Closeness testing from distributed measurements

Clément Canonne CLEMENT.CANONNE@SYDNEY.EDU.AU
Aditya Vikram Singh ADITYAVIKRAM.SINGH@SYDNEY.EDU.AU
The University of Sydney

Editors: Matus Telgarsky and Jonathan Ullman

Abstract

We consider the fundamental task of two-sample composite hypothesis testing (i.e., "closeness testing") in a distributed setting, where a central party holds a dataset of $m \gg 1$ observations from an unknown discrete probability distribution q over a universe of size k, and n individual parties each independently observes one realization from an unknown distribution p. The goal is for the central party to test whether p and q are equal, or differ significantly in statistical distance, while only receiving a small amount of information (at most $\ell \leq \log_2 k$ bits) from each of the n distributed entities. Our main contribution is a time- and sample-efficient algorithm for this task, applicable across the whole regime of parameters. Our theoretical guarantees match the optimal sample complexities in the specific cases already studied in the literature, e.g., when $\ell = \log_2 k$ (no information constraint) or $m \to \infty$ (reference distribution fully known to the central party).

Keywords: distribution testing, distributed inference, communication constraints, two-sample testing, closeness testing

1. Introduction

A company has acquired a dataset S, containing m data points from a (large) universe of size $k \gg 1$. This dataset, while potentially valuable and useful to the company, is only worth analyzing and feeding to its various data processing pipelines to inform policies and decision making *if it is representative of the customers:* if the statistical features of S match those of the larger target population. Equivalently, the company seeks to check whether the distribution of these m data points matches the probability distribution of "true" customers' data. *This is good news:* this boils down to running a statistical test, a two-sample goodness-of-fit test for discrete distributions, for which a range of sample-optimal tests are known! *But this is bad news:* this means the company must obtain a number n of samples from this "true" customers' data distribution, exactly what it hoped to avoid by acquiring the dataset S. Gathering data from customers is a slow and costly process, and subject to many constraints – constraints of various sorts: technical, social, economic… What is the smallest number n of new, "true" data samples the company needs to obtain, in order to be confident that this dataset S is representative of the population?

This scenario is an example of the many scenarios our work sets out to address, focusing on a specific, common type of "technical constraint:" namely, a *bandwidth constraint*. Each of the n new data points, coming from the "true" (unknown) distribution p, must be quantized to at most ℓ bits before being sent to the central server. After gathering these n (quantized) independent data points from p, the central server – which also holds the m data points from the dataset S, independent realizations from some (unknown) distribution q – must run a statistical test to decide whether $p = q$ (the dataset

is representative of the population!), or, alternatively, if the two distributions are statistically far[1] (the dataset's distribution significantly deviates from the true data distribution). The motivation for addressing such a problem also arises from situations where it is expensive and/or cumbersome for one user to send a lot of data, but the number of users is large. For instance, the number of users possessing a smartphone device is large, but each device has multiple apps and services competing for finite resources; in such a case, it might be much more efficient and robust to ask a user to send, say, a 1-bit summary of their sample (and do this with a large number of users) than to make large demands from a few users (which might cause delays and failures). Similar considerations arise when it is more practical to deploy a large number of easy-to-manufacture low-power sensors to gather data, rather than a small number of expensive and sophisticated sensors. A bit more formally, the question we tackle in this paper can be phrased as follows:

> As a function of k, ℓ, m, and a distance parameter $\varepsilon > 0$, what is the minimum number n of independent data points needed to distinguish with high probability between $p = q$ and $\mathrm{TV}(p, q) > \varepsilon$, where TV denotes the statistical distance between distributions? And what (computationally efficient) algorithm achieves this guarantee?

To make our algorithms as general as possible, we further allow for the n observations to be quantized in a joint way: that is, the quantizers are allowed to depend on a common (short) random seed of up to s bits (independent of the n observations), where s is a parameter. This enables us to capture a wide range of distributed settings, including, for instance, those where the central server can broadcast a message to all distributed entities, or where the n external entities first agree on a randomized protocol before setting out to obtain their respective data points.

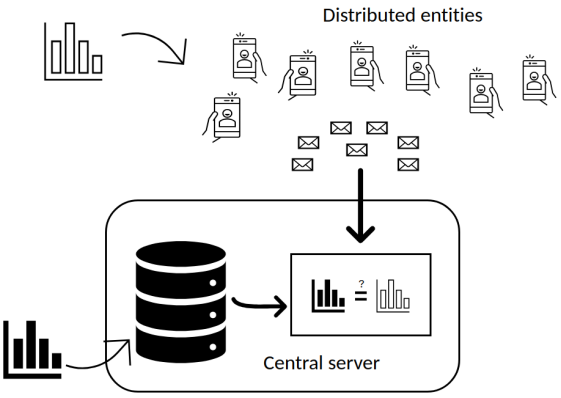

Figure 1: A depiction of the setting addressed by our work: n distributed entities each hold an (independent) observation from an unknown probability distribution p, and must quantize these observations into short ℓ-bit messages to send to a central server. The central server holds a dataset S comprising m independent samples from an unknown distribution q: based on these m observations and the n ℓ-bit messages, the center must decide whether p and q are equal, or statistically different.

1. Throughout, and following the standard in distribution testing, we use statistical distance (also known as total variation distance) as our measure of distance between probability distributions: $\mathrm{TV}(p, q) = \sup_{S \subseteq \Omega}(p(S) - q(S))$ for two distributions p, q over a measurable space Ω. See Section 2 for more.

The formal setting. Formalizing the above discussion leads to the following problem formulation: there is a (known) discrete domain $\mathcal{X}$ of size k, without loss of generality $\mathcal{X} = \{1, 2, \ldots, k\}$, as well as a communication constraint $1 \leq \ell \leq \log_2 k$, a target accuracy $\varepsilon \in (0, 1]$ and an integer m.
A *protocol* Π for the distributed closeness testing task with parameters k, ℓ, m, ε for n parties using s bits of common randomness is defined as follows. Π consists of:

- n (possibly randomized) *channels* $W_1, \ldots, W_n \colon \mathcal{X} \times \{0,1\}^s \to \mathcal{Y}$, where $\mathcal{Y} = \{0,1\}^\ell$
- A central (possibly randomized) algorithm $A \colon \mathcal{X}^m \times \mathcal{Y}^n \times \{0,1\}^s \to \{\mathsf{accept}, \mathsf{reject}\}$

Given $\Pi = (A, W_1, \ldots, W_n)$, the protocol is executed as follows:

1. A random seed $r \in \{0,1\}^s$ is drawn uniformly, and provided to the n parties and center $\mathcal{C}$.
2. A multiset $S \in \mathcal{X}^m$ of m i.i.d. samples from an unknown distribution q is provided to the center $\mathcal{C}$.
3. n samples $x_1, \ldots, x_n$ (independent from r) are drawn i.i.d. from an unknown distribution p over $\mathcal{X}$. Sample x_i is given to party i, which then computes $y_i = W_i(x_i, r) \in \{0,1\}^\ell$ and sends y_i to $\mathcal{C}$.
4. The center $\mathcal{C}$, upon receiving the tuple $T = (y_1, \ldots, y_n)$ of n messages, computes the outcome of the protocol, $A(S, T, r)$.

