with domain size n and TV distance ε , the optimal scaling in general is $\Theta\left(\frac{1}{\varepsilon^2} \left(\sqrt{n \log \frac{1}{\delta}} + \log \frac{1}{\delta}\right)\right)$ (Canonne, 2022, Eq. (2.50)). Thus, in the limit $\delta \rightarrow 0$, the leading term is $\frac{1}{\varepsilon^2} \log \frac{1}{\delta}$. However, taking this limit neglects the important \sqrt{n} dependence in the general bound. For instance, one could have derived an upper bound with $\sqrt{n \log \frac{1}{\delta}}$ replaced by $n \sqrt{\log \frac{1}{\delta}}$ or even $2^n \sqrt{\log \frac{1}{\delta}}$, and it would still have been asymptotically optimal in the limit $\delta \rightarrow 0$, yet highly sub-optimal for constant δ .

Let $\mathbf{D} = (D_1, \dots, D_k)$ be the k distributions in the 2-clustering problem. The (expected) sample complexity derived in (Yavas et al., 2025, Thm. 2) is instance dependent (i.e., depends on \mathbf{D}), and

takes the following form as $\delta \rightarrow 0$:

$$\mathbb{E}[\#\text{samples}] = \frac{1}{T^*(\mathbf{D})} \log \frac{1}{\delta} \cdot (1 + o(1)), \quad (1)$$

$$T^*(\mathbf{D}) = \sup_w \inf_{\mathbf{D}' \in \text{Alt}(\mathbf{D})} \sum_{i=1}^k w_i d_{\text{KL}}(D_i, D'_i), \quad (2)$$

where $w \in [0, 1]^k$ is optimized over the probability simplex, and $\text{Alt}(\mathbf{D})$ is the set of alternative instances whose correct answer is different from \mathbf{D} . We focus here on the case of known cluster sizes, meaning that both \mathbf{D} and \mathbf{D}' have r clusters of one type and $k - r$ of the other.

We can simplify $T^*(\mathbf{D})$ for 2-clustering by noting the following:

- By the non-negativity of KL divergence, the minimizing \mathbf{D}' swaps just two distributions (one from each cluster) with one another.
- The maximizing w places equal probability within the r distributions in one cluster, and also within the $k - r$ distributions in the other cluster. This is because, with the intra-cluster distributions being identical, the $\inf_{\mathbf{D}'}$ will be achieved by picking the minimum-weight element of each cluster; then, using uniform weights ensures that each cluster's minimum weight is as high as possible.

Accordingly, we fix $\lambda \in [0, 1]$ and assign weights $\frac{\lambda}{r}$ and $\frac{1-\lambda}{k-r}$ within the size- r and size- $(k - r)$ clusters respectively. Then, the quantity $\sum_{i=1}^k w_i d_{\text{KL}}(D_i, D'_i)$ in (2) becomes $\frac{\lambda}{r} d_{\text{KL}}(P, Q) + \frac{1-\lambda}{k-r} d_{\text{KL}}(Q, P)$ with P and Q being the two clusters' distributions. Applying Pinsker's inequality and $d_{\text{TV}}(P, Q) \geq \varepsilon$ gives $d_{\text{KL}}(\cdot, \cdot) \geq 2\varepsilon^2$, so the preceding expression is lower bounded by $2\varepsilon^2 \left(\frac{\lambda}{r} + \frac{1-\lambda}{k-r} \right)$. By handling the cases $r < \frac{k}{2}$ and $r > \frac{k}{2}$ separately, we readily obtain that this behaves as $\Theta\left(\frac{\varepsilon^2}{k}\right)$ upon maximizing over λ (in accordance with the \sup_w operation), and it follows that the sample complexity is upper bounded by $O\left(\frac{k}{\varepsilon^2} \log \frac{1}{\delta}\right)$ as $\delta \rightarrow 0$.

In analogy with the earlier discussion on uniformity testing, this asymptotic analysis hides important dependencies – the result shows no dependence on the domain size n or cluster size r , because any such dependence is hidden in lower-order terms in δ . Moreover, an inspection of the proof in (Yavas et al., 2025) reveals that their analysis is unsuitable for optimizing such dependencies; for instance, the proof of Lemma 14 therein is based on showing that a quantity of the form $Ae^{-B \log^2 t}$ approaches zero for all $t \geq \Omega\left(\log \frac{1}{\delta}\right)$, where A and B are complicated functions of the problem parameters. We also note that a linear dependence on the domain size (which can often be avoided via distributed testing methods; see Table 2) appears much earlier in their analysis, e.g., see Eq. (66) therein.

On the other hand, it is worth highlighting that the above-mentioned results from (Yavas et al., 2025) indicate that our approach of amplifying a $2/3$ -success guarantee to $1 - \delta$ (with $O\left(\log \frac{1}{\delta}\right)$ multiplicative overhead) is *not* optimal in general. Thus, it remains an open problem to understand the joint dependencies on all parameters *including* δ , thus interpolating between our constant- δ results and the existing (very) small- δ results.

Appendix B. Identity To Uniformity Reduction

The Identity To Uniformity reduction transforms the problem of testing identity to a known distribution D into the problem of testing uniformity Goldreich (2020). The reduction is a randomised pro-

cedure that maps the original domain $[n]$ to a domain $[n']$ of size $n' = 6n$. This mapping constructs a transformed distribution P' from the unknown distribution P such that the reference distribution D maps exactly to the uniform distribution U_{6n} .

Theorem 10 (Theorem 1 from (Goldreich, 2020)) *For every distribution P over $[n]$ and every $\varepsilon > 0$, determining $d_{\text{TV}}(P, Q) = 0$ vs. $d_{\text{TV}}(P, Q) \geq \varepsilon$ reduces to testing $d_{\text{TV}}(P', U_{6n}) = 0$ vs. $d_{\text{TV}}(P', U_{6n}) \geq \varepsilon/3$, where U_m denotes the uniform distribution over $[m]$. Furthermore, the same reduction F can be used for all $\varepsilon > 0$.*

Theorem 10 guarantees that this transformation preserves the distance separation required for testing:

- If $P = D$, then $P' = U_{6n}$.
- If $d_{\text{TV}}(P, D) \geq \varepsilon$, then $d_{\text{TV}}(P', U_{6n}) \geq \varepsilon/3$.

Consequently, any algorithm that tests uniformity over $[6n]$ with distance parameter $\varepsilon/3$ also tests identity to D over $[n]$ with parameter ε . This allows the use of standard uniformity testing bounds for the general identity testing problem.

Appendix C. Derivation of Algorithmic Upper Bounds

This section provides a detailed derivation of the sample complexity upper bounds for the algorithms presented in Section 3. We show how the complexities of the two stages combine under different parameter regimes.

C.1. Proof of Theorem 1 (Sample Complexity of MultiLFHT)

The MultiLFHT procedure classifies a set of k distributions $\{D_i\}_{i=1}^k$, each promised to be either P or Q , by applying a Likelihood-Free Hypothesis Testing (LFHT) subroutine to each one. To achieve the target total error probability of at most $\frac{1}{9}$ even after a union bound, the error tolerance for each of the k individual classification tests is set to $\delta = \frac{1}{9k}$. The procedure involves drawing a number of samples (denoted s_1) from the exemplar distributions P and Q , and a number of samples (denoted s_2) from each distribution D_i that is being classified. We state the sample complexity as follows:

Theorem 1 *Let P and Q be two distributions with $d_{\text{TV}}(P, Q) \geq \varepsilon$. Given sample access to P and Q , there exists a procedure (which we refer to as MultiLFHT) that, with probability at least $\frac{8}{9}$, correctly classifies distributions $\{D_i\}_{i=1}^k$ as P or Q . The sample complexity is as follows in the $O(\cdot)$ sense:*

$$\begin{cases} \left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3} & \text{if } n \gtrsim \frac{k \log(k)}{\varepsilon^4} \\ \frac{\sqrt{nk \log(k)}}{\varepsilon^2} & \text{if } k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4} \\ \frac{k \log(k)}{\varepsilon^2} & \text{if } n \lesssim k \log(k) \end{cases}$$

Proof The proof is based on choosing parameters to minimize the total ‘‘cost’’ (number of samples) $C = s_1 + k s_2$ subject to suitable constraints in various scaling regimes. To achieve a total error probability of at most $\frac{1}{9}$, we set the per-distribution classification error to $\delta = \frac{1}{9k}$ and apply a union

bound. The necessary sample sizes must satisfy several constraints given in Table 2 of (Gerber et al., 2023). To state the bounds in the language of our paper, we replace the symbols (n, m, k) in their paper with (s_1, s_2, n) .

The first constraint is common for all parameter regimes:

$$s_2 \gtrsim \frac{\log(k)}{\varepsilon^2} \quad (3)$$

The other constraints depend on the domain size n :

$$\left(\text{if } n \gtrsim \frac{\log(k)}{\varepsilon^4}\right) \quad s_1^2 s_2 \gtrsim \frac{n^2 \log(k)}{\varepsilon^4} \quad \text{and} \quad s_1 s_2 \gtrsim \frac{n \log(k)}{\varepsilon^4} \quad (4)$$

$$\left(\text{if } n \lesssim \frac{\log(k)}{\varepsilon^4}\right) \quad s_1 s_2 \gtrsim \frac{(\sqrt{n \log(k)} + \log(k))^2}{\varepsilon^4} \quad \text{and} \quad s_1 \gtrsim \frac{\sqrt{n \log(k)} + \log(k)}{\varepsilon^2} \quad (5)$$

The optimal strategy for choosing s_1 and s_2 changes depending on which constraint is dominant, leading to three distinct regimes for the total complexity.

Regime 1: $n \gtrsim \frac{k \log(k)}{\varepsilon^4}$ In this regime, the $s_1^2 s_2$ constraint from (4) is the more restrictive of the two. The cost $C = s_1 + k s_2$ is minimized by balancing the two terms, leading to the choice $s_1 \asymp \left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}$ and $s_2 \asymp \left(\frac{n^2 \log(k)}{k^2 \varepsilon^4}\right)^{1/3}$. The total cost is $O\left(\left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}\right)$, and all constraints are satisfied, as we verify as follows:

- The $s_1^2 s_2$ constraint is met by construction.
- The $s_1 s_2$ constraint requires $\left(\frac{n^4 \log^2(k)}{k \varepsilon^8}\right)^{1/3} \gtrsim \frac{n \log(k)}{\varepsilon^4}$, which simplifies to $n \gtrsim \frac{k \log(k)}{\varepsilon^4}$, which in turn holds by the definition of the regime.
- The s_2 constraint requires $\left(\frac{n^2 \log(k)}{k^2 \varepsilon^4}\right)^{1/3} \gtrsim \frac{\log(k)}{\varepsilon^2}$, which simplifies to $n \gtrsim \frac{k \log(k)}{\varepsilon}$ and thus also holds.

Regime 2: $k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4}$ Since $n \gtrsim k \log(k)$, we have $\sqrt{n \log(k)} \gtrsim \log(k)$, and thus the $s_1 s_2$ constraint from (5) becomes $s_1 s_2 \gtrsim \frac{n \log(k)}{\varepsilon^4}$, matching (4). Our problem is to minimize $C = s_1 + k s_2$ subject to all the relevant constraints. We minimize the cost by balancing the terms, $s_1 \asymp k s_2$.

- Substituting $s_1 \asymp k s_2$ into the $s_1 s_2$ constraint gives

$$(k s_2) s_2 \gtrsim \frac{n \log(k)}{\varepsilon^4} \iff s_2^2 \gtrsim \frac{n \log(k)}{k \varepsilon^4} \implies \text{Can set } s_2 \asymp \frac{1}{\varepsilon^2} \sqrt{\frac{n \log(k)}{k}}.$$

- This determines s_1 as follows:

$$s_1 \asymp k s_2 \asymp \frac{\sqrt{n k \log(k)}}{\varepsilon^2}.$$

We verify this solution against all constraints:

- The s_1 constraint from (5) is satisfied as $s_1 \asymp \frac{\sqrt{n k \log(k)}}{\varepsilon^2} \gtrsim \frac{\sqrt{n \log(k)} + \log(k)}{\varepsilon^2}$.

- The $s_1 s_2$ constraint from (4) is met by construction: $s_1 s_2 \asymp \left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2} \right) \left(\frac{1}{\varepsilon^2} \sqrt{\frac{n \log(k)}{k}} \right) = \frac{n \log(k)}{\varepsilon^4}$.

- The s_2 constraint from (3) requires $s_2 \gtrsim \frac{\log(k)}{\varepsilon^2}$. Our choice $s_2 \asymp \frac{1}{\varepsilon^2} \sqrt{\frac{n \log(k)}{k}}$ satisfies this, since:

$$\frac{1}{\varepsilon^2} \sqrt{\frac{n \log(k)}{k}} \gtrsim \frac{\log(k)}{\varepsilon^2} \iff \sqrt{\frac{n}{k \log(k)}} \gtrsim 1 \iff n \gtrsim k \log(k),$$

which holds by the definition of the regime.