A *valid* protocol Π must have the following guarantees:

- If $p = q$, then $\Pr[A(S, T, r) = \mathsf{accept}] \geq \frac{9}{10}$;
- If $\mathrm{TV}(p, q) \geq \varepsilon$, then $\Pr[A(S, T, r) = \mathsf{reject}] \geq \frac{9}{10}$,

where the randomness is taken over the draw of the n samples from p, the m samples from q, as well as the uniformly random choice of the common seed $r \in \{0,1\}^s$ (and the internal randomness of A and $W_1, \ldots, W_n$). In other words, Π must output accept with high probability when $p = q$, and reject when p, q are at distance at least ε.
The objective is to, given k, ℓ, m, ε, as well as $s \geq 0$, to design a valid protocol Π with n as small as possible. We call this $n = n(k, \ell, m, \varepsilon, s)$ the *sample complexity of* Π.[2]

1.1. Our results

In order to put our results in context, recall that, absent any ℓ-bits communication constraints (that is, in the "centralized setting" where the same entity holds all the observations), it is known that $\Theta(k^{2/3}/\varepsilon^{4/3} \vee k^{1/2}/\varepsilon^2)$ samples from both p and q are necessary and sufficient in the "equal-sample" case (i.e., $n = m$) (Chan et al., 2014). For the *unequal sample size* generalization, where the number of samples from both distributions is not required to be the same, the optimal trade-off between m and n is known to be

$$n = \Theta\left(\frac{k}{m^{1/2}\varepsilon^2} \vee \frac{k^{1/2}}{\varepsilon^2}\right)$$

2. To keep the number of parameters (relatively) under control, we fixed in our formulation of the task both Type I and Type II errors to $1/10$. By standard arguments, one can then achieve any $\delta \in (0, 1]$, at cost of a $O(\log(1/\delta))$ factor in the sample complexity and the number of random bits s.

as long as $m = \Omega(k^{2/3}/\varepsilon^{4/3} \vee k^{1/2}/\varepsilon^2)$, and assuming without loss of generality (in this setting) $m \geq n$ (Bhattacharya and Valiant, 2015; Diakonikolas and Kane, 2016).
Another important point of comparison, *with* ℓ-bits communication constraints, is the optimal sample complexity of *identity* testing, where the distribution q is fully known to the center; which can be equivalently seen as the limiting case $m \to \infty$. From Acharya et al. (2020b,c), it is known that this sample complexity is

$$n = \Theta\left(\frac{k}{2^{\ell/2}\varepsilon^2}\left(\sqrt{\frac{k}{2^{\ell+s}}} \vee 1\right)\right)$$

With these baselines in hand, we are ready to state our results. For simplicity, we only provide here the most salient use cases, fixing the parameter s to common or best settings; the full theorem statement is provided as Theorem 7 in Section 3.

Theorem 1 *Suppose $\varepsilon \geq 1/k^{1/4}$. Then there exists a protocol for the distributed testing task with sample complexity*

$$n = O\left(\frac{k^4}{2^\ell m^4 \varepsilon^8} \vee \frac{k^2}{2^\ell m \varepsilon^4} \vee \frac{k}{2^{\ell/2}\varepsilon^2}\right)$$

for $\ell \ll \log k$, provided that $m = \Omega\big(k^{1/2}/\varepsilon^2\big)$. In particular, for $m = \Omega\big(k^{2/3}/\varepsilon^{4/3}\big)$,

$$n = O\left(\frac{k^2}{2^\ell m \varepsilon^4} \vee \frac{k}{2^{\ell/2}\varepsilon^2}\right)$$

is enough, and having $m = \Omega\big(k/(\varepsilon^2 2^{\ell/2})\big)$ is enough to achieve the optimal

$$n = O\left(\frac{k}{2^{\ell/2}\varepsilon^2}\right)$$

sample complexity of identity *testing.*

This last result is quite surprising: as discussed before, this corresponds to the sample complexity of the identity testing task, which corresponds to $m = \infty$. Yet, our result shows that it is enough to have the finite sample bound $m \gg k/(\varepsilon^2 2^{\ell/2})$! Now, one could argue that a finite bound is not *too* surprising, since as soon as the center has a number m of samples sufficient to *learn* q to high accuracy, one can hope closeness (unknown q) and identity testing (known q) become equivalent. But $\Omega(k/\varepsilon^2)$ samples would be necessary to learn q to TV distance ε: far greater than $\Omega\big(k/(\varepsilon^2 2^{\ell/2})\big)$.
The above theorem focuses on the "not-too-small" distance parameter regime. Our algorithm also applies to the "vanishing ε regime," with a much simpler bound to state.

Theorem 2 *Suppose $\varepsilon \leq 1/k^{1/4}$. Then there exists a protocol for the distributed testing task with sample complexity*

$$n = O\left(\frac{k^2}{2^\ell m \varepsilon^4} \vee \frac{k}{2^{\ell/2}\varepsilon^2}\right)$$

provided that $m = \Omega\big(k^{1/2}/\varepsilon^2\big)$.

To interpret this, note that (1) for $\ell = \log_2 k$ we exactly retrieve the optimal sample complexity of the centralized setting (as we should, as the communication constraint vanishes for that value of ℓ), while (2) we again retrieve the known optimal sample complexity of identity testing whenever

$m \gg k/(\varepsilon^2 2^{\ell/2})$. Finally, the restriction (in both theorems) to $m = \Omega\big(k^{1/2}/\varepsilon^2\big)$ is expected, as this is already information-theoretically necessary in the centralized setting, *even* if p was fully known. While we do not establish proof of optimality in all parameter regimes (which we see as the main limitation of our work), the above discussion shows our bounds are optimal (up to constant factors) in several important specific cases (the settings previously studied), and smoothly interpolate between these bounds in the new parameter regimes. This provides strong evidence for our conjecture that our bounds are, in fact, optimal (up to constant factors) for *all* parameter regimes. Another important feature of our results is that all these bounds are achieved by a *single* protocol Π, instantiated with different parameters (in particular, for suitable settings of the common random seed size s). In this sense, our results are much more general and flexible than the simple cases stated above, as they enable one to take full advantage of whichever amount of common randomness s is available (in case the "optimal" setting to use, which underlies the theorem statements, is too high).

And indeed, in this respect, one intriguing feature of our algorithm is that, even when there is a large common random seed available to all n parties, it is not always necessary to use *all* of it in order to achieve the best sample complexity (we elaborate on the reason for this in the next subsection). This is in contrast to the sample-optimal distributed *identity* testing algorithm of Acharya et al. (2020c), where the distributed q is fully known to the central server, and for which the authors show that using as much common randomness as possible is always the best strategy.

Possible extensions and future directions. As noted above, we conjecture that the sample complexity achieved by our protocol is optimal, up to constants; establishing this has proved to be quite challenging, especially in view of the number of parameters to balance $(k, \varepsilon, \ell, m, n, s)$, and constitutes an obvious direction of future research. Another interesting extension of our results would be to solve a version of the distributed closeness testing protocol where each user has multiple samples. A user having multiple samples widens the scope of protocols that may be used; *e.g.*, while a user cannot form an unbiased estimate of the squared ℓ_2 norm of their distribution from one sample, this becomes possible when given access to at least two, as $\Pr_{X_1,X_2\sim\mathbf{p}}[X_1 = X_2] = \|\mathbf{p}\|_2^2$.

1.2. Overview of our techniques

At a very high level, our algorithm can be seen as a three-stage process which *reduces the distributed setting to the centralized one* (as the cost of a necessary loss in parameters due to the communication constraints), followed by the use of an off-the-shelf testing algorithm for this well-understood centralized setting. First, *compress* the domain size using common randomness, to reduce the cost of the communication constraints (the smaller the domain, the less stringent the ℓ-bit communication constraint is); then, *simulate* samples (at the center) from the "compressed probability distribution" in a distributed fashion; and finally, *expand* the domain at the center side, to make the resulting task more amenable to the off-the-shelf centralized tester.

We now give a simplified description of our algorithm in terms of the basic building blocks while keeping track of the relevant parameters. (The full algorithm needs to perform a careful distinction of cases, depending on the values of k, m, and ε, in order to choose how much of the s bits of the common random seed to use in the first stage, and the best parameters to select when invoking these building blocks.)

1. Using DOMAIN COMPRESSION with the s bits of common randomness applied to the n samples from p and the multiset S of m samples from q, we go from $\mathcal{X}$ to a smaller domain

$\mathcal{X}'$ of size $k' := k/2^s$, contracting the distance parameter from ε to $\varepsilon' := \varepsilon/\sqrt{2^s}$. This results in n (distributed) samples from some p' and a multiset S' of m i.i.d. samples from some q', both probability distributions on $\mathcal{X}'$. *This is the only use of the common randomness.*

$$(k, \varepsilon, \ell, m, n, s) \mapsto (k', \varepsilon', \ell, m, n, 0)$$

2. Using DISTRIBUTED SIMULATION on the n distributed samples from p' (subject to the ℓ-bit communication constraint), the n parties allow the center $\mathcal{C}$ to simulate $n' := n \cdot 2^\ell/k' \ll n$ i.i.d. samples from p': *this handles the distributed setting*, reducing it to the centralized setting with fewer samples.
$$(k', \varepsilon', \ell, m, n, 0) \mapsto (k', \varepsilon', \infty, m, n', 0)$$

3. Using DOMAIN EXPANSION using a subset of the $m = n' + O(m')$ samples from q', "flatten" the distribution q' into a distribution $\bar{q}$ on a larger domain $\bar{\mathcal{X}}$ of size $\bar{k} := k' + O(m')$. Apply the same transformation to the n' samples from p', to obtain n' i.i.d. samples from some distribution $\bar{p}$ on the same larger domain $\bar{\mathcal{X}}$. This is all done at the center $\mathcal{C}$, and "consumes" $m - O(m')$ of the samples from q'; while increasing the domain size and reducing the number of available samples (from $\bar{q}$) to n', this preserves the distance parameter ε', and crucially *reduces the ℓ_2 norm of one of the distributions*, ensuring that $\|\bar{q}\|_2^2 = O(1/\bar{k})$.
$$(k', \varepsilon', \ell, m, n, 0) \mapsto (\bar{k}, \varepsilon', \infty, n', n', 0)$$

4. Using an optimal (known) CLOSENESS TESTER for ℓ_2 on $\bar{p}, \bar{q}$ (both with n' i.i.d. samples, on the domain $\bar{\mathcal{X}}$ of size $\bar{k}$) with ℓ_2 distance parameter $\varepsilon_{\ell_2} := \varepsilon'/\sqrt{\bar{k}}$, and leveraging the guarantee that $\|\bar{q}\|_2^2 = O(1/\bar{k})$, we conclude by testing whether $\bar{p} = \bar{q}$ or $\|\bar{p} - \bar{q}\|_2 > \varepsilon_{\ell_2}$ (which is, by Cauchy–Schwarz, implied by $\mathrm{TV}(\bar{p}, \bar{q}) > \varepsilon'$). Choosing the best value of s possible, and minding the various implicit constraints on $k, \varepsilon, n, m, \ell$ (swept under the rug here) leads to the result.

Now, as alluded above, in the first stage it is sometimes better *not* to use the full amount of common randomness available to perform DOMAIN COMPRESSION, leading to some interesting case distinctions to get the best possible sample complexity in our protocol. While this may seem counterintuitive, this can be explained as follows: first, each of the n parties has a communication budget of ℓ, and so can send *without any loss or compression required* any element of a domain of size up to 2^ℓ. As a result, compressing the domain size to $k' < 2^\ell$ would be counterproductive (as the distance parameter ε' would shrink unnecessarily as well in the process, and this impacts the resulting sample complexity). The other reason is that, to test ε'-closeness of $\bar{p}$ and $\bar{q}$ at the center, we information-theoretically need at least $\sqrt{k'}/\varepsilon'^2 = \sqrt{k2^s}/\varepsilon^2$ samples from both distributions (Paninski, 2008; Diakonikolas and Kane, 2016); in particular, since the center has a fixed number of samples m, this imposes $m \geq \sqrt{k2^s}/\varepsilon^2$, or, equivalently, that we cannot use more that $\log_2(m^2\varepsilon^4/k)$ bits of common randomness.

1.3. Related work

There is a long and rich line of research on distribution testing, starting with the influential work of Batu et al. (2000). We only here discuss the most relevant to this work, and refer the interested reader to the surveys (Rubinfeld, 2012; Canonne, 2020; Balakrishnan and Wasserman, 2018),

monograph (Canonne, 2022) and textbook (Goldreich, 2017, Chapter 11) for a comprehensive overview.

Closeness testing in the centralized setting (i.e., two-sample goodness-of-fit) with *unequal samples* (i.e., the non-distributed version of the problem we consider here) was introduced in Acharya et al. (2014), with its optimal sample complexity pinpointed in Bhattacharya and Valiant (2015); Diakonikolas and Kane (2016). In the distributed setting, the systematic study of (discrete) distribution learning and testing with communication (and, more generally, local information constraints) was initiated in Acharya et al. (2020a), with follow-up works in Acharya et al. (2020b), and Acharya et al. (2021), focusing respectively on communication and then local privacy constraints. Acharya et al. (2020c) established the optimal sample complexity of the task including the dependency on the amount of common randomness s available, for *identity* testing (i.e., one-sample goodness-of-fit), which can be seen as a special, easier case of closeness testing.