- The $s_1^2 s_2$ constraint from (4) requires $s_1^2 s_2 \gtrsim \frac{n^2 \log(k)}{\varepsilon^4}$. We have

$$s_1^2 s_2 \asymp \left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2} \right)^2 \left(\frac{1}{\varepsilon^2} \sqrt{\frac{n \log(k)}{k}} \right) = \left(\frac{nk \log(k)}{\varepsilon^4} \right) \left(\frac{\sqrt{n \log(k)}}{\sqrt{k} \varepsilon^2} \right) = \frac{n \sqrt{nk} \log(k)^{3/2}}{\varepsilon^6}.$$

The constraint is thus satisfied if $\frac{n \sqrt{nk} \log(k)^{3/2}}{\varepsilon^6} \gtrsim \frac{n^2 \log(k)}{\varepsilon^4}$, which simplifies to $\frac{\sqrt{k \log(k)}}{\varepsilon^2} \gtrsim \sqrt{n}$ and thus $\frac{k \log(k)}{\varepsilon^4} \gtrsim n$. This holds by the definition of the regime.

Thus, all constraints are satisfied, and the total cost is $O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right)$.

Regime 3: $n \lesssim k \log(k)$ In this range, we choose $s_2 \asymp \frac{\log(k)}{\varepsilon^2}$ and $s_1 \asymp \frac{k \log(k)}{\varepsilon^2}$. The total cost $C = s_1 + k s_2$ is thus $O\left(\frac{k \log(k)}{\varepsilon^2}\right)$. We verify the constraints as follows:

- The s_2 constraint from (3) is met by construction, as $s_2 \asymp \frac{\log(k)}{\varepsilon^2}$.
- The $s_1 s_2$ constraints from (4) and (5) are met, as $s_1 s_2 \asymp \left(\frac{k \log(k)}{\varepsilon^2} \right) \left(\frac{\log(k)}{\varepsilon^2} \right) \gtrsim \frac{k \log(k)^2}{\varepsilon^4} \gtrsim \frac{n \log(k)}{\varepsilon^4}$ by the definition of the regime.
- The $s_1^2 s_2$ constraint from (4) is met, as $s_1^2 s_2 \asymp \left(\frac{k \log(k)}{\varepsilon^2} \right)^2 \left(\frac{\log(k)}{\varepsilon^2} \right) \gtrsim \frac{n^2 \log(k)}{\varepsilon^4}$. This holds by the definition of the regime.
- The s_1 constraint from (5) requires $s_1 \gtrsim \frac{\sqrt{n \log(k) + \log(k)}}{\varepsilon^2}$, which is satisfied provided that $s_1 \gtrsim \frac{\log(k)}{\varepsilon^2}$ and $s_1 \gtrsim \frac{n}{\varepsilon^2}$ (since $\sqrt{n \log k}$ is the geometric mean of n and $\log k$). The former inequality is trivial, and the latter follows from the definition of the regime.

Combining these results gives the three distinct complexity regimes for the algorithm:

- For $n \gtrsim \frac{k \log(k)}{\varepsilon^4}$, the complexity is $O\left(\left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}\right)$.
- For $k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4}$, the complexity is $O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right)$.
- For $n \lesssim k \log(k)$, the complexity is $O\left(\frac{k \log(k)}{\varepsilon^2}\right)$.

■

C.2. Proof of Lemma 2 (ESW-Tester correctness)

In this section, we detail the ESW-Tester procedure (Algorithm 1) and prove its correctness guarantees stated in Lemma 2.

Algorithm 1: ESW-Tester

Input: $\varepsilon \in (0, 1]$, sample access to distribution P , distributions $\{D_i\}_{i \in [k]}$, such that for $\chi \subseteq [k]$ we have $\{D_i\}_{i \in \chi} = P$, and $\{D_i\}_{i \in [k] \setminus \chi} = U$ with $d_{\text{TV}}(P, U) \geq \varepsilon$.

Output: A set $\hat{\chi} \subseteq [k]$ that, with probability at least $8/9$, satisfies $\hat{\chi} = \chi$.

```

 $s_1 \leftarrow \min \left( n, \frac{\sqrt{nk \log(k)}}{\varepsilon^2} \right)$ 
 $s_2 \leftarrow c \frac{n \log(k)}{s_1 \varepsilon^4}$  //  $c$  is a universal constant
 $\mathcal{S} \sim P^{\otimes s_1}$ 
 $S \leftarrow \text{set}(\mathcal{S})$ 
 $\tau \leftarrow \frac{s_1 \varepsilon^2}{64n}$ 
 $\hat{\chi} \leftarrow \emptyset$ 
for  $j \in \{1, \dots, k\}$  do
     $\mathcal{S}_j \sim D_j^{s_2}$ 
     $Z_j \leftarrow \frac{1}{s_2} \sum_{i \in \mathcal{S}_j} \mathbb{1}\{i \in S\}$ 
    if  $Z_j \geq 1 - (1 - \frac{1}{n})^{s_1} + 2\tau$  then
         $\hat{\chi} \leftarrow \hat{\chi} \cup j$ 
return  $\hat{\chi}$ 

```

We restate the result as follows, and then proceed with the proof.

Lemma 2 *ESW-Tester takes in sample access to k distributions $\{D_i\}_{i \in [k]}$ over $[n]$, parameter $\varepsilon \gtrsim 1/n^{1/4}$, and sample access to a distribution P . It is promised that for an unknown set $\chi \subseteq [k]$, $D_i = P$ for $i \in \chi$, and $D_i = U$ for $i \in [k] \setminus \chi$, with $d_{\text{TV}}(P, U) \geq \varepsilon$. ESW-Tester outputs χ with probability at least $\frac{8}{9}$, and the sample complexity is*

$$\begin{cases} O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } n \gtrsim \frac{k \log(k)}{\varepsilon^4} \\ O\left(\frac{k \log(k)}{\varepsilon^4}\right) & \text{if } n \lesssim \frac{k \log(k)}{\varepsilon^4} \end{cases}.$$

Let $\mathcal{S} = \{X_1, \dots, X_{s_1}\}$ be a multiset of s_1 samples drawn i.i.d. from a distribution P , a process we denote by $\mathcal{S} \sim P^{\otimes s_1}$. Let $S = \text{set}(\mathcal{S})$ be the set of unique elements in \mathcal{S} (i.e., duplicates are removed). The notation $\Pr_{\mathcal{S} \sim P^{\otimes s_1}}[\cdot]$ denotes the probability of an event where the probability measure is induced by the sampling of \mathcal{S} from P . Such events may be expressed in terms of $S = \text{set}(\mathcal{S})$, in which case the contributions from all such \mathcal{S} yielding that set are summed. Given $\mathcal{S} \sim P^{\otimes s_1}$, a result from (Acharya et al., 2022) establishes that the mass of S is noticeably heavier in P than in U .

We rely on a variation of this result from Lemma 2.18 of (Canonne, 2022). Our restatement omits the constants, and uses a tighter probability upper bound that only amounts to slight changes in those constants.

Lemma 11 (Variation of Lemma 2.18 in (Canonne, 2022)) *For P such that $d_{\text{TV}}(P, U) \geq \varepsilon$ with $\varepsilon \gtrsim 1/n^{1/4}$, and for sets S, S' constructed from s_1 samples, if $\frac{\sqrt{n}}{\varepsilon^2} \lesssim s_1 \lesssim n$ (with suitable constants), then*

$$\begin{aligned} \Pr_{S \sim U^{\otimes s_1}} [U(S) \geq \mathbb{E}_{S' \sim U^{\otimes s_1}}[U(S')] + \tau] &\leq \frac{1}{36} \\ \Pr_{S \sim P^{\otimes s_1}} [P(S) \leq \mathbb{E}_{S' \sim U^{\otimes s_1}}[U(S')] + 3\tau] &\leq \frac{1}{36} \end{aligned}$$

for some τ satisfying $\tau \lesssim \frac{s_1 \varepsilon^2}{n}$, where $U(S) = \sum_{x \in S} U(x)$, $P(S) = \sum_{x \in S} P(x)$, $S = \text{set}(S)$, and $S' = \text{set}(S')$.

The following lemma can be summarized as “non-uniform distributions tend to yield fewer distinct elements (more collisions) than the uniform distribution”.

Lemma 12 *For any $t \in \mathbb{R}$ and distribution P ,*

$$\Pr_{S \sim P^{\otimes s_1}} [U(S) \geq t] \leq \Pr_{S \sim U^{\otimes s_1}} [U(S) \geq t].$$

Proof The proof uses only elementary steps, and is deferred to Appendix G.2. ■

Corollary 13 *Under the setup of Lemma 11, we have*

$$\Pr_{S \sim P^{\otimes s_1}} [U(S) \geq \mathbb{E}_{S' \sim U^{\otimes s_1}}[U(S')] + \tau] \leq \frac{1}{36}.$$

Proof This follows from Lemmas 11 and 12. ■

For a fixed set \tilde{S} , the probability of a sample drawn from D_j being in \tilde{S} is $D_j(\tilde{S})$. We let $Z_j = \frac{1}{s_2} \sum_{t=1}^{s_2} \mathbb{1}\{X_t \in \tilde{S}\}$ denote an estimate of $D_j(\tilde{S})$ based on s_2 samples from D_j . We observe that:

$$\mathbb{E}[Z_j] = \begin{cases} U(\tilde{S}) & \text{if } D_j = U \\ P(\tilde{S}) & \text{if } D_j = P. \end{cases} \quad (6)$$

We now define two events for S generated from P : G_1 is the event that $P(S) \geq \mathbb{E}_{S' \sim U}[U(S')] + 3\tau$, and G_2 the event that $U(S) \leq \mathbb{E}_{S' \sim U}[U(S')] + \tau$. We then have from Lemma 11 that $\Pr[G_1] \geq 35/36$, and from Corollary 13 that $\Pr[G_2] \geq 35/36$. By the union bound, we obtain $\Pr[G_1 \cap G_2] \geq 17/18$. We will henceforth assume $G_1 \cap G_2$ and suitably account for the total error probability later.

We compute the expected mass (under U) of a set of s_1 samples from U as $\mathbb{E}_{S' \sim U^{\otimes s_1}}[U(S')] = 1 - (1 - \frac{1}{n})^{s_1}$ and we denote this as γ ; we note for later use that $\gamma \leq \frac{s_1}{n}$. Substituting the preceding findings into (6) (using S for the generic set \tilde{S}), we find that we have to distinguish between two cases for Z_j , for $\tau \lesssim \frac{s_1 \varepsilon^2}{n}$:

$$\mathbb{E}[Z_j] \leq \gamma + \tau \quad \text{and} \quad \mathbb{E}[Z_j] \geq \gamma + 3\tau$$

Since we have k distributions, we seek to do this with error probability at most $\frac{1}{18k}$ per distribution. This is the same as determining with probability $1 - \frac{1}{18k}$ whether, for a coin with unknown bias

α , we have $\alpha \leq \beta$ vs. $\alpha \geq \beta(1 + \eta)$, where $\beta = \gamma + \tau$ and $\eta = \frac{2\tau}{\beta}$. By the multiplicative Chernoff bound, this requires $\Theta(\frac{\log(1/\delta)}{\beta\eta^2})$ samples for any error probability δ . Since $\delta = \frac{1}{18k}$, and $\beta = \gamma + \tau \lesssim \frac{s_1}{n} \cdot (1 + \varepsilon^2) \lesssim \frac{s_1}{n}$, and $\eta^2 = \frac{4\tau^2}{\beta^2}$, we find that the per-index sample complexity is $s_2 \asymp \frac{\log(18k)}{\beta\eta^2} \asymp \frac{\log(k)n}{s_1\varepsilon^4}$. Let E_3 be the event that at least one of the tests fail. Then, by the union bound, $\Pr[E_3] \leq \frac{1}{18}$. The total failure probability of the tester is bounded by the failure of the ‘‘good set’’ generation and the failure of the Chernoff bounds: $\Pr[\overline{G_1} \cap \overline{G_2}] + \Pr[E_3] \leq \frac{1}{18} + \frac{1}{18} = \frac{1}{9}$. Thus, the success probability is at least $\frac{8}{9}$.

The total sample complexity is $s_1 + k \cdot s_2$. We analyse the two cases from the lemma statement, which are determined by the definition of s_1 :

- **Case 1:** $n \gtrsim \frac{k \log(k)}{\varepsilon^4}$. In this case $s_1 = \frac{\sqrt{nk \log(k)}}{\varepsilon^2}$, and the complexity is

$$s_1 + k \cdot s_2 \asymp \frac{\sqrt{nk \log(k)}}{\varepsilon^2} + k \cdot \frac{n \log(k)}{s_1 \varepsilon^4} \asymp \frac{\sqrt{nk \log(k)}}{\varepsilon^2} + \frac{kn \log(k)}{\frac{\sqrt{nk \log(k)}}{\varepsilon^2} \varepsilon^4} = O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right).$$

- **Case 2:** $n \lesssim \frac{k \log(k)}{\varepsilon^4}$. In this case $s_1 = n$, and the complexity is

$$s_1 + k \cdot s_2 = n + k \cdot \frac{n \log(k)}{n \varepsilon^4} = n + \frac{k \log(k)}{\varepsilon^4} = O\left(\frac{k \log(k)}{\varepsilon^4}\right),$$

where the last step holds by the case condition.

Thus, we have established Lemma 2.