Acharya et al. (2024) then generalized some of the techniques, and in particular distributed simulation, to design optimal algorithms *nonparametric* distribution learning (i.e., estimation) under communication constraints; while Acharya et al. (2023) developed a general information-theoretic framework to prove information-theoretic lower bounds for these local information constraint inference questions.

Finally, Diakonikolas et al. (2019) study analogous questions, namely identity and closeness testing in a distributed setting, under a different distributed model where entities hold a (small) number of samples from both distributions, and one seeks to minimize the *total* communication between all parties. The results obtained, as a result, are incomparable to ours.

2. Preliminaries

To avoid notational clutter, everything we state holds up to constant factors. Throughout, we use standard asymptotic notation ($O(\cdot), \Omega(\cdot), \Theta(\cdot)$), and for simplicity will omit floors and ceilings when clear from context. We also rely on the following notation: for two sequences $(a_n)_n, (b_n)_n$, $a_n \lesssim b_n$ indicates that there exist an absolute constant $C > 0$ such that $a_n \leq C \cdot b_n$ for all n. $a_n \asymp b_n$ then means that both $a_n \lesssim b_n$ and $b_n \lesssim a_n$ hold. We write $\wedge$ (resp. $\vee$) to denote the minimum (resp. maximum) of two values.

We identify a probability distribution p over a finite domain $\mathcal{X}$ of size k to its probability mass function (pmf), that is, we view p as a k-dimensional vector such that $0 \leq p_x \leq 1$ for all $x \in \mathcal{X}$ and $\sum_{x\in\mathcal{X}} p_x = 1$. The *total variation distance* (or statistical distance) between two probability distributions over $\mathcal{X}$ is then the metric defined by

$$\mathrm{TV}(p,q) = \sup_{S\subseteq\mathcal{X}} (p(S) - q(S)) = \frac{1}{2}\sum_{x\in\mathcal{X}} |p_x - q_x| = \frac{1}{2}\|p-q\|_1$$

where for a subset $S \subseteq \mathcal{X}$ we write $p(S) = \sum_{x\in S} p_x$. Finally, we write $\mathrm{Poi}(\lambda)$ for the Poisson distribution with parameter $\lambda \geq 0$. We refer the reader to the surveys (Canonne, 2022) and textbook (Goldreich, 2017, Chapter 11) for further details on this metric, and the use of Poissonization in distribution testing.

2.1. Centralized closeness testing

As already mentioned, our algorithm works by reducing the problem of distributed closeness testing to one of centralized closeness testing. Here, we review the centralized closeness testing paradigm

of Diakonikolas and Kane (2016) that optimally tests closeness in total variation (i.e., ℓ_1 norm) by using domain expansion followed by running an optimal ℓ_2-closeness tester. The following lemma shows existence of an algorithm that tests closeness in total variation but depends on the ℓ_2 norms of the distributions being tested.

Lemma 3 ((Diakonikolas and Kane, 2016, Lemma 2.3)) *Let p and q be two unknown distributions on $\mathcal{X}$ of cardinality k. There exists an algorithm that, on inputs k, $\varepsilon > 0$ and $b \geq (\|p\|_2 \wedge \|q\|_2)$, draws $n = O(bk/\varepsilon^2)$ samples from each of p and q, and, with probability at least $2/3$, distinguishes between $p = q$ versus $\mathrm{TV}(p, q) > \varepsilon$.*

We note that the tester in this lemma invokes an optimal ℓ_2-closeness tester, e.g., that of Chan et al. (2014). Henceforth, we will refer to the tester in Lemma 3 as ℓ_2-CLOSENESS-TESTER. Furthermore, this lemma shows that if the ℓ_2 norm of even one of the distributions is small, the sample complexity of closeness testing is small as well, which motivates the notion of domain expansion.

Domain expansion. Let $\mathcal{X}$ be a domain of cardinality k. Then, given a multiset $S \in \mathcal{X}^m$, define an expanded domain $\bar{\mathcal{X}}$ of cardinality $k + m$ as follows: if an element $x \in \mathcal{X}$ occurs $n_x \in [0, m]$ times in S, the new domain $\bar{\mathcal{X}}$ has the corresponding elements $(0, x), (1, x), \ldots, (n_x, x)$ in it. Now, let q be a distribution on $\mathcal{X}$. Define the induced distribution $\bar{q}$ on the expanded domain $\bar{\mathcal{X}}$ by dividing $q(x)$ uniformly over the elements $(0, x), (1, x), \ldots, (n_x, x)$; i.e. $\bar{q}((i, x)) = q(x)/(1 + n_x)$. Note that, given a sample x from q, a sample $\bar{x}$ from $\bar{q}$ can be created by uniformly randomly choosing an element from the set $\{(0, x), (1, x), \ldots, (n_x, x)\}$. It is apparent that, for any two distributions p, q on $\mathcal{X}$ and the corresponding induced distributions $\bar{p}, \bar{q}$ on $\bar{\mathcal{X}}$, we have that $\mathrm{TV}(p, q) = \mathrm{TV}(\bar{p}, \bar{q})$. This means that testing closeness between p and q can be accomplished by testing closeness between $\bar{p}$ and $\bar{q}$; the following lemma from Diakonikolas and Kane (2016) illustrates why doing so might be helpful.

Lemma 4 ((Diakonikolas and Kane, 2016, Lemma 2.6)) *Let q be a distribution on $\mathcal{X}$ of cardinality k. Let $M \sim \mathrm{Poi}(m)$, i.e., M is sampled from a Poisson distribution with parameter m. Given M, form a multiset S by taking M samples from q, and do a domain expansion using S. Let $\bar{q}$ be the induced distribution on the expanded domain $\bar{\mathcal{X}}$. Then, $\mathbb{E}[\|\bar{q}\|_2] \leq 1/m$. In particular, this implies that with high (constant) probability, $|S| = O(m)$ and $\|\bar{q}\|_2 = O(1/\sqrt{m})$.*

Centralized closeness testing algorithm. Using domain expansion followed by using ℓ_2-CLOSENESS-TESTER (Lemma 3) gives us an optimal tester in the centralized setting, which is the algorithm proposed by Diakonikolas and Kane (2016). We recall this algorithm below (Algorithm 1), having tailored it to correspond with our distributed setting: i.e., we assume that there is a "center" that has a multiset S of m i.i.d. samples from q; we want to see how many samples n is required from distribution p to test closeness of p and q. Recall that, for existence of a valid testing algorithm, it is necessary that $m \geq \sqrt{k}/\varepsilon^2$.

Note that, for DOMAIN EXPANSION, the algorithm uses samples from center's multiset S (containing samples from q) only if $m \geq k^{2/3}/\varepsilon^{4/3}$; otherwise, the samples for this task are collected from p. We now give a short analysis of Algorithm 1 to make sense of the values chosen in the algorithm.