C.3. Proof of Theorem 3 (Sample Complexity of Stage 2, One Known One Unknown)

Theorem 3 *Let P be a distribution such that $d_{\text{TV}}(P, U) \geq \varepsilon$. There exists an algorithm with sample access to P that, with probability at least $\frac{8}{9}$, correctly classifies all distributions $\{D_i\}_{i=1}^k$ as P or U . The total sample complexity is as follows:*

$$\begin{cases} O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } n \gtrsim k \log(k) \\ O\left(\frac{k \log(k)}{\varepsilon^4}\right) & \text{if } n \lesssim k \log(k). \end{cases}$$

Proof Recall that for Stage 2, or the ‘Find All’ phase, we had two possible algorithms, `ESW-Tester` and `MultiLFHT`. We choose the more efficient algorithm of the two, and that is decided based only on the domain size n . Hence, the total sample complexity is the minimum of the sample complexity of the two.

The total sample complexity for the `ESW-Tester` is given by:

$$T_{\text{ESW}} = \begin{cases} O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } n \gtrsim \frac{k \log(k)}{\varepsilon^4} \\ O\left(\frac{k \log(k)}{\varepsilon^4}\right) & \text{if } n \lesssim \frac{k \log(k)}{\varepsilon^4} \end{cases} \quad (\text{From Lemma 2})$$

The complexity for MultiLFHT, is:

$$T_{\text{Mul}} = \begin{cases} O\left(\left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}\right) & \text{if } n \gtrsim \frac{k \log(k)}{\varepsilon^4} \\ O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4} \\ O\left(\frac{k \log(k)}{\varepsilon^2}\right) & \text{if } n \lesssim k \log(k) \end{cases} \quad (\text{From Theorem 1})$$

Recall that Lemma 2 (on ESW-Tester) requires $n \gtrsim \frac{1}{\varepsilon^4}$; we will ensure this condition below. The overall sample complexity of Stage 2 is $T_2 = \min(T_{\text{ESW}}, T_{\text{Mul}})$, which we study through a case analysis.

- **Case 1: Large Domain** ($n \gtrsim \frac{k \log(k)}{\varepsilon^4}$). We compare $T_{\text{ESW}} = O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right)$ with $T_{\text{Mul}} = O\left(\left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}\right)$. To simplify, we compare their 6th powers.

$$\begin{aligned} (T_{\text{ESW}})^6 &\propto \left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right)^6 = \frac{k^3 n^3 (\log k)^3}{\varepsilon^{12}} \\ (T_{\text{Mul}})^6 &\propto \left(\left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}\right)^6 = \frac{n^4 k^2 (\log k)^2}{\varepsilon^8} \end{aligned}$$

Dividing both sides by the common term $\frac{n^3 k^2 (\log k)^2}{\varepsilon^8}$ shows that this comparison is equivalent to comparing $\frac{k \log k}{\varepsilon^4}$ with n . Since $n \gtrsim \frac{k \log k}{\varepsilon^4}$ in this regime, it follows that $T_{\text{ESW}} \lesssim T_{\text{Mul}}$, i.e., ESW-Tester is the better choice.

- **Case 2: Intermediate and Small Domains** ($n \lesssim \frac{k \log(k)}{\varepsilon^4}$). We compare the relevant complexities for T_{Mul} and T_{ESW} in this range by splitting it based on the boundaries for T_{Mul} :
 - If $k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4}$: We compare $T_{\text{ESW}} = O\left(\frac{k \log(k)}{\varepsilon^4}\right)$ with $T_{\text{Mul}} = O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right)$. Plugging the maximum value of n in this regime, $n \asymp \frac{k \log(k)}{\varepsilon^4}$, we see that MultiLFHT is more efficient over the entire regime.
 - If $n \lesssim k \log(k)$: We compare $T_{\text{ESW}} = O\left(\frac{k \log(k)}{\varepsilon^4}\right)$ with $T_{\text{Mul}} = O\left(\frac{k \log(k)}{\varepsilon^2}\right)$. The $1/\varepsilon^4$ term in T_{ESW} ensures it is larger than T_{Mul} .

Hence in both sub-cases of Case 2, MultiLFHT is the better choice.

By combining these results, we find that the complexity from Case 1 and the first part of Case 2 merge into a single regime for $n \gtrsim k \log(k)$. The second part of Case 2 covers $n \lesssim k \log(k)$. The complexity for Stage 2 is therefore:

$$\begin{cases} O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } n \gtrsim k \log(k) \\ O\left(\frac{k \log(k)}{\varepsilon^2}\right) & \text{if } n \lesssim k \log(k) \end{cases}$$

■

C.4. Sample Complexity of Clustering (One Known One Unknown)

Theorem 14 *Our algorithm solves the 2-clustering problem with one known distribution with success probability at least $\frac{2}{3}$. The total sample complexity is given by:*

- For $n \gtrsim k \log(k)$:

$$O\left(\frac{k \log(k) \sqrt{n}}{r \varepsilon^2} + \frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right).$$

- For $n \lesssim k \log(k)$:

$$O\left(\frac{k \log(k) \sqrt{n}}{r \varepsilon^2} + \frac{k \log(k)}{\varepsilon^2}\right).$$

Proof The algorithm for this case proceeds in two stages, and the total sample complexity is the sum of their costs, $T = T_1 + T_2$. We have the component complexities as follows:

$$\begin{aligned} T_1 &= O\left(\frac{k \log(k) \sqrt{n}}{r \varepsilon^2}\right) && \text{(From Stage 1 in Section 3.1)} \\ T_2 &= \begin{cases} O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } n \gtrsim k \log(k) \\ O\left(\frac{k \log(k)}{\varepsilon^2}\right) & \text{if } n \lesssim k \log(k) \end{cases} && \text{(From Theorem 3)} \end{aligned}$$

The bounds follow by summing T_1 and T_2 for each of the two corresponding regimes for n . ■

C.5. Finding One Distribution in Many

Our procedure for finding one unknown distribution is given in Algorithm 2. Since we will apply this procedure on a suitably chosen subset of our distributions rather than the full set of distributions, we use generic notation m for the number of distributions here.

In this section, we will prove the following theorem, and the proof closely follows the proof of Proposition 2.11 in (Diakonikolas and Kane, 2016).

Theorem 15 *Find-One-Unknown (Alg. 2) takes in $m \geq 2$ distributions $\{D_i\}_{i \in [m]}$ over $[n]$ (for $n \geq 27$), and parameters $\varepsilon, \delta \in (0, 1)$, such that for an unknown set $\chi \subseteq [m]$, and unknown distributions P and Q , with $d_{\text{TV}}(P, Q) \geq \varepsilon$, $D_i = P$ for $i \in \chi$, and $D_i = Q$ for $i \in [m] \setminus \chi$. With probability at least $2/3$, if $\chi = [m]$, the algorithm returns 'All Identical', and if $\chi \neq [m]$, it outputs an index j such that $D_j \neq D_1$. The sample complexity is*

$$O\left(\max\left(\left(\frac{nm}{\varepsilon^2}\right)^{2/3}, \frac{m\sqrt{n}}{\varepsilon^2}\right) \log(m)\right)$$

We use the following result from Diakonikolas and Kane (2016) where it appears as Lemma 2.3. The original lemma stated a success probability of $\frac{2}{3}$, but we state it in terms of $1 - \delta$ where $\delta \in (0, 1)$. This is possible via a majority vote, which can boost the success probability from $\frac{2}{3}$ to $1 - \delta$ at a multiplicative cost of $O(\log(\frac{1}{\delta}))$ in the sample complexity.

Algorithm 2: Find-One-Unknown

Input: $\varepsilon \in (0, 1)$, sample access to distributions $\{D_i\}_{i=1}^m$.

Output: A pair of indices $\mathcal{E} \subseteq [m]$ representing one exemplar from each cluster, or a report that all distributions are identical.

```

1  $s_1 \leftarrow \min(n, (\frac{nm}{\varepsilon^2})^{2/3})$ 
2  $\delta' \leftarrow \frac{1}{30m}$ 
3  $b \leftarrow \sqrt{\frac{30}{s_1}}$ 
4  $s_2 \leftarrow Cns_1^{-1/2} \frac{\log(1/\delta')}{\varepsilon^2}$  //  $C$  is a universal constant
5  $s \sim \text{Poi}(s_1)$  //  $\Pr[\text{Poi}(\mu) = j] = \frac{e^{-\mu}\mu^j}{j!}$ 
6 if  $s > 2s_1$  then
7   return  $\perp$ 
8 Define a multiset  $\mathcal{S}$  by taking  $s$  samples from  $D_1$ 
9 for  $i \in [m] \setminus \{1\}$  do
10   Run the algorithm from Lemma 16 for  $D_{1,\mathcal{S}}, D_{i,\mathcal{S}}$  with  $s_2$  samples,  $\varepsilon, \delta'$ , and  $b$ 
11   if the test returns  $d_{\text{TV}}(D_{1,\mathcal{S}}, D_{i,\mathcal{S}}) > \varepsilon$  then
12     return  $\{1, i\}$ 
13 return All Identical

```

Lemma 16 *Let P and Q be two unknown distributions on $[n]$. There exists an algorithm that on inputs $\varepsilon, \delta \in (0, 1)$, and $b \geq \min\{\|P\|_2, \|Q\|_2\}$, draws $O(bn \log(\frac{1}{\delta})/\varepsilon^2)$ samples from each of P and Q and, with probability at least $1 - \delta$, distinguishes between $d_{\text{TV}}(P, Q) = 0$ and $d_{\text{TV}}(P, Q) \geq \varepsilon$.*

Proof [Proof of Theorem 15] The algorithm draws s samples from D_1 to create the multiset \mathcal{S} , where s is a random variable drawn from a Poisson distribution with mean s_1 . If $s > 2s_1$, the algorithm terminates immediately and returns \perp . Let E_1 denote the event that $s > 2s_1$. By the Chernoff bound and $s_1 \geq n^{2/3}$, we have $\Pr[E_1] = \Pr[s > 2s_1] \leq (\frac{\varepsilon}{4})^{s_1} \leq (\frac{\varepsilon}{4})^{n^{2/3}} \leq \frac{1}{30}$ (for $n \geq 27$).

As laid out in (Diakonikolas and Kane, 2016), the multiset \mathcal{S} is used to transform all of the $\{D_i\}_{i \in [m]}$ into new “flattened” distributions $\{D_{i,\mathcal{S}}\}_{i \in [m]}$ satisfying the following properties:

- These distributions $D_{i,\mathcal{S}}$ are defined on an expanded domain of size $n + |\mathcal{S}|$;
- The pairwise TV distances of these distributions are identical to those of $\{D_i\}_{i \in [m]}$ (see Fact 2.5 of (Diakonikolas and Kane, 2016));
- These distributions have low ℓ_2 norm on average; specifically, $\mathbb{E}[\|D_{i,\mathcal{S}}\|_2^2] \leq \frac{1}{s_1}$ (see Lemma 2.6 of (Diakonikolas and Kane, 2016)).

Let E_2 denote the event that $\|D_{1,\mathcal{S}}\|_2 > \sqrt{\frac{30}{s_1}}$. By Markov’s inequality,

$$\Pr[E_2] = \Pr\left[\|D_{1,\mathcal{S}}\|_2 > \sqrt{\frac{30}{s_1}}\right] = \Pr\left[\|D_{1,\mathcal{S}}\|_2^2 > \frac{30}{s_1}\right] \leq \frac{\mathbb{E}[\|D_{1,\mathcal{S}}\|_2^2]}{30/s_1} \leq \frac{1}{30}.$$

We perform our analysis conditioned on \bar{E}_2 (the complement of E_2), i.e., $\|D_{1,S}\|_2 \leq \sqrt{\frac{30}{s_1}}$. The algorithm sets $b = \sqrt{\frac{30}{s_1}}$. Thus, on this event, the prerequisite condition $b \geq \min(\|D_{1,S}\|_2, \|D_{i,S}\|_2)$ for Lemma 16 is satisfied, since $\min(\|D_{1,S}\|_2, \|D_{i,S}\|_2) \leq \|D_{1,S}\|_2 \leq b$. The tester then distinguishes between $d_{\text{TV}}(D_{1,S}, D_{i,S}) = 0$ and $d_{\text{TV}}(D_{1,S}, D_{i,S}) \geq \varepsilon$, which is equivalent to distinguishing between $d_{\text{TV}}(D_1, D_i) = 0$ and $d_{\text{TV}}(D_1, D_i) \geq \varepsilon$. The tester draws $s_2 = Cbn \log(1/\delta')/\varepsilon^2 \asymp s_1^{-1/2} n \log(\frac{1}{\delta'})/\varepsilon^2$ samples, and errs with probability at most δ' . Let E_3 be the event that at least one of the $m - 1$ tests fail. Since $\delta' = \frac{1}{30m}$, a union bound over the $m - 1$ tests limits the total error to be at most $\Pr[E_3] \leq (m - 1)\delta' \leq \frac{m-1}{30m} \leq \frac{1}{30}$. The total probability of error is at most $\Pr[E_1] + \Pr[E_2] + \Pr[E_3] \leq \frac{1}{10}$, by the union bound.