Analysis for $k\varepsilon^4 \geq 1$: Suppose center has $m \geq k^{2/3}/\varepsilon^{4/3}$ samples from q. Center performs DOMAIN EXPANSION with $m' \asymp \min\{m, k\}$ samples. After this, the number of samples required by ℓ_2-CLOSENESS-TESTER from $\bar{p}$ and $\bar{q}$ is $k/(\sqrt{m'}\varepsilon^2)$ (Lemma 3). Thus, the total number of samples

Algorithm 1: Centralized Closeness Testing

Input: Multiset $S \in \mathcal{X}^m$ of samples from q; sample access from p; distance parameter ε

if $m < \frac{\sqrt{k}}{\varepsilon^2}$ **then**
  Output "Number of samples insufficient" and abort
end
if $m \geq k^{2/3}/\varepsilon^{4/3}$ **then**
  Let $m' \asymp \min\{m, k\}$ and $n \asymp k/(\sqrt{m'}\varepsilon^2)$
  Perform DOMAIN EXPANSION using $\mathrm{Poi}(m')$ samples from S; call the resulting distributions $\bar{p}$, $\bar{q}$
  Get n samples from p and use them to create n samples from $\bar{p}$
  Use n samples from S to create n samples from $\bar{q}$
  Run ℓ_2-CLOSENESS-TESTER to test closeness between $\bar{p}$ and $\bar{q}$ using (n, n) samples
end
else if $m < k^{2/3}/\varepsilon^{4/3}$ **then**
  Let $n' \asymp k^2/(m^2\varepsilon^4)$
  Perform DOMAIN EXPANSION using $\mathrm{Poi}(n')$ samples from p; call the resulting distributions $\bar{p}$, $\bar{q}$
  Get m samples from p and use them to create m samples from $\bar{p}$
  Use m samples from S to create m samples from $\bar{q}$
  Run ℓ_2-CLOSENESS-TESTER to test closeness between $\bar{p}$ and $\bar{q}$ using (m, m) samples
end
Output the result of ℓ_2-CLOSENESS-TESTER

used from q is $\left(m' + k/(\sqrt{m'}\varepsilon^2)\right) \asymp m' \leq m$; and the number of samples required from p is $n = k/(\sqrt{m'}\varepsilon^2) = \max\{k/(\sqrt{m}\varepsilon^2), \sqrt{k}/\varepsilon^2\}$.

On the other hand, suppose center has $\sqrt{k}/\varepsilon^2 \leq m < k^{2/3}/\varepsilon^{4/3}$ samples from q. In this case, DOMAIN EXPANSION is done using samples from p: let the number of samples used for this be n'. After this, the number of samples required by ℓ_2-CLOSENESS-TESTER from $\bar{p}$ and $\bar{q}$ is $k/(\sqrt{n'}\varepsilon^2)$ (Lemma 3). Since center has m samples, we have $m = k/(\sqrt{n'}\varepsilon^2)$, which gives $n' = k^2/(m^2\varepsilon^4)$. This means that the total number of samples required from p is $n = n' + k/(\sqrt{n'}\varepsilon^2) \asymp k^2/(m^2\varepsilon^4)$.

Analysis for $k\varepsilon^4 < 1$: In this case, we have $k < \frac{k^{2/3}}{\varepsilon^{4/3}} < \frac{\sqrt{k}}{\varepsilon^2}$. Thus, center uses $m' = k$ samples from q for domain expansion, after which ℓ_2-CLOSENESS-TESTER requires $\sqrt{k}/\varepsilon^2$ samples from $\bar{p}$ and $\bar{q}$; in particular, the total number of samples required from p is $n = \sqrt{k}/\varepsilon^2$.

3. The main algorithm

Our algorithm for testing closeness in the distributed setting reduces the problem to that of testing closeness in the centralized setting (with modified parameters). To achieve this reduction, we rely on two useful ideas from the distributed testing literature.

(i) Distributed simulation. Suppose n distributed parties (called "players" henceforth) all hold an independent sample from p, but each player can only send an ℓ-bit message to the center. Can the players craft their message in a way that allows the center to simulate a sample from p based on the

ℓ-bit messages from possibly multiple players? This is indeed possible, as shown in Acharya et al. (2020b). For our algorithm, we use the following theoretical guarantee of the *distributed simulation* primitive of Acharya et al. (2020b) (see also (Canonne, 2022, Section 4.2) for an exposition), which can be found in Acharya et al. (2024):

Lemma 5 ((Acharya et al., 2024, Fact 6)) *For any $1 \leq \ell < \log k$, there exists a (randomized) simulation protocol (denoted* DISTRSIM$_\ell$*) that lets the referee simulate (with probability at least $1 - e^{-\frac{n2^\ell}{32k}}$) a multiset of $n2^\ell/(8k)$ i.i.d. samples from an unknown k-ary probability distribution p using ℓ-bit messages from n players, where each player holds an independent sample from p. Moreover, the protocol is deterministic at the players' side, and only requires private randomness at the center.*

We note that the distributed simulation protocol in Lemma 5 is allowed to output "abort" sometimes, and conditioned on the output being not "abort", the distribution of the samples output by the protocol is exactly the original distribution; the lemma guarantees that the probability of "abort" is exponentially small.

(ii) Domain compression. Note that DISTRSIM$_\ell$ is already sufficient to reduce the problem of distributed testing to that of centralized testing – the sample complexity simply blows up by a factor of $k/2^\ell$. But we can do much better if the players and the center have access to common randomness. A primitive that optimally exploits common randomness to reduce sample complexity of distributed protocols is proposed in Acharya et al. (2020c). It does so by using randomness to "compress the domain" – the randomness must be shared so that every party agrees to the new domain. We state here a self-contained version of the randomness-efficient domain compression primitive from Acharya et al. (2020c).

Lemma 6 ((Acharya et al., 2020c, Theorem 3.2)) *Let $\mathcal{X}$ be such that $|\mathcal{X}| = k$. Then, there exist absolute constants $s_0, c_0, c', \delta_0 > 0$ such that the following holds. For every $s_0 < s < \log k - c_0$, there exist (efficient) mappings $\Psi_u : \mathcal{X} \to \mathcal{X}'$, $u \in \{0,1\}^s$, satisfying*

$$\Pr_{U \sim \mathrm{Unif}\{0,1\}^s}\left[\mathrm{TV}(p_U, q_U) \geq \frac{c'\varepsilon}{2^{s/2}}\right] \geq 1 - \delta_0,$$

where p, q are arbitrary distributions on $\mathcal{X}$ satisfying $\mathrm{TV}(p,q) \geq \varepsilon$; p_U denotes the distribution on $\mathcal{X}'$ induced by p and Ψ_u via $p_U(x') = p(x : \Psi_u(x) = x')$; and the cardinality of $\mathcal{X}'$ is $|\mathcal{X}'| = ck/2^s$.

Furthermore, this domain compression protocol can be run in parallel *without using additional randomness* in order to amplify the probability of success of any testing algorithm that uses samples from p_U and q_U (see Lemma 4.2 in Acharya et al. (2020c)).

Combining compression and simulation. As noted earlier, compressing the domain all the way to size 2^ℓ may not be optimal. In such cases, where the cardinality of the compressed domain k' is strictly more than 2^ℓ, our algorithm uses DISTRSIM$_\ell$ to simulate samples from the compressed domain. Algorithm 2 (player's side) and Algorithm 3 (center's side) together specify our overall protocol for distributed closeness testing.

Sample complexity. Suppose the algorithm uses s bits of common randomness for domain compression. Let the induced distribution on the new domain be p' (corresponding to distribution p of players) and q' (corresponding to distribution q of center). Center then outputs its verdict based on testing

Algorithm 2: Distributed Closeness Testing (Player i)

Input: sample x_i from p; s; access to common randomness; ℓ

1 Perform DOMAIN COMPRESSION using s bits of common randomness. Denoting the compressed domain by $\mathcal{X}'$, let $k' := |\mathcal{X}'| = k/2^s$.

2 Compute $x'_i \in \mathcal{X}'$ (the sample induced on $\mathcal{X}'$ by $x_i \in \mathcal{X}$)

3 Send ℓ-bit message y_i to the center $\mathcal{C}$, where

$$y_i \leftarrow \begin{cases} x'_i, & \text{if } 2^\ell \geq k' \\ \textsc{DistrSim}_\ell(i, x'_i), & \text{if } 2^\ell < k' \end{cases}$$

closeness between p' and q' with parameters $k' = k/2^s$ and $\varepsilon' = \varepsilon/\sqrt{2^s}$. Note that $k'/\varepsilon'^2 = k/\varepsilon^2$ which, in particular, implies that $k'^{2/3}/\varepsilon'^{4/3} = k^{2/3}/\varepsilon^{4/3}$.

Case 1: $m \geq k'^{2/3}/\varepsilon'^{4/3}$. In this case, center does domain expansion using $\mathrm{Poi}(\bar{m})$ samples from the multiset available with it, where $\bar{m} \asymp \min\{m, k'\}$. For closeness testing, center obtains $n' \asymp k'/(\sqrt{\bar{m}}\varepsilon'^2)$ samples using DISTSIM with $n = n'\big(k'/2^\ell\big)$ players. Thus

$$n \asymp \frac{n'k'}{2^\ell} = \frac{k'^2}{\varepsilon'^2 2^\ell \sqrt{\bar{m}}} = \frac{k'^2}{\varepsilon'^2 2^\ell \sqrt{m \wedge k'}} = \frac{k^2}{\varepsilon^2 2^\ell 2^s \sqrt{m \wedge (k/2^s)}} = \frac{k^2}{\varepsilon^2 2^\ell 2^s \sqrt{m}} \vee \frac{k^{3/2}}{\varepsilon^2 2^\ell 2^{s/2}}$$

Case 2: $m < k'^{2/3}/\varepsilon'^{4/3}$. In this case, center does domain expansion using $\mathrm{Poi}(\bar{n})$ samples from p' that it obtains using DISTSIM with $n_1 = \bar{n}\big(k'/2^\ell\big)$ players. Here, $\bar{n}$ is chosen so that ℓ_2 closeness testing done after domain expansion requires m samples each from $\bar{p}$ and $\bar{q}$. As in the analysis of the unconstrained setting, this is ensured by setting $\bar{n} \asymp k'^2/(m^2\varepsilon'^4)$. Center obtains m samples each from $\bar{p}$ using DISTSIM with $n_2 = m\big(k'/2^\ell\big)$ players. Thus

$$n = n_1 + n_2 = (\bar{n} + m)\frac{k'}{2^\ell} = \left(\frac{k'^2}{m^2\varepsilon'^4} + m\right)\frac{k'}{2^\ell} \asymp \frac{k'^2}{m^2\varepsilon'^4}\frac{k'}{2^\ell} = \frac{k^3}{\varepsilon^4 2^\ell 2^s m^2}$$

Final sample complexity. Doing a case-by-case analysis (details in the supplementary) and setting s to minimize n finally gives us the following theorem (where s_{part} and s_{full} are such that $2^{s_{\text{part}}} = m^2\varepsilon^4/k$ and $2^{s_{\text{full}}} = k/2^\ell$).

Theorem 7 *The distributed closeness testing protocol specified by Algorithm 2 and Algorithm 3 has the following sample complexity:*
For $k\varepsilon^4 > 1$ (in which case, $\frac{\sqrt{k}}{\varepsilon^2} < \frac{k^{2/3}}{\varepsilon^{4/3}} < k < \frac{k}{\varepsilon^2}$):

<table>
<tr><td></td><td colspan="2">$\frac{\sqrt{k}}{\varepsilon^2} \leq m < \frac{k^{2/3}}{\varepsilon^{4/3}}$</td><td colspan="2">$\frac{k^{2/3}}{\varepsilon^{4/3}} \leq m \leq \frac{k}{\varepsilon^2}$</td></tr>
<tr><td rowspan="2">$1 \leq 2^\ell \leq \frac{k^{2/3}}{\varepsilon^{4/3}}$</td><td colspan="2" rowspan="2">$n = \frac{k^4}{\varepsilon^8 m^4 2^\ell}$
$(s = s_{\text{part}})$</td><td>$\frac{k^{2/3}}{\varepsilon^{4/3}} \leq m < \frac{k}{\varepsilon^2 2^{\ell/2}}$</td><td>$\frac{k}{\varepsilon^2 2^{\ell/2}} \leq m \leq \frac{k}{\varepsilon^2}$</td></tr>
<tr><td>$n = \frac{k^2}{\varepsilon^4 m 2^\ell}$
$(s = s_{\text{part}})$</td><td>$n = \frac{k}{\varepsilon^2 2^{\ell/2}}$
$(s = s_{\text{full}})$</td></tr>
<tr><td rowspan="2">$\frac{k^{2/3}}{\varepsilon^{4/3}} < 2^\ell \leq k$</td><td>$\frac{\sqrt{k}}{\varepsilon^2} \leq m < \frac{k}{\varepsilon^2 2^{\ell/2}}$</td><td>$\frac{k}{\varepsilon^2 2^{\ell/2}} \leq m < \frac{k^{2/3}}{\varepsilon^{4/3}}$</td><td colspan="2" rowspan="2">$n = \frac{k}{\varepsilon^2 2^{\ell/2}}$
$(s = s_{\text{full}})$</td></tr>
<tr><td>$n = \frac{k^4}{\varepsilon^8 m^4 2^\ell}$
$(s = s_{\text{part}})$</td><td>$n = \frac{k^2}{\varepsilon^4 m^2}$
$(s = s_{\text{full}})$</td></tr>
</table>

Algorithm 3: Distributed Closeness Testing (Center)

Input: multiset $S \in \mathcal{X}^m$ of samples from q; access to common randomness; k; ε; ℓ

Decide the number of random bits s to use based on m, ℓ and amount of randomness available; broadcast s to players

(Assuming no limit on amount of randomness available, we set $s = s_{\mathrm{part}}$ or $s = s_{\mathrm{full}}$)