We condition on the success event $\bar{E} = \bar{E}_1 \cap \bar{E}_2 \cap \bar{E}_3$, which happens with probability at least $1 - \frac{1}{10} = \frac{9}{10}$. If $\chi = [m]$, then since the test from Lemma 16 run correctly, the loop completes and the algorithm returns ‘‘All Identical’’. If $\chi \neq [m]$, then there exists at least one $i \in [m] \setminus \{1\}$ such that $d_{\text{TV}}(D_1, D_i) \geq \varepsilon$. Since the test does not fail, it will correctly return that index i .

The total sample complexity, $T = O(s_1 + m \cdot s_2)$, is determined by the sketching cost (s_1) and the total comparison cost ($m \cdot s_2$). The algorithm chooses $s_1 = \min(n, (\frac{nm}{\varepsilon^2})^{2/3})$ to balance these costs. In more detail, the per-comparison cost is $s_2 \asymp ns_1^{-1/2} \cdot \frac{\log(m)}{\varepsilon^2}$, and this creates two regimes:

- When $m < \varepsilon^2 \sqrt{n}$, the algorithm balances the costs by setting $s_1 = (\frac{nm}{\varepsilon^2})^{2/3}$. This results in a total complexity of $O\left(\left(\frac{nm}{\varepsilon^2}\right)^{2/3} \log(m)\right)$.
- When $m \geq \varepsilon^2 \sqrt{n}$, we cap the sketching cost at $s_1 = n$ to prevent the expanded domain size $n + |\mathcal{S}|$ (with $|\mathcal{S}| = O(s_1)$) from being significantly higher than the original size n .¹ The comparison cost dominates, leading to a total complexity of $O\left(\frac{m\sqrt{n}}{\varepsilon^2} \log(m)\right)$.

The overall complexity is the maximum of the resulting terms from these two regimes:

$$O\left(\max\left(\left(\frac{nm}{\varepsilon^2}\right)^{2/3}, \frac{m\sqrt{n}}{\varepsilon^2}\right) \log(m)\right).$$

■

C.6. Sample Complexity of Clustering (Both Unknown)

Theorem 17 *Our algorithm solves the 2-clustering problem with two unknown distributions with success probability at least $\frac{2}{3}$. The total sample complexity is given by:*

- For $n \gtrsim \frac{k \log(k)}{\varepsilon^4}$:

$$O\left(\max\left(\left(\frac{nk}{r\varepsilon^2}\right)^{2/3}, \frac{k\sqrt{n}}{r\varepsilon^2}\right) \log(k) + \left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}\right).$$

- For $k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4}$:

$$O\left(\max\left(\left(\frac{nk}{r\varepsilon^2}\right)^{2/3}, \frac{k\sqrt{n}}{r\varepsilon^2}\right) \log(k) + \frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right).$$

1. In principle we could allow $n + |\mathcal{S}| \gg n$, but it can be verified that doing so is not beneficial here.

- For $n \lesssim k \log(k)$:

$$O\left(\max\left(\left(\frac{nk}{r\varepsilon^2}\right)^{2/3}, \frac{k\sqrt{n}}{r\varepsilon^2}\right)\log(k) + \frac{k \log(k)}{\varepsilon^2}\right).$$

Proof This algorithm proceeds in two stages, with the total complexity given by the sum of their costs, $T = T_1 + T_2$. For the first stage we call Find-One-Unknown (Algorithm 2) with $3k/r$ sub-sampled arms, i.e., we set $m \leftarrow 3k/r$. Thus we get following component complexities:

$$T_1 = O\left(\max\left(\left(\frac{nk}{r\varepsilon^2}\right)^{2/3}, \frac{k\sqrt{n}}{r\varepsilon^2}\right)\log(k/r)\right) \quad (\text{From Theorem 15})$$

$$T_2 = \begin{cases} O\left(\left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{1/3}\right) & \text{if } n \gtrsim \frac{k \log(k)}{\varepsilon^4} \\ O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right) & \text{if } k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4} \\ O\left(\frac{k \log(k)}{\varepsilon^2}\right) & \text{if } n \lesssim k \log(k) \end{cases} \quad (\text{From Theorem 1})$$

The theorem's bounds follow by summing T_1 and T_2 for each of the three corresponding regimes for n . ■

Appendix D. Details and Proofs for d -Clustering

In this appendix, we extend our results to d -clustering distributions where $d > 2$. We generalize the problem setup to allow for a mix of known and unknown cluster centers. Specifically, we consider a setting with d total clusters, comprising d_{kn} unknown centers and d_{un} known centers, such that $d = d_{\text{un}} + d_{\text{kn}}$.

The Distribution d -Clustering Problem

Input: Sample access to k distributions $\{D_i\}_{i=1}^k$ over domain $[n]$, parameter $\varepsilon \in (0, 1]$, and a set of known reference distributions $\mathcal{Q} = \{Q_1, \dots, Q_{d_{\text{kn}}}\}$. The distributions $\{D_i\}_{i=1}^k$ are allowed to be queried adaptively.

Promise: There exists a hidden partition $\{\mathcal{I}_j\}_{j=1}^d$ of $[k]$ that satisfies the following:

- The clusters correspond to d centers, comprised of the d_{kn} known distributions and d_{un} unknown distributions (where $d = d_{\text{un}} + d_{\text{kn}}$).
- $\forall(a, b \in \mathcal{I}_j), D_a = D_b$.
- $\forall(a \in \mathcal{I}_i, b \in \mathcal{I}_j)$ with $i \neq j$, $d_{\text{TV}}(D_a, D_b) \geq \varepsilon$.

Goal: With probability at least $2/3$, output the true partition $\{\mathcal{I}_j\}_{j=1}^d$.

To solve this problem, we propose a two-stage algorithm. In Stage 1, we identify d_{un} exemplars, one for each unknown distribution. In Stage 2, we classify every distribution in the input set against all d exemplars (both known and discovered) to recover the partition.

D.1. Stage 1: Finding All Exemplars

Since d_{kn} centers are already known, our goal is to obtain sample access to the remaining d_{un} unknown centers. We achieve this by sampling a subset of distributions uniformly at random. Let r be the size of the smallest cluster among those with unknown distributions. The problem of finding at least one member from each of the d_{un} unknown clusters is analogous to the Coupon Collector's Problem. We state this formally as follows and defer the proof to Appendix G.1.

Lemma 18 *Let $\mathcal{K} \subseteq [k]$ be a subset of indices chosen uniformly at random without replacement. If $|\mathcal{K}| = \lceil \frac{k}{r} \log(9d_{\text{un}}) \rceil$, then with probability at least $\frac{8}{9}$, the set $\{D_i\}_{i \in \mathcal{K}}$ contains at least one representative from each of the d_{un} unknown clusters.*

Once the subset \mathcal{K} of size $\lceil \frac{k}{r} \log(9d_{\text{un}}) \rceil$ is drawn, we must verify which distributions are new exemplars. We first filter out the distributions that correspond to the known exemplars (if $d_{\text{kn}} \geq 1$), and hence cannot be the new unknown exemplars. To do this, we perform the following procedure: for each $D \in \mathcal{K}$, we test if D is equivalent to any distribution in \mathcal{Q} (known centers) and if so, we remove the said distribution from \mathcal{K} . This requires running $O(|\mathcal{K}| \cdot d_{\text{kn}})$ standard identity tests against each distribution in \mathcal{Q} , drawing $O\left(\frac{|\mathcal{K}|\sqrt{n}}{\varepsilon^2} \log_+(d_{\text{kn}})\right)^2$ samples in total (see Table 2). The log factor arises as we reuse the samples from each distribution d_{kn} times, hence each test must succeed with probability $1 - O(1/d_{\text{kn}})$.

Then, we apply the Find-All-Exemplars algorithm (described in the following subsection) to identify the unique set of unknown exemplars $\mathcal{P} = \{P_1, \dots, P_{d_{\text{un}}}\}$. Since the filtered subset has at most $|\mathcal{K}|$ distributions, running Find-All-Exemplars has the following sample complexity (see Theorem 19 below):

$$O\left(\max\left(\left(\frac{d_{\text{un}}|\mathcal{K}|^2 n^2}{\varepsilon^4}\right)^{1/3}, \frac{|\mathcal{K}|\sqrt{n}}{\varepsilon^2}\right) \log(|\mathcal{K}|)\right).$$

Combining the complexity of the initial filtering operation and the subsequent run of Find-All-Exemplars, and accounting for the fact that $d_{\text{un}} = d_{\text{kn}} = O(1)$ we see that the sample complexity of Stage 1 is:

$$O\left(\max\left(\left(\frac{k^2 n^2}{r^2 \varepsilon^4}\right)^{1/3}, \frac{k\sqrt{n}}{r\varepsilon^2}\right) \log\left(\frac{k}{r}\right)\right).$$

If the minimum cluster size r is unknown, we can employ the adaptive strategy described in Appendix H. Specifically, we proceed in rounds, geometrically increasing the size of the random subset \mathcal{K} and running Algorithm 3 in each round, terminating once all d_{un} distinct unknown exemplars are identified.

D.1.1. THE FIND-ALL-EXEMPLARS ALGORITHM

The procedure for finding one exemplar from each cluster is given in Algorithm 3. Since we apply this procedure on a suitably-chosen subset of our distributions rather than the full set of distributions, we use generic notation m for the number of distributions here.

We will prove the following theorem, which mostly uses the same ideas as Theorem 15.

2. We use $\log_+(x)$ to denote $\max(\log(x), 1)$.

Algorithm 3: Find-All-Exemplars

Input: $\varepsilon \in (0, 1)$, distributions $\{D_i\}_{i=1}^m$, d_{un} (putative number of clusters).

Output: A set of indices $\mathcal{E} \subseteq [m]$ representing one exemplar from each cluster, or a report that fewer than expected clusters found.

```

1  $s_1 \leftarrow \min(n, (\frac{nm}{d_{\text{un}}\varepsilon^2})^{2/3})$ 
2  $\delta' \leftarrow \frac{1}{9m^2}$ 
3  $b \leftarrow \sqrt{\frac{9d_{\text{un}}}{s_1}}$ 
4  $s_2 \leftarrow Cns_1^{-1/2} \frac{\log(1/\delta')}{\varepsilon^2}$  with  $C$  being the hidden  $O(\cdot)$  constant in Lemma 16
5 Draw  $s_2$  samples from every distribution  $D_i$  for  $i \in \mathcal{U}$ 
6  $\mathcal{E} \leftarrow \emptyset$ 
7 while  $\mathcal{U} \neq \emptyset$  do
8   Pick an arbitrary index  $j \in \mathcal{U}$  to be the exemplar
9    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j\}$ 
10   $\mathcal{E} \leftarrow \mathcal{E} \cup \{j\}$ 
11  if  $|\mathcal{E}| = d_{\text{un}}$  then
12    return  $\mathcal{E}$ 
13  Draw  $s \sim \text{Poi}(s_1)$  samples from  $D_j$  to form sketch  $\mathcal{S}$ 
14  if  $s > e^2 \log(9d_{\text{un}})s_1$  then
15    return  $\perp$ 
16   $\mathcal{C} \leftarrow \emptyset$ 
17  for  $i \in \mathcal{U}$  do
18    Run the test from Lemma 16 comparing  $D_{i,\mathcal{S}}$  and  $D_{j,\mathcal{S}}$  with the pre-drawn  $s_2$  samples,
    and with parameters  $\varepsilon, \delta'$  and  $b$ 
19    if test returns  $d_{\text{TV}}(D_{i,\mathcal{S}}, D_{j,\mathcal{S}}) = 0$  then
20       $\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$ 
21   $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{C}$ 
22 if  $|\mathcal{E}| < d_{\text{un}}$  then
23  return Fewer than expected clusters found

```

Theorem 19 *Find-All-Exemplars (Alg. 3) takes in parameters $\varepsilon \in (0, 1)$ and m distributions containing at most d_{un} unique clusters. With probability at least $2/3$:*

1. *If there are exactly d_{un} clusters, it returns a set \mathcal{E} containing exactly one index from each cluster.*
2. *If there are fewer than d_{un} clusters, it returns \perp .*

In the case of constant d_{un} , the sample complexity is:

$$O\left(\max\left(\left(\frac{m^2 n^2}{\varepsilon^4}\right)^{1/3}, \frac{m\sqrt{n}}{\varepsilon^2}\right) \log(m)\right).$$

Proof In each iteration of the while loop, the algorithm picks an arbitrary distribution D_j among those still under consideration, and designates it as an exemplar, adding it to the exemplar set \mathcal{E} . Then, the algorithm draws s samples from D_j to create the multiset \mathcal{S} , where s is a random variable drawn from a Poisson distribution with mean s_1 . If $s > cs_1$, where $c = e^2 \log(9d_{\text{un}})$, the algorithm terminates immediately and returns \perp . The algorithm iterates through the outer while loop at most d_{un} times (once per unique cluster). Let $s^{(\ell)}$ denote the number of samples drawn for the ℓ -th exemplar, where $s^{(\ell)} \sim \text{Poi}(s_1)$.