Perform DOMAIN COMPRESSION using s bits of common randomness. Call the compressed domain $\mathcal{X}'$. Let $k' = |\mathcal{X}'| = k/2^s$. Let $\varepsilon' = \varepsilon/\sqrt{2^s}$. (Call the induced distributions on $\mathcal{X}'$ as p', q')

Compute multiset $S' \in \mathcal{X}'^m$ induced on the compressed domain by $S \in \mathcal{X}^m$

if $m < \frac{\sqrt{k'}}{\varepsilon'^2}$ **then**

  Output "Number of samples insufficient to test for the given s" and abort

end

if $m \geq k'^{2/3}/\varepsilon'^{4/3}$ **then**

  Let $\bar{m} \asymp \min\{m, k'\}$ and $n' \asymp k'/(\sqrt{\bar{m}}\varepsilon'^2)$

  Perform DOMAIN EXPANSION using $\mathrm{Poi}(\bar{m})$ samples from S'; call the resulting distributions $\bar{p}$, $\bar{q}$

  Get n' samples from p' using DISTRSIM on $n = n'\big(k'/2^\ell\big)$ players

  Use n' samples from p' to create n' samples from $\bar{p}$

  Use n' samples from S' to create n' samples from $\bar{q}$

  Run ℓ_2-CLOSENESS-TESTER to test closeness between $\bar{p}$ and $\bar{q}$ using (n', n') samples

else if $m < k'^{2/3}/\varepsilon'^{4/3}$ **then**

  Let $\bar{n} \asymp k'^2/(m^2\varepsilon'^4)$

  Get $\bar{n}$ samples from p' using DISTRSIM on $n_1 = \bar{n}\big(k'/2^\ell\big)$ players

  Perform DOMAIN EXPANSION using $\mathrm{Poi}(\bar{n})$ samples from p'; call the resulting distributions $\bar{p}$, $\bar{q}$

  Get m samples from p' using DISTRSIM on $n_2 = m\big(k'/2^\ell\big)$ players

  Use m samples from p' to create m samples from $\bar{p}$

  Use m samples from S' to create m samples from $\bar{q}$

  Run ℓ_2-CLOSENESS-TESTER to test closeness between $\bar{p}$ and $\bar{q}$ using (m, m) samples

Output the result of ℓ_2-CLOSENESS-TESTER

For $k\varepsilon^4 \leq 1$ (in which case, $k < \frac{k^{2/3}}{\varepsilon^{4/3}} < \frac{\sqrt{k}}{\varepsilon^2} < \frac{k}{\varepsilon^2}$):

	$\frac{\sqrt{k}}{\varepsilon^2} \leq m < \frac{k}{\varepsilon^2 2^{\ell/2}}$	$\frac{k}{\varepsilon^2 2^{\ell/2}} \leq m \leq \frac{k}{\varepsilon^2}$
$1 \leq 2^\ell \leq k$	$n = \frac{k^2}{\varepsilon^4 m 2^\ell}$ $(s = s_{\mathrm{part}})$	$n = \frac{k}{\varepsilon^2 2^{\ell/2}}$ $(s = s_{\mathrm{full}})$

References

Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, and Ananda Theertha Suresh. Sublinear algorithms for outlier detection and generalized closeness testing. In *ISIT*, pages 3200–3204. IEEE, 2014.

Jayadev Acharya, Clément L. Canonne, and Himanshu Tyagi. Inference under information constraints I: Lower bounds from chi-square contraction. *IEEE Trans. Inform. Theory*, 66(12):7835–7855, 2020a. ISSN 0018-9448. doi: 10.1109/TIT.2020.3028440. URL https://doi.org/10.1109/TIT.2020.3028440. Preprint available at arXiv:abs/1812.11476.

Jayadev Acharya, Clément L. Canonne, and Himanshu Tyagi. Inference under information constraints II: Communication constraints and shared randomness. *IEEE Trans. Inform. Theory*, 66(12):7856–7877, 2020b. ISSN 0018-9448. doi: 10.1109/TIT.2020.3028439. URL https://doi.org/10.1109/TIT.2020.3028439.

Jayadev Acharya, Clément L. Canonne, Yanjun Han, Ziteng Sun, and Himanshu Tyagi. Domain compression and its application to randomness-optimal distributed goodness-of-fit. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3–40. PMLR, 09–12 Jul 2020c. URL http://proceedings.mlr.press/v125/acharya20a.html.

Jayadev Acharya, Clément L. Canonne, Cody Freitag, Ziteng Sun, and Himanshu Tyagi. Inference under information constraints III: local privacy constraints. *IEEE J. Sel. Areas Inf. Theory*, 2(1): 253–267, 2021.

Jayadev Acharya, Clément L. Canonne, Ziteng Sun, and Himanshu Tyagi. Unified lower bounds for interactive high-dimensional estimation under information constraints. In *NeurIPS*, 2023.

Jayadev Acharya, Clément L. Canonne, Aditya Vikram Singh, and Himanshu Tyagi. Optimal rates for nonparametric density estimation under communication constraints. *IEEE Trans. Inform. Theory*, 70(3):1939–1961, 2024. ISSN 0018-9448. doi: 10.1109/tit.2023.3325902. URL https://doi.org/10.1109/tit.2023.3325902.

Sivaraman Balakrishnan and Larry Wasserman. Hypothesis testing for high-dimensional multinomials: a selective review. *Ann. Appl. Stat.*, 12(2):727–749, 2018. ISSN 1932-6157. doi: 10.1214/18-AOAS1155SF. URL https://doi.org/10.1214/18-AOAS1155SF.

Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing that distributions are close. In *FOCS*, pages 189–197, 2000.

Bhaswar B. Bhattacharya and Gregory Valiant. Testing closeness with unequal sized samples. In *NeurIPS*, pages 2611–2619, 2015.

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

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

Siu On Chan, Ilias Diakonikolas, Paul Valiant, and Gregory Valiant. Optimal algorithms for testing closeness of discrete distributions. In *SODA*, pages 1193–1203. SIAM, 2014.

Ilias Diakonikolas and Daniel M. Kane. A new approach for testing properties of discrete distributions. In *FOCS*. IEEE Computer Society, 2016.

Ilias Diakonikolas, Themis Gouleakis, Daniel M. Kane, and Sankeerth Rao. Communication and memory efficient testing of discrete distributions. In *COLT*, volume 99 of *Proceedings of Machine Learning Research*, pages 1070–1106. PMLR, 2019.

Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. ISBN 978-1-107-19405-2.