Let E_1 denote the event that for any of the d_{un} iterations, $s^{(\ell)} > cs_1$. By the Chernoff bound, for any $s_1 \geq 1$, we have

$$\Pr[s^{(\ell)} > cs_1] \leq \frac{e^{-s_1}(es_1)^{cs_1}}{(cs_1)^{cs_1}} = \left(\frac{e^{c-1}}{c^c}\right)^{s_1} \leq \frac{e^c}{c^c} \leq \frac{1}{e^c} \leq \frac{1}{9d_{\text{un}}},$$

and by a union bound over d_{un} iterations, $\Pr[E_1] \leq \frac{1}{9}$.

As discussed in Theorem 15, the multi-set \mathcal{S} drawn from D_j is used to transform all the distributions into “flattened” distributions and the equivalence tests are conducted on these flattened distributions. The ℓ_2 norm of these flattened distributions determines the sample complexity of the test, and we know that it is small in expectation, i.e., $\mathbb{E}[\|D_{j,\mathcal{S}}\|_2^2] \leq \frac{1}{s_1}$.

Let $E_{2,\ell}$ denote the event that for exemplar ℓ (of the d_{un} exemplars), it is the case that $\|D_{j,\mathcal{S}^{(\ell)}}\|_2 > \sqrt{\frac{9d_{\text{un}}}{s_1}}$. By Markov’s inequality:

$$\Pr[E_{2,\ell}] = \Pr\left[\|D_{j,\mathcal{S}^{(\ell)}}\|_2 > \sqrt{\frac{9d_{\text{un}}}{s_1}}\right] = \Pr\left[\|D_{j,\mathcal{S}^{(\ell)}}\|_2^2 > \frac{9d_{\text{un}}}{s_1}\right] \leq \frac{\mathbb{E}[\|D_{j,\mathcal{S}^{(\ell)}}\|_2^2]}{9d_{\text{un}}/s_1} \leq \frac{1}{9d_{\text{un}}}. \quad (7)$$

Let E_2 be the event denoting the case that for at least one exemplar, the $\|D_{j,\mathcal{S}^{(\ell)}}\|_2 > \sqrt{\frac{9d_{\text{un}}}{s_1}}$ condition holds. Hence, since $E_2 = \bigcup_{\ell} E_{2,\ell}$, a union bound over d_{un} exemplars gives $\Pr[E_2] \leq \frac{1}{9}$.

Testing Event (E_3). We henceforth condition on $\overline{E_1} \cap \overline{E_2}$. On this event, recalling that $b = \sqrt{\frac{9d_{\text{un}}}{s_1}}$, the prerequisite condition $b \geq \min(\|D_{i,\mathcal{S}^{(\ell)}}\|_2, \|D_{j,\mathcal{S}^{(\ell)}}\|_2)$ for Lemma 16 is satisfied, since $\min(\|D_{i,\mathcal{S}^{(\ell)}}\|_2, \|D_{j,\mathcal{S}^{(\ell)}}\|_2) \leq \|D_{j,\mathcal{S}^{(\ell)}}\|_2 \leq b$. The testing subroutine on Line 18 distinguishes between $d_{\text{TV}}(D_{i,\mathcal{S}}, D_{j,\mathcal{S}}) = 0$ and $d_{\text{TV}}(D_{i,\mathcal{S}}, D_{j,\mathcal{S}}) \geq \varepsilon$, which is equivalent to distinguishing between $d_{\text{TV}}(D_i, D_j) = 0$ and $d_{\text{TV}}(D_i, D_j) \geq \varepsilon$. The tester draws $s_2 = Cbn \log(1/\delta)/\varepsilon^2 \asymp s_1^{-1/2} n \log(\frac{1}{\delta})/\varepsilon^2$ samples and errs with probability at most δ' . Let E_3 be the event that *any* of the pairwise tests return an incorrect result.

In the worst case, the algorithm compares every candidate against every identified exemplar. The total number of pairwise tests is thus bounded by $md_{\text{un}} \leq m^2$. The algorithm sets the error per-test to be $\delta' \leftarrow \frac{1}{9m^2}$, so by a union bound over at most m^2 tests, $\Pr[E_3] \leq m^2\delta' \leq \frac{1}{9}$.

The total probability of error is $\Pr[E_1] + \Pr[E_2] + \Pr[E_3] \leq \frac{1}{3}$. Conditioned on the success event $\overline{E} = \overline{E_1} \cap \overline{E_2} \cap \overline{E_3}$, the algorithm operates as follows: In each iteration, it picks a distribution j from \mathcal{U} , which must belong to a previously undiscovered cluster. It correctly identifies all $i \in \mathcal{U}$ equivalent to j and removes them.

- If there are exactly d_{un} clusters, the loop runs d_{un} times, collecting one index per cluster in \mathcal{E} .
- If there are fewer than d_{un} clusters, the loop terminates with $|\mathcal{E}| < d_{\text{un}}$, and the algorithm returns \perp .

Sample Complexity. The total sample complexity is determined by the sketching cost paid d_{un} times at most, and the comparison cost (paid for all m distributions): $O(d_{\text{un}}s_1 + ms_2)$. The algorithm chooses $s_1 \asymp \min(n, (\frac{mn}{d_{\text{un}}\varepsilon^2})^{2/3})$, and $s_2 \asymp ns_1^{-1/2} \cdot \frac{\log(m)}{\varepsilon^2}$ to balance the costs. This creates two regimes:

- When $\frac{m}{d_{\text{un}}} < \varepsilon^2\sqrt{n}$, we use $s_1 = (\frac{mn}{d_{\text{un}}\varepsilon^2})^{2/3}$. The total complexity is $\Theta((\frac{d_{\text{un}}m^2n^2}{\varepsilon^4})^{1/3} \log(m))$.
- $\frac{m}{d_{\text{un}}} \geq \varepsilon^2\sqrt{n}$, we cap the fingerprinting cost at $s_1 = n$. The complexity is dominated by the m comparisons: $ms_2 = \Theta(\frac{m\sqrt{n}}{\varepsilon^2} \log(m))$.

Combining these regimes, we obtain a sample complexity of

$$O\left(\max\left(\left(\frac{d_{\text{un}}m^2n^2}{\varepsilon^4}\right)^{1/3}, \frac{m\sqrt{n}}{\varepsilon^2}\right)\log(m)\right),$$

which simplifies to the stated scaling when d_{un} is constant. ■

D.2. Stage 2: All-Pairs Classification

In this stage, we have the full set of $d = d_{\text{un}} + d_{\text{kn}}$ exemplars. Let $\mathcal{E} = \mathcal{Q} \cup \mathcal{P}$ denote this set. To classify the k input distributions, we adopt an *all-pairs* tournament strategy. For every input distribution D_i ($i \in [k]$), and for every distinct pair of exemplars $(E_a, E_b) \in \mathcal{E} \times \mathcal{E}$, we run a specific pairwise test to decide if D_i is closer to E_a or E_b :

1. Both Known ($E_a, E_b \in \mathcal{Q}$): We run Scheffé’s test as described in Section 1.1.
2. One Known, One Unknown ($E_a \in \mathcal{Q}, E_b \in \mathcal{P}$): We run ESW-Tester as used for $d = 2$.
3. Both Unknown ($E_a, E_b \in \mathcal{P}$): We run MultiLFHT as used for $d = 2$.

Specifically, we test $D_i = E_a$ vs. $D_i = E_b$. If D_i truly belongs to the cluster of E_a , it will win all pairwise comparisons against other exemplars E_b . Since we treat d as a constant, the number of such pairs is $O(1)$. Consequently, the analysis of the sample complexity mirrors the $d = 2$ case exactly, with the total complexity dominated by the hardest pairwise distinction summed over k distributions.

Theorem 20 *Given d_{kn} known distributions and sample access to d_{un} unknown distributions, there exists an algorithm that, with success probability at least $\frac{8}{9}$, classifies k input distributions against the $d = d_{\text{kn}} + d_{\text{un}}$ exemplars. Assuming d is a constant:*

- **If $d_{\text{un}} = 0$:** The sample complexity is $O\left(\frac{k \log k}{\varepsilon^2}\right)$.
- **If $d_{\text{un}} = 1$:** The sample complexity matches the bound for the “One Known, One Unknown” classification stage (Theorem 3).
- **If $d_{\text{un}} \geq 2$:** The sample complexity matches the bound for the “Both Unknown” classification stage (Theorem 1).

Proof The algorithm performs $O(d^2k)$ pairwise tests. Since $d = O(1)$, this scales as $O(k)$. The necessary sample complexity is dictated by the difficulty of the specific pairwise comparisons involved.

- **Case $d_{\text{un}} = 0$:** We only compare against known distributions using Scheffé’s test, requiring $O(\frac{\log k}{\varepsilon^2})$ samples per distribution.
- **Case $d_{\text{un}} = 1$:** The hardest comparison is between the single unknown exemplar and the known exemplars. The optimization of samples (s_1 from the unknown exemplar, s_2 from the inputs) is identical to the derivation in Section C.3, yielding the bounds in Theorem 3.
- **Case $d_{\text{un}} \geq 2$:** The hardest comparison involves distinguishing between two unknown exemplars. This necessitates the use of MultiLFHT as derived in Section C.1. The constraints and resulting regimes are identical to the classification stage of the “Both Unknown” case, yielding the bounds in Theorem 1.

We omit the detailed calculations for general d as they follow the identical structure of the proofs in Section 3 up to constant factors depending on d . ■

Appendix E. Derivation of Lower Bounds

E.1. Proof of Theorem 4 (Unequal Sample Testing Based Lower Bounds)

Theorem 4 *Let $k \geq 60$ and $1 \leq r \leq \frac{k}{60}$. Consider a set of k distributions $\{D_i\}_{i \in [k]}$, where for an unknown subset $\mathcal{I} \subseteq [k]$ of size r , we have $D_i = P$ for all $i \in \mathcal{I}$ and $D_i = Q$ for all $i \in [k] \setminus \mathcal{I}$, with $d_{\text{TV}}(P, Q) \geq \varepsilon$. Any algorithm (possibly adaptive and/or knowing r) that returns an index in \mathcal{I} with probability at least 0.9 requires a number of samples lower bounded as follows:*

- (a) **(One Known)** *If Q is known to be uniform, then the required number of samples is $\Omega\left(\frac{k\sqrt{n}}{r\varepsilon^2}\right)$.*
- (b) **(Both Unknown)** *If both P and Q are unknown, the required number of samples is $\Omega\left(\max\left\{\frac{k\sqrt{n}}{r\varepsilon^2}, \left(\frac{nk}{r\varepsilon^2}\right)^{2/3}\right\}\right)$.*

Proof The proof proceeds via a reduction. We first establish in the following lemma that any algorithm A capable of identifying an index from the unknown set \mathcal{I} can also be used to solve the UST problem defined in Section 4.1. Recall that $\text{UST}(P, Q, s_1, s_2)$ is the task of distinguishing, with probability at least $2/3$, between the hypotheses $H_0 : P = Q$ and $H_1 : \text{TV}(P, Q) \geq \varepsilon$, given s_1 samples from P and s_2 samples from Q . The lower bounds for $\text{UST}(P, Q, s_1, s_2)$ (Lemma 5) are proved by constructing two explicit distributions: Λ_{alt} , supported on pairs (P, Q) that are ε -far, and Λ_0 , supported on pairs (P, Q) with $P = Q$, such that no algorithm can distinguish (with at least 0.9 probability) whether $(P, Q) \sim \Lambda_{\text{alt}}$ or $(P, Q) \sim \Lambda_0$ unless the number of samples s_1 and s_2 satisfies as mentioned in Lemma 5.

For example, the first part of Lemma 5, i.e, when $Q = U$, is proved in Paninski (2008) by taking $\Lambda_0 = \{(U, U)\}$, and letting $\Lambda_{\text{alt}} = (U, Q)$ where Q is the distribution chosen uniformly at random from the family of non-uniform distributions $\{Q_S : S \subseteq [n], |S| = n/2\}$, where each Q_S is defined as

$$Q_S(i) = \begin{cases} \frac{1 + 2\varepsilon}{n}, & i \in S, \\ \frac{1 - 2\varepsilon}{n}, & i \notin S, \end{cases} \quad i \in [n].$$

Similarly, the proof of the second part of Lemma 5 relies on constructing two explicit distributions, Λ_{alt} and Λ_0 . For the purposes of proving Theorem 4, we only require the *existence* of these distributions as a black box; their explicit descriptions are not needed. Therefore, we do not define the concrete constructions for the second case here, it can be found in [Bhattacharya and Valiant \(2015\)](#). The proof of Theorem 4 relies on the following lemma.

Lemma 21 *Suppose that there exists a randomized algorithm A as stated in Theorem 4. Let A draw a total of T samples. Then there exists an algorithm A' that solves the UST problem (distinguishing $(P_1, P_2) \sim \Lambda_0$ from $(P_1, P_2) \sim \Lambda_{\text{alt}}$) with probability $\geq 2/3$, using at most $s_1 \leq T$ samples from P_1 and $s_2 \leq 60rT/k$ samples from P_2 .*

Before proving the above lemma, we show how this lemma implies Theorem 4. We apply the known lower bounds for the UST(P_1, P_2, s_1, s_2) problem to the sample complexities $s_1 \leq T$ and $s_2 \leq 60rT/k$ of algorithm A' .

Part (a): One Known. The relevant lower bound from Lemma 5 for this testing problem is $s_2 \gtrsim \frac{\sqrt{n}}{\varepsilon^2}$. Substituting into s_2 : $s_2 \gtrsim \frac{\sqrt{n}}{\varepsilon^2} \implies \frac{60rT}{k} \gtrsim \frac{\sqrt{n}}{\varepsilon^2} \implies T \gtrsim \frac{k\sqrt{n}}{r\varepsilon^2}$. The required number of samples is $T \in \Omega\left(\frac{k\sqrt{n}}{r\varepsilon^2}\right)$.

Part (b): Both Unknown. The bounds from Lemma 5 are split into two regimes.

- **Regime 1:** $n \lesssim 1/\varepsilon^4$. The bounds are identical to Part (a), yielding a lower bound of $T \gtrsim \frac{k\sqrt{n}}{r\varepsilon^2}$.
- **Regime 2:** $n \gtrsim 1/\varepsilon^4$. The relevant bounds are $s_1 s_2^2 \gtrsim \left(\frac{n}{\varepsilon^2}\right)^2$. Substituting s_1 and s_2 :

$$- s_1 s_2^2 \gtrsim \left(\frac{n}{\varepsilon^2}\right)^2 \implies \left(\frac{60rT}{k}\right)^2 \cdot T \gtrsim \frac{n^2}{\varepsilon^4} \implies T^3 \gtrsim \frac{n^2 k^2}{r^2 \varepsilon^4} \implies T \gtrsim \left(\frac{nk}{r\varepsilon^2}\right)^{2/3}.$$

In the $n \gtrsim 1/\varepsilon^4$ regime, the $\left(\frac{nk}{r\varepsilon^2}\right)^{2/3}$ bound dominates the $\left(\frac{k\sqrt{n}}{r\varepsilon^2}\right)^{1/2}$ bound. Therefore, $T \in \Omega\left(\max\left\{\frac{k\sqrt{n}}{r\varepsilon^2}, \left(\frac{nk}{r\varepsilon^2}\right)^{2/3}\right\}\right)$. ■

E.1.1. PROOF OF LEMMA 21

Proof To describe the algorithm A' , we will first construct a set of indices J with some nice properties. This set J will depend on the algorithm A and the distribution Λ_0 .

We construct a random input instance Δ_0 as follows: we sample $(P_1, P_2) \sim \Lambda_0$. Then we sample a uniformly random subset $\mathcal{I}_0 \sim [k]$ such that $|\mathcal{I}_0| = r$. Then we set $\{D_i\}_{i \in \mathcal{I}_0} = P_2$, and $\{D_i\}_{i \in [k] \setminus \mathcal{I}_0} = P_1$.

Let N_a^0 be the random variable for the number of times distribution a is sampled from by A under Δ_0 , and let p_a^0 be the probability that A outputs index a under Δ_0 . The expectations and probabilities are over the drawing of (P_1, P_2) , A 's internal randomness and the samples drawn. We have $\sum_{a=1}^k \mathbb{E}[N_a^0] = T$ and $\sum_{a=1}^k p_a^0 = 1$.

Define three sets of indices:

- $J_1 = \{a \mid \mathbb{E}[N_a^0] \leq 3T/k\}$
- $J_2 = \{a \mid p_a^0 \leq 3/k\}$
- $J = J_1 \cap J_2$

Since $\sum_{a=1}^k \mathbb{E}[N_a^0] = T$, at most $k/3$ indices can satisfy $\mathbb{E}[N_a^0] > 3T/k$. Thus, $|J_1| \geq 2k/3$. Similarly, since $\sum_{a=1}^k p_a^0 = 1$, at most $k/3$ indices can satisfy $p_a^0 > 3/k$, and hence $|J_2| \geq 2k/3$. Then $|J| = |J_1| + |J_2| - |J_1 \cap J_2| \geq k/3$. Let $M_0 = 60rT/k$. For any set $I' \subseteq J$ of size $|I'| \leq r$, we have

$$\Pr \left[\sum_{j \in I'} N_j^0 > M_0 \right] \leq \frac{\mathbb{E} \left[\sum_{j \in I'} N_j^0 \right]}{M_0} = \frac{\sum_{j \in I'} \mathbb{E} [N_j^0]}{M_0} \leq \frac{|I'| \cdot 3T/k}{M_0} \leq \frac{3rT/k}{60rT/k} = 0.05 \quad (8)$$

Now, we construct an algorithm A' (using A) to solve the testing problem (distinguishing $(P_1, P_2) \sim \Lambda_0$ from $(P_1, P_2) \in \Lambda_{\text{alt}}$). A' works as follows:

1. A' takes as input a pair of distributions (P_1, P_2) (which is either drawn from Λ_0 or Λ_{alt}).
2. A' picks r indices arbitrarily from the set J , call it $I' \subseteq J$. It sets up an instance for A : for $j \in I'$, $D_j = P_2$, and for $j \in [k] \setminus I'$, $D_j = P_1$.
3. A' simulates A on this instance. Let N_j be the number of times A draws a sample from D_j , and let Y_A be the index output by A .
4. If, at any point in the simulation, $\sum_{j \in I'} N_j > M_0$, A' aborts and outputs Λ_{alt} .
5. Otherwise ($\sum_{j \in I'} N_j \leq M_0$) A' outputs:
 - Λ_{alt} , if $(Y_A \in I')$.
 - Λ_0 , if $(Y_A \notin I')$.

We now show that A' correctly classifies $(P_1, P_2) \sim \Lambda_0$ or Λ_{alt} with at least 0.9 probability.

Case 1: $(P_1, P_2) \sim \Lambda_0$. This corresponds to the Δ_0 scenario for A . A' is in error if it outputs Λ_{alt} . This occurs if $(\sum_{j \in I'} N_j^0 > M_0)$ or $(\sum_{j \in I'} N_j^0 \leq M_0$ and $Y_A \in I')$. Since these two events are disjoint,

$$\begin{aligned} \Pr[\text{error} | (P_1, P_2) \sim \Lambda_0] &= \Pr \left[\sum_{j \in I'} N_j^0 > M_0 \right] + \Pr \left[(Y_A \in I') \wedge \left(\sum_{j \in I'} N_j^0 \leq M_0 \right) \right] \\ &\leq \underbrace{0.05}_{\text{from eq. (8)}} + \Pr [Y_A \in I'] \\ &\leq 0.05 + \sum_{j \in I'} p_j^0 \leq 0.05 + \frac{3r}{k} \leq 0.05 + \underbrace{0.05}_{r \leq k/60} = 0.1. \end{aligned}$$

Case 2: $(P_1, P_2) \sim \Lambda_{\text{alt}}$. Recall I' defined in step 2 above, and that we set $D_j = P_2$ for $j \in I'$, and $D_j = P_1$ for $j \in [k] \setminus I'$. This is an instance A is designed for. A' errs if it outputs $(P_1, P_2) \in \Lambda_0$. This happens if $(\sum_{j \in I'} N_j \leq M_0$ and $Y_A \notin I')$, so, $\Pr[\text{error} | (P_1, P_2) \sim \Lambda_{\text{alt}}] \leq \Pr[Y_A \notin I']$. By property (3) of A , $\Pr[Y_A \in I'] \geq 0.9$, and thus $\Pr[\text{error} | (P_1, P_2) \sim \Lambda_{\text{alt}}] \leq \Pr[Y_A \notin I'] \leq 0.1$.

Therefore, if $r \leq k/60$, algorithm A' solves the testing problem (distinguishing Λ_0 from Λ_{alt}) with success probability at least 0.9. It uses $s_1 = \sum_{j \in [k] \setminus I'} N_j \leq T$ samples from P_1 and $s_2 = \sum_{j \in I'} N_j \leq M_0 = 60rT/k$ samples from P_2 . \blacksquare

E.2. Proof of Theorem 6 (LFHT Based Lower Bounds)

Theorem 6 *Let $n \geq 1$, $k \geq 100$, and $\varepsilon \in (0, 1]$. Consider a set of k distributions $\{D_i\}_{i \in [k]}$ where for an unknown subset $\mathcal{I} \subseteq [k]$ of size r , we have $D_i = P$ for all $i \in \mathcal{I}$ and $D_i = Q$ for all $i \in [k] \setminus \mathcal{I}$ with $d_{\text{TV}}(P, Q) \geq \varepsilon$. Any algorithm (possibly adaptive and/or knowing r) that returns the partition $\{\mathcal{I}, [k] \setminus \mathcal{I}\}$ with probability at least 0.9 must have a total sample complexity of:*

- (a) $\Omega\left(\frac{k+\sqrt{nk}}{\varepsilon^2}\right)$, when one of the underlying distributions is uniform.
- (b) $\Omega\left(\left(\frac{kn^2}{\varepsilon^4}\right)^{1/3} + \frac{k+\sqrt{nk}}{\varepsilon^2}\right)$ when both underlying distributions are unknown.

The lower bound for LFHT(P, Q, s_1, s_2) (Lemma 7) is proved by constructing an explicit distribution Λ_{alt} supported on pairs (P, Q) that are ε -far. It is shown that, for $(P, Q) \sim \Lambda_{\text{alt}}$, no algorithm can distinguish, with probability at least 0.9, whether a given unknown distribution $D \in \{P, Q\}$, is P or Q unless the number of samples s_1 and s_2 satisfies as mentioned in Lemma 7. Again, as in Section E.1, we will make use of Λ_{alt} in a ‘‘black box’’ manner, and we therefore do not describe their constructions here. Their detailed definitions can be found in Gerber et al. (2023).

Lemma 22 *Suppose there exists a randomized algorithm A satisfying the conditions in Theorem 6. Let A draw a total of T samples. Then there exists an algorithm A' that solves the LFHT(P, Q, s_1, s_2) problem when $(P, Q) \sim \Lambda_{\text{alt}}$, using $s_1 \leq T$ samples from P and Q each, and $s_2 \leq \frac{80T}{k}$ samples from D , i.e., A' solves LFHT($P, Q, T, \frac{80T}{k}$) when $(P, Q) \sim \Lambda_{\text{alt}}$.*

Before proving the above lemma, we show that it implies Theorem 6.

Part (a): One Known We apply Lemma 22 setting $Q = U$. The premise of Theorem 6 (Part 1) assumes an algorithm A satisfying the conditions of Lemma 22 for these distributions. The lemma guarantees an algorithm A' solving LFHT(P, U, s_1, s_2) with $s_1 \leq T$ samples from P and $s_2 \leq \frac{80T}{k}$ samples from D . From Lemma 7, such an algorithm must satisfy $s_1 \gtrsim \frac{\sqrt{n}}{\varepsilon^2}$, $s_1 s_2 \gtrsim \frac{n}{\varepsilon^4}$, and $s_2 \gtrsim \frac{1}{\varepsilon^2}$. Substituting $s_1 \leq T$ and $s_2 \leq \frac{80T}{k}$:

- $T \gtrsim s_1 \gtrsim \frac{\sqrt{n}}{\varepsilon^2}$
- $T\left(\frac{80T}{k}\right) \gtrsim s_1 s_2 \gtrsim \frac{n}{\varepsilon^4} \implies T \gtrsim \frac{\sqrt{nk}}{\varepsilon^2}$
- $\frac{80T}{k} \gtrsim s_2 \gtrsim \frac{1}{\varepsilon^2} \implies T \gtrsim \frac{k}{\varepsilon^2}$

Combining these, we get $T \gtrsim \frac{k+\sqrt{nk}}{\varepsilon^2}$.

Part (b): Both Unknown We apply Lemma 22 with unknown distributions P and Q . The premise of Theorem 6 (Part 2) assumes an algorithm A that identifies the $k - r$ P distributions from the r Q distributions, which satisfies the conditions of Lemma 22. The lemma guarantees an algorithm A' solving LFHT(P, Q, s_1, s_2) with $s_1 \leq T$ samples from Q and $s_2 \leq \frac{80T}{k}$ samples from D . From Lemma 7, such an algorithm must satisfy:

$$\begin{cases} s_1^2 s_2 \gtrsim \left(\frac{n}{\varepsilon^2}\right)^2 & s_1 s_2 \gtrsim \frac{n}{\varepsilon^4} & s_2 \gtrsim \frac{1}{\varepsilon^2} & \text{if } n \gtrsim \frac{1}{\varepsilon^4} \\ s_1 \gtrsim \frac{\sqrt{n}}{\varepsilon^2} & s_1 s_2 \gtrsim \frac{n}{\varepsilon^4} & s_2 \gtrsim \frac{1}{\varepsilon^2} & \text{if } n \lesssim \frac{1}{\varepsilon^4} \end{cases}$$

Substituting $s_1 \leq T$ and $s_2 \leq \frac{80T}{k}$:

Case 1: $n \lesssim 1/\varepsilon^4$. The analysis is identical to Part 1, yielding $T \gtrsim \frac{k+\sqrt{nk}}{\varepsilon^2}$.

Case 2: $n \gtrsim 1/\varepsilon^4$.

- $\frac{80T}{k} \gtrsim s_2 \gtrsim \frac{1}{\varepsilon^2} \implies T \gtrsim \frac{k}{\varepsilon^2}$
- $T(\frac{80T}{k}) \gtrsim s_1 s_2 \gtrsim \frac{n}{\varepsilon^4} \implies T \gtrsim \frac{\sqrt{nk}}{\varepsilon^2}$
- $T^2(\frac{80T}{k}) \gtrsim s_1^2 s_2 \gtrsim \frac{n^2}{\varepsilon^4} \implies T \gtrsim \left(\frac{n^2 k}{\varepsilon^4}\right)^{1/3}$

In the $n \lesssim \frac{1}{\varepsilon^4}$ regime we have $\frac{\sqrt{nk}}{\varepsilon^2} \gtrsim \left(\frac{n^2 k}{\varepsilon^4}\right)^{1/3}$, so we merge the two cases to arrive at the lower bound: $\Omega\left(\left(\frac{n^2 k}{\varepsilon^4}\right)^{1/3} + \frac{k+\sqrt{nk}}{\varepsilon^2}\right)$.

E.2.1. PROOF OF LEMMA 22

Proof Let A be the algorithm as stated in Theorem 6. If the k distributions given as input to A cannot be partitioned into $\{\mathcal{I}, [k] \setminus \mathcal{I}\}$ with $|\mathcal{I}| = r$, then the behaviour of A could in principle be arbitrary; for example, it may attempt to draw more than T samples and/or it may not return a partition of size $(r, k - r)$. However, A can be trivially modified to always terminate after T samples and return a partition of size $(r, k - r)$. Let \mathcal{I}_A denote the subset returned by A of size $k - r$.

Note that the problem is symmetric with respect to the cluster sizes r and $k - r$. Therefore, without loss of generality, we assume in this proof that $r \geq k/2$.

Lemma 23 *Consider the following random input instance for A : let $(P, Q) \sim \Lambda_{\text{alt}}$. Set $D_i = P$ for all $i \in \{r + 2, \dots, k\}$ and $D_i = Q$ for all $i \in \{1, \dots, r + 1\}$. Let the number of samples from D_j be N_j for all $j \in [k]$. Then there exists an index $j \in \{1, \dots, r + 1\}$ (determinable given A and Λ_{alt}) such that:*

- $\Pr[\mathcal{I}_A = \{r + 2, \dots, k\} \cup \{j\}] \leq \frac{4}{k}$
- $\mathbb{E}[N_j] \leq \frac{4T}{k}$

Proof Let $S_i = \{i\} \cup \{r + 2, \dots, k\}$. Since $\sum_{i \in [r+1]} \Pr[\mathcal{I}_A = S_i] \leq 1$, by an averaging argument, there are more than $(r + 1)/2$ indices i for which $\Pr[\mathcal{I}_A = S_i] \leq 2/(r + 1)$. Furthermore, $\sum_{i \in [r+1]} \mathbb{E}[N_i] \leq T$, so there are more than $(r + 1)/2$ indices i for which $\mathbb{E}[N_i] \leq 2T/(r + 1)$. Thus, there exists at least one $j \in [r + 1]$ in both sets. For this j , since $r \geq k/2$ (implying $r + 1 > k/2$), we have: $\Pr[\mathcal{I}_A = S_j] \leq \frac{2}{r+1} < \frac{4}{k}$ and $\mathbb{E}[N_j] \leq \frac{2T}{r+1} < \frac{4T}{k}$. \blacksquare

We now present algorithm A' to solve LFHT($P, Q, T, \frac{80T}{k}$) when $(P, Q) \sim \Lambda_{\text{alt}}$. A' has sample access to P, Q , and an unknown $D \in \{P, Q\}$.

1. A' selects the index j from Lemma 23.
2. A' sets up an instance for A :
 - D_j is set to D .
 - Distributions $\{1, \dots, r + 1\} \setminus \{j\}$ are set to Q .
 - Distributions $\{r + 2, \dots, k\}$ are set to P .

3. A' calls A . When A requests a sample from D_i , A' queries the corresponding oracle D , P , or Q . That is, if A requests a sample from D_i then A' queries P when $i \in \{r+2, \dots, k\}$, Q when $i \in \{1, \dots, r+1\} \setminus \{j\}$, and D when $i = j$.
4. If N_j (number of samples from D) exceeds $\frac{80T}{k}$, A' aborts.
5. Let $S_j = \{j\} \cup \{r+2, \dots, k\}$. A' returns:
 - Q if the run completes ($N_j \leq \frac{80T}{k}$) AND $\mathcal{I}_A \neq S_j$.
 - P if the run aborts ($N_j > \frac{80T}{k}$) OR $\mathcal{I}_A = S_j$.

We proceed to analyze the sample complexity and correctness of A' . Clearly A' draws at most T samples from both P and Q and at most $\frac{80T}{k}$ samples from D . We now show that A' correctly classifies $D \in \{P, Q\}$ with required probability.

Case 1: $D = Q$. The instance given to A has $r+1$ copies of Q and $k-r-1$ copies of P . This is the setup of Lemma 23. By the lemma, $\Pr[\mathcal{I}_A = S_j] \leq 4/k$. By Markov's inequality, $\Pr[N_j > \frac{80T}{k}] \leq \mathbb{E}[N_j]/(\frac{80T}{k}) \leq (4T/k)/(\frac{80T}{k}) = 1/20$. Assuming $k \geq 80$, we have $4/k \leq 1/20$. By the union bound, the probability that A' fails (i.e., aborts or $\mathcal{I}_A = S_j$) is $\leq 1/20 + 1/20 = 0.1$. Thus, A' correctly returns Q with probability ≥ 0.9 .

Case 2: $D = P$. The instance given to A has r copies of Q and $k-r$ copies of P . This is an instance A was designed for. The true set of base indices is $S_j = \{j\} \cup \{r+2, \dots, k\}$. A guarantees that $\Pr[\mathcal{I}_A = S_j] \geq 0.9$. In this event, A' correctly returns P . Thus, A' is correct with probability ≥ 0.9 .

Hence, we find that A' solves LFHT(P, Q, s_1, s_2) with probability ≥ 0.9 . The total number of samples A' draws from the Q oracle is $s_1 = \sum_{i \in [r+1] \setminus \{j\}} N_i \leq T$. The total samples from D is $s_2 = N_j \leq \frac{80T}{k}$ (as it aborts otherwise). ■

Appendix F. Tightness of 2-Clustering Bounds

Here we provide a case-by-case comparison of our upper and lower bounds, as summarized in Table 1. In both the ‘‘One-Known-One-Unknown’’ and ‘‘Both-Unknown’’ settings: the bounds are tight up to a $\log k$ factor. We now discuss these gaps in detail.

Across all scenarios, the r -dependent terms in our sample complexity are tight up to $\log k$ factors (this can be seen directly in Table 1). Hence, whenever the r -dependent term dominates, the overall upper bound of the algorithm matches the lower bound (up to a $\log k$ factor). It therefore remains to compare the r -independent terms from Stage 2.

One Known, One Unknown

- ($n \gtrsim k \log(k)$): Upper bound $O(\frac{\sqrt{nk \log(k)}}{\epsilon^2})$; Lower bound $\Omega(\frac{\sqrt{nk}}{\epsilon^2})$; Gap $\sqrt{\log k}$.
- ($n \lesssim k \log(k)$): Upper bound $O(\frac{k \log(k)}{\epsilon^2})$; Lower bound $\Omega(\frac{k}{\epsilon^2})$; Gap $\log k$.

Both Unknown

- $(n \gtrsim \frac{k \log(k)}{\varepsilon^4})$: Upper bound $O\left(\left(\frac{n^2 k \log(k)}{\varepsilon^4}\right)^{\frac{1}{3}}\right)$, Lower bound $\Omega\left(\left(\frac{n^2 k}{\varepsilon^4}\right)^{\frac{1}{3}}\right)$; Gap $(\log(k))^{\frac{1}{3}}$.
- $(k \log(k) \lesssim n \lesssim \frac{k \log(k)}{\varepsilon^4})$: Upper bound $O\left(\frac{\sqrt{nk \log(k)}}{\varepsilon^2}\right)$; Lower bound $\Omega\left(\frac{\sqrt{nk}}{\varepsilon^2}\right)$; Gap $\sqrt{\log k}$.
- $(n \lesssim k \log(k))$: Upper bound $O\left(\frac{k \log(k)}{\varepsilon^2}\right)$; Lower bound $\Omega\left(\frac{k}{\varepsilon^2}\right)$; Gap: $\log k$.

Appendix G. Miscellaneous Proofs

G.1. Proof of Lemma 18 (Coupon Collector Type Result)

Lemma 18 *Let $\mathcal{K} \subseteq [k]$ be a subset of indices chosen uniformly at random without replacement. If $|\mathcal{K}| = \lceil \frac{k}{r} \log(9d_{\text{un}}) \rceil$, then with probability at least $\frac{8}{9}$, the set $\{D_i\}_{i \in \mathcal{K}}$ contains at least one representative from each of the d_{un} unknown clusters.*

Proof Note that sampling without replacement yields a strictly lower failure probability than sampling with replacement; for simplicity, we analyze the latter to establish the upper bound.

Let $C_1, \dots, C_{d_{\text{un}}}$ denote the sets of indices corresponding to the d_{un} unknown clusters. By the problem promise, the size of each cluster is at least r , i.e., $|C_j| \geq r$ for all $j \in [d_{\text{un}}]$. For a single index drawn uniformly at random from $[k]$, the probability that it belongs to cluster C_j is $p_j = \frac{|C_j|}{k} \geq \frac{r}{k}$. Let E_j be the event that the cluster C_j is absent from the sample set \mathcal{S} . Since the samples are independent, we have: $\Pr[E_j] = (1 - p_j)^{|\mathcal{S}|} \leq \left(1 - \frac{r}{k}\right)^{|\mathcal{S}|} \leq e^{-|\mathcal{S}| \frac{r}{k}}$ using the fact that $1 - x \leq e^{-x}$. By the union bound over all d_{un} clusters we find the probability of at least one cluster C_j not being found:

$$\Pr\left[\bigcup_{j=1}^{d_{\text{un}}} E_j\right] \leq \sum_{j=1}^{d_{\text{un}}} \Pr[E_j] \leq d_{\text{un}} e^{-|\mathcal{S}| \frac{r}{k}}.$$

To ensure this failure probability is at most $\frac{1}{9}$, we set: $d_{\text{un}} e^{-|\mathcal{S}| \frac{r}{k}} \leq \frac{1}{9}$. Thus, choosing $|\mathcal{S}| \geq \frac{k}{r} \log(9d_{\text{un}})$ guarantees that all unknown clusters are discovered with probability at least $\frac{8}{9}$. ■

G.2. Proof of Lemma 12 (Collision-Type Bounds)

In the following, recall that \mathcal{S} represents a random multi-set, and $S = \text{set}(\mathcal{S})$ is the resulting set with duplicates removed.

For a vector $\mathbf{v} = (v_1, v_2, \dots, v_n)$, let \mathbf{v}^\downarrow denote the permutation of \mathbf{v} that follows a non-increasing order, i.e., $\mathbf{v}^\downarrow = (v_1^\downarrow, \dots, v_n^\downarrow)$ is a permutation of \mathbf{v} such that $v_1^\downarrow \geq v_2^\downarrow \dots \geq v_n^\downarrow$. For vectors \mathbf{v} and \mathbf{w} such that $\sum_{i \in [n]} \mathbf{v}(i) = \sum_{i \in [n]} \mathbf{w}(i)$, we say that \mathbf{v} is *majorised* by \mathbf{w} , denoted as $\mathbf{w} \succcurlyeq \mathbf{v}$ if $\sum_{i=1}^r w_i^\downarrow \geq \sum_{i=1}^r v_i^\downarrow$ for all $r \in [k]$.

We will use the following definition of Schur-Concave functions.

Definition 24 (Sec. 3 Def. A.1 in (Marshall et al., 2011)) *A real-valued function f is said to be Schur-concave if $\mathbf{q} \succcurlyeq \mathbf{p} \implies f(\mathbf{q}) \leq f(\mathbf{p})$.*

Lemma 12 For any $t \in \mathbb{R}$ and distribution P ,

$$\Pr_{\mathcal{S} \sim P^{\otimes s_1}} [U(\mathcal{S}) \geq t] \leq \Pr_{\mathcal{S} \sim U^{\otimes s_1}} [U(\mathcal{S}) \geq t].$$

Proof Consider the case that \mathcal{S} results from S containing s_1 i.i.d. samples from P . We wish to bound

$$\Pr_{\mathcal{S} \sim P^{\otimes s_1}} [U(\mathcal{S}) \geq t] = \Pr_{\mathcal{S} \sim P^{\otimes s_1}} [|\mathcal{S}| \geq tn].$$

where we used the fact that $U(\mathcal{S}) = \frac{|\mathcal{S}|}{n}$.

For a distribution D , let $\mathbf{D} = (d_1, \dots, d_n)$ be its probability vector. We define $f(\mathbf{D}) = \Pr_{\mathcal{S} \sim D^{\otimes s_1}} \left[\frac{|\mathcal{S}|}{n} \geq t \right]$. It is known that f is Schur-concave for all n and t (Proposition E.11.b in (Marshall et al., 2011)). Moreover, it is easily verified that the uniform distribution U is such that $\mathbf{P} \succcurlyeq \mathbf{U}$, i.e., U is majorised by every other distribution \mathbf{P} . We can therefore apply Definition 24 as follows:

$$\begin{aligned} & \mathbf{P} \succcurlyeq \mathbf{U} \\ \implies & f(\mathbf{P}) \leq f(\mathbf{U}) \\ \implies & \Pr_{\mathcal{S} \sim P^{\otimes s_1}} \left[\frac{|\mathcal{S}|}{n} \geq t \right] \leq \Pr_{\mathcal{S} \sim U^{\otimes s_1}} \left[\frac{|\mathcal{S}|}{n} \geq t \right] \\ \implies & \Pr_{\mathcal{S} \sim P^{\otimes s_1}} [U(\mathcal{S}) \geq t] \leq \Pr_{\mathcal{S} \sim U^{\otimes s_1}} [U(\mathcal{S}) \geq t]. \end{aligned}$$

■

Appendix H. Adapting to Unknown Cluster Sizes

Our algorithms, as presented, require knowledge of the parameter r representing the size of one of the clusters. This parameter is used only in Stage 1 to determine the number of samples needed to find an exemplar from each cluster; Stage 2 does not depend on r . In this section, we describe a variation that does not require knowledge of r , but requires more rounds of adaptivity, namely, $O(\log \frac{k}{r})$ instead of two.

We focus here on the case that one distribution is known to be uniform and the other is unknown; the argument for ‘both unknown’ is entirely analogous and thus omitted. Recall from Section 3.1 that Stage 1 selects a subset of $\lceil \frac{18k}{r} \rceil$ distributions uniformly at random, and performs uniformity testing with a suitably-chosen target success probability. We showed in the analysis of Stage 1 that the ‘‘good event’’ $\overline{E}_1 \wedge \overline{E}_2$ occurs with probability at least $\frac{8}{9}$, where \overline{E}_1 ensures that at least one non-uniform distribution appears in the subset, and \overline{E}_2 ensures that all of the uniformity tests are correct.

When r is unknown, we modify this procedure to *iteratively add and test distributions until the first success*. We do this in rounds indexed by $i = 1, 2, \dots$, with the i -th round testing an additional 2^i distributions for uniformity, each with an associated error probability of $c \frac{2^{-i}}{i^2}$ for some constant c . As shown in the known- r case, with probability at least $\frac{17}{18}$, one of the first $\lceil \frac{18k}{r} \rceil$ distributions considered will be non-uniform (event \overline{E}_1 above), so our main goal is to obtain a counterpart to event \overline{E}_2 .

Let i^* be the iteration at which the number of distributions becomes at least $\lceil \frac{18k}{r} \rceil$ (and thus at most $2\lceil \frac{18k}{r} \rceil$ since we are doubling in each iteration). By the above choice of target error probability and the union bound, the probability of receiving an erroneous decision in any of these uniformity tests is at most $\sum_{i=1}^{i^*} c \frac{2^{-i}}{i^2} 2^i \leq c \sum_{i=1}^{\infty} \frac{1}{i^2}$, which is at most $\frac{1}{18}$ when c is sufficiently small. This is the desired counterpart to \overline{E}_2 , as combined with the good event from the previous paragraph, it ensures the success of Stage 1.

It remains to study the sample complexity. We first observe that stopping by iteration i^* implies considering at most twice as many distributions (in Stage 1) compared to the known- r case. For known r , the union bound incurred a multiplicative $O(\log \frac{k}{r})$ factor, whereas here the multiplicative factor becomes $O(\log \frac{1}{2^{-i^*}/(i^*)^2}) = O(i^*)$. By the definition of i^* and the fact that the number of distributions added is exponentially increasing (hence dominated by the final term), this simplifies to $O(\log \frac{k}{r})$, thus matching the known- r case to within constant factors. Note also that i^* represents the number of stages of adaptivity by definition.

Appendix I. Adapting to Unknown ε

Our algorithms, as presented, require knowledge of the parameter $\varepsilon^* = d_{\text{TV}}(P, Q)$ that determines the separation of the two clusters. In this section, we relax the assumption that ε^* is provided as input. We employ an iterative guess-and-verify strategy: at iteration $j \in \{0, 1, 2, \dots\}$, we guess that the separation is $\varepsilon_j = 2^{-j}$ and cluster the distributions using this guess. To do so, we use a clustering algorithm, denoted by $\mathcal{C}(\eta, \delta)$, that returns a correct partition with probability at least $1 - \delta$, provided the input distributions are separated by at least η . We then verify the result using a *tolerant test* (defined below) to determine if the separation is indeed at least ε_j . See Algorithm 4 for the details.

The sample complexity of clustering algorithms (for known ε) typically depends on the domain size n , the number of distributions k , and the cluster size r , but we will suppress these dependencies in order to reduce notation. For our adaptive procedure, we will henceforth assume a sample complexity upper bound $N_{\mathcal{C}}(\varepsilon, \delta)$ that scales polynomially with $1/\varepsilon$ and logarithmically with $1/\delta$, as is the case for the algorithms presented in the paper. Thus, we have $N_{\mathcal{C}}(\varepsilon, \delta) \in O(\text{poly}(1/\varepsilon) \cdot \log(1/\delta))$.

Lemma 25 (*Valiant and Valiant, 2011, Theorem 4*) *Given sample access to two distributions P and Q , parameters $\delta \in (0, 1)$, and a threshold $0 \leq \varepsilon \leq 1$, a tolerant test $\mathcal{T}(P, Q, \varepsilon, \delta)$ is an algorithm that with probability at least $1 - \delta$:*

- returns NEAR if $d_{\text{TV}}(P, Q) \leq \frac{\varepsilon}{2}$;
- returns FAR if $d_{\text{TV}}(P, Q) \geq \varepsilon$.

If $\varepsilon/2 < d_{\text{TV}}(P, Q) < \varepsilon$, the test may return either NEAR or FAR. Then, the sample complexity of tolerant testing satisfies an upper bound of the form:

$$N_{\mathcal{T}}(n, \varepsilon, \delta) = O\left(\frac{n}{\log(n)\varepsilon^2} \log \frac{1}{\delta}\right) \tag{9}$$

The following theorem states the sample complexity of our adaptive 2-clustering algorithm in terms of the sample complexities of the underlying non-adaptive clustering and tolerant testing algorithms.

Algorithm 4: 2-Clustering Adaptive to Unknown ϵ

Input: Sample access to $\{D_i\}_{i=1}^k$, failure probability δ .

Oracles: Clustering algorithm \mathcal{C} , tolerant tester \mathcal{T} .

Output: Partition $\{\mathcal{I}_1, \mathcal{I}_2\}$.

for $j = 0, 1, \dots$ **do**

$\epsilon_j \leftarrow 2^{-j}$

$\delta_j \leftarrow \frac{2\delta}{\pi^2(j+1)^2}$

$\{\mathcal{I}_1, \mathcal{I}_2\} \leftarrow \mathcal{C}(\epsilon_j, \delta_j)$

Pick arbitrary indices $a \in \mathcal{I}_1$ and $b \in \mathcal{I}_2$

if $\mathcal{T}(D_a, D_b, \epsilon_j, \delta_j)$ returns FAR **then**

$\{\mathcal{I}_1, \mathcal{I}_2\} \leftarrow \mathcal{C}(\frac{\epsilon_j}{2}, \delta_j)$

return $\{\mathcal{I}_1, \mathcal{I}_2\}$

Theorem 26 Let $N_{\mathcal{C}}(\epsilon, \delta)$ (resp., $N_{\mathcal{T}}(n, \epsilon, \delta)$) be any sample complexity upper bounds for clustering (resp., tolerating testing) whose (ϵ, δ) dependence scales as $O(\text{poly}(1/\epsilon) \cdot \log(1/\delta))$. Then, Algorithm 4 returns the true partition with probability at least $1 - \delta$, with sample complexity at most

$$O\left(\left(N_{\mathcal{C}}(\epsilon^*, \delta) + N_{\mathcal{T}}(n, \epsilon^*, \delta)\right) \left[1 + \frac{\log \log(1/\epsilon^*)}{\log(1/\delta)}\right]\right),$$

where ϵ^* is the true separation.

Remark 27 We note that from (9) the tolerant testing term $N_{\mathcal{T}}$ scales linearly with the domain size n (up to log factors) but is independent of the number of distributions k . In contrast, the clustering complexity $N_{\mathcal{C}}$ grows with both k and n (e.g., \sqrt{nk}), as seen in Table 1. Thus, in regimes where k is large, the cost is dominated by $N_{\mathcal{C}}$, and the overhead from the tolerant testing verification is negligible. As a concrete example, if $k = \Omega(n)$ then the $N_{\mathcal{T}}$ does not change the overall scaling, since $N_{\mathcal{C}}$ is always at least as high as $\frac{k}{\epsilon^2}$. However, for certain regimes of smaller k , the tightness of our upper bound remains open.

Proof The probability that the clustering algorithm or the tolerant tester errs in iteration j is bounded by δ_j . Since in each iteration we make no more than 3 calls to the oracles, the total error is bounded by $3\delta_j$. Using the identity $\sum_{j=0}^{\infty} (j+1)^{-2} = \pi^2/6$, a union bound over all iterations ensures the total error probability is at most $3 \sum_{j=0}^{\infty} \frac{2\delta}{\pi^2(j+1)^2} = 3\delta/3 = \delta$. We henceforth assume that all oracle calls return correct results according to their promises.

Let j^* be the first index such that $\epsilon_{j^*} \leq \epsilon^*$; since $\epsilon_j = 2^{-j}$, we have $j^* = \lceil \log(\frac{1}{\epsilon^*}) \rceil$. For any $j < j^*$, the tolerant tester is invoked with threshold ϵ_j . If $\epsilon^* \leq \frac{\epsilon_j}{2}$, then \mathcal{T} correctly returns NEAR, whereas if $\frac{\epsilon_j}{2} < \epsilon^* < \epsilon_j$ (the “undefined” region), then \mathcal{T} may return FAR or NEAR. Then:

- If \mathcal{T} returns NEAR, the loop continues.
- If \mathcal{T} returns FAR, the algorithm executes Line 10, calling $\mathcal{C}(\frac{\epsilon_j}{2}, \delta_j)$. Since this can only occur in the region where $\epsilon^* > \frac{\epsilon_j}{2}$, the separation condition for \mathcal{C} is satisfied ($d_{\text{TV}}(P, Q) \geq \frac{\epsilon_j}{2}$), and it returns the correct partition.

At iteration $j = j^*$, we have $\varepsilon_{j^*} \leq \varepsilon^* < 2\varepsilon_{j^*}$. The clustering algorithm $\mathcal{C}(\varepsilon_{j^*}, \delta_{j^*})$ succeeds since $\varepsilon^* \geq \varepsilon_{j^*}$. The tolerant tester verifies $d_{\text{TV}}(D_a, D_b) = \varepsilon^* \geq \varepsilon_{j^*}$ and returns FAR. The subsequent ‘‘safety call’’ $\mathcal{C}(\varepsilon_{j^*}/2, \delta_{j^*})$ also succeeds since $\varepsilon^* \geq \varepsilon_{j^*} > \varepsilon_{j^*}/2$.

The total sample complexity is bounded by the sum of sample costs across all iterations. Since the complexities $N_{\mathcal{C}}$ and $N_{\mathcal{T}}$ scale polynomially with $1/\varepsilon$ (as $O(\varepsilon^{-p})$ for some constant $p > 0$), the sum forms a geometric series dominated by its final term. Specifically,

$$\sum_{j=0}^{j^*} (N_{\mathcal{C}}(\varepsilon_j, \delta_j) + N_{\mathcal{T}}(n, \varepsilon_j, \delta_j)) \leq O(N_{\mathcal{C}}(\varepsilon_{j^*}, \delta_{j^*}) + N_{\mathcal{T}}(n, \varepsilon_{j^*}, \delta_{j^*})).$$

By our choices of ε_j , δ_j , and j^* , we have $\varepsilon_{j^*} = \Theta(\varepsilon^*)$ and $\delta_{j^*} = \Theta\left(\frac{\delta}{\log^2(1/\varepsilon^*)}\right)$. Moreover, by assumption, the sample complexities factor as $N(\varepsilon, \delta) = f(\varepsilon) \cdot \log(1/\delta)$. Substituting δ_{j^*} yields:

$$\log\left(\frac{1}{\delta_{j^*}}\right) = \log\left(\Theta\left(\frac{\log^2(1/\varepsilon^*)}{\delta}\right)\right) = \log\left(\frac{1}{\delta}\right) + 2 \log \log\left(\frac{1}{\varepsilon^*}\right) + O(1).$$

Therefore, the complexity $N(\varepsilon_{j^*}, \delta_{j^*})$ decomposes into the standard non-adaptive cost plus an additive overhead:

$$N(\varepsilon_{j^*}, \delta_{j^*}) = f(\varepsilon^*) \log\left(\frac{1}{\delta}\right) + f(\varepsilon^*) \cdot O\left(\log \log\left(\frac{1}{\varepsilon^*}\right)\right).$$

Applying this to both \mathcal{C} and \mathcal{T} , the total sample complexity is bounded by

$$O\left((N_{\mathcal{C}}(\varepsilon^*, \delta) + N_{\mathcal{T}}(n, \varepsilon^*, \delta)) \left[1 + \frac{\log \log(1/\varepsilon^*)}{\log(1/\delta)}\right]\right).$$

■