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

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

Appendix A. Deferred Proofs

In this appendix, we provide the details of the analysis of our main algorithm (Theorem 7), restated below:

Theorem 8 *The distributed closeness testing protocol specified by Algorithm 2 and Algorithm 3 is a valid protocol whenever $m = \Omega(\sqrt{k}/\varepsilon^2)$, with the following sample complexity:*

- *If $k\varepsilon^4 > 1$ (in which case, $\frac{\sqrt{k}}{\varepsilon^2} < \frac{k^{2/3}}{\varepsilon^{4/3}} < k < \frac{k}{\varepsilon^2}$):*
  - *If $\frac{\sqrt{k}}{\varepsilon^2} \le m < \frac{k^{2/3}}{\varepsilon^{4/3}}$*

$$n = \begin{cases} O\left(\frac{k^2}{\varepsilon^4 m^2}\right) & \text{if } \frac{k^{2/3}}{\varepsilon^{4/3}} < 2^\ell \le k \text{ and } m > \frac{k}{\varepsilon^2 2^{\ell/2}}, \text{ taking } s = s_{\text{full}} \\ O\left(\frac{k^4}{\varepsilon^8 m^4 2^\ell}\right) & \text{otherwise}, \qquad \text{taking } s = s_{\text{part}} \end{cases}$$

  - *If $\frac{k^{2/3}}{\varepsilon^{4/3}} \le m \le \frac{k}{\varepsilon^2}$*

$$n = \begin{cases} O\left(\frac{k^2}{\varepsilon^4 m 2^\ell}\right) & \text{if } 1 \le 2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}} \text{ and } m \le \frac{k}{\varepsilon^2 2^{\ell/2}}, \text{ taking } s = s_{\text{part}} \\ O\left(\frac{k^2}{\varepsilon^4 2^{\ell/2}}\right) & \text{otherwise}, \qquad \text{taking } s = s_{\text{full}} \end{cases}$$

- *If $k\varepsilon^4 \le 1$ (in which case, $k < \frac{k^{2/3}}{\varepsilon^{4/3}} < \frac{\sqrt{k}}{\varepsilon^2} < \frac{k}{\varepsilon^2}$):*

$$n = \begin{cases} O\left(\frac{k^2}{\varepsilon^4 m 2^\ell}\right) & \text{if } \frac{\sqrt{k}}{\varepsilon^2} \le m \le \frac{k}{\varepsilon^2 2^{\ell/2}}, \text{ taking } s = s_{\text{part}} \\ O\left(\frac{k}{\varepsilon^2 2^{\ell/2}}\right) & \text{otherwise}, \qquad \text{taking } s = s_{\text{full}} \end{cases}$$

where s denotes the number of common random bits used by the protocol, $s_{\text{part}} = \log_2\big(m^2\varepsilon^4/k\big)$, and $s_{\text{full}} = \log_2\big(k/2^\ell\big)$.

The sample complexities are summarized in tables below.

For $k\varepsilon^4 > 1$ (in which case, $\frac{\sqrt{k}}{\varepsilon^2} < \frac{k^{2/3}}{\varepsilon^{4/3}} < k < \frac{k}{\varepsilon^2}$):

<table>
<tr><td></td><td colspan="2">$\frac{\sqrt{k}}{\varepsilon^2} \le m < \frac{k^{2/3}}{\varepsilon^{4/3}}$</td><td colspan="2">$\frac{k^{2/3}}{\varepsilon^{4/3}} \le m \le \frac{k}{\varepsilon^2}$</td></tr>
<tr><td rowspan="2">$1 \le 2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$</td><td colspan="2" rowspan="2">$n = \frac{k^4}{\varepsilon^8 m^4 2^\ell}$
$(s = s_{\text{part}})$</td><td>$\frac{k^{2/3}}{\varepsilon^{4/3}} \le m < \frac{k}{\varepsilon^2 2^{\ell/2}}$</td><td>$\frac{k}{\varepsilon^2 2^{\ell/2}} \le m \le \frac{k}{\varepsilon^2}$</td></tr>
<tr><td>$n = \frac{k^2}{\varepsilon^4 m 2^\ell}$
$(s = s_{\text{part}})$</td><td>$n = \frac{k}{\varepsilon^2 2^{\ell/2}}$
$(s = s_{\text{full}})$</td></tr>
<tr><td rowspan="2">$\frac{k^{2/3}}{\varepsilon^{4/3}} < 2^\ell \le k$</td><td>$\frac{\sqrt{k}}{\varepsilon^2} \le m < \frac{k}{\varepsilon^2 2^{\ell/2}}$</td><td>$\frac{k}{\varepsilon^2 2^{\ell/2}} \le m < \frac{k^{2/3}}{\varepsilon^{4/3}}$</td><td colspan="2" rowspan="2">$n = \frac{k}{\varepsilon^2 2^{\ell/2}}$
$(s = s_{\text{full}})$</td></tr>
<tr><td>$n = \frac{k^4}{\varepsilon^8 m^4 2^\ell}$
$(s = s_{\text{part}})$</td><td>$n = \frac{k^2}{\varepsilon^4 m^2}$
$(s = s_{\text{full}})$</td></tr>
</table>

For $k\varepsilon^4 \le 1$ (in which case, $k < \frac{k^{2/3}}{\varepsilon^{4/3}} < \frac{\sqrt{k}}{\varepsilon^2} < \frac{k}{\varepsilon^2}$):

	$\frac{\sqrt{k}}{\varepsilon^2} \le m < \frac{k}{\varepsilon^2 2^{\ell/2}}$	$\frac{k}{\varepsilon^2 2^{\ell/2}} \le m \le \frac{k}{\varepsilon^2}$
$1 \le 2^\ell \le k$	$n = \frac{k^2}{\varepsilon^4 m 2^\ell}$ $(s = s_{\text{part}})$	$n = \frac{k}{\varepsilon^2 2^{\ell/2}}$ $(s = s_{\text{full}})$

Theorem 8 will follow from the more general theorem (Theorem 9) below, which keeps the choice of the common random seed size, s, as a free parameter:

Theorem 9 *Let $s_0 \ge 0$ denotes the total number of common random bits available to all parties. The distributed closeness testing protocol specified by Algorithm 2 and Algorithm 3 is a valid protocol whenever $m = \Omega(\sqrt{k}/\varepsilon^2)$, involving the free parameter*

$$0 \le s \le \min\left\{ s_0,\ (\log_2 k) - \ell,\ \log_2 \frac{m^2\varepsilon^4}{k} \right\}$$

which denotes the number of common random bits used by the protocol. Its sample complexity is given by

$$n = \begin{cases} O\left(\frac{k^{3/2}}{\varepsilon^2 2^\ell 2^{s/2}}\left(\sqrt{\frac{k}{m2^s}} \vee 1\right)\right) & \text{when } m \ge \frac{k^{2/3}}{\varepsilon^{4/3}}\,; \\ O\left(\frac{k^3}{\varepsilon^4 m^2 2^{\ell+s}}\right) & \text{when } m < \frac{k^{2/3}}{\varepsilon^{4/3}}\,. \end{cases}$$

Note that the second case can only happen when $k\varepsilon^4 \ge 1$, as otherwise $k^{2/3}/\varepsilon^{4/3} \le \sqrt{k}/\varepsilon^2 \le m$ (ignoring constant factors).

The following tables note the optimal values of s (given m, ℓ, s_0) that minimizes n in our protocol.

When $k\varepsilon^4 > 1$ ***and*** $m \geq k^{2/3}/\varepsilon^{4/3}$***:***

	$1 \leq 2^\ell \leq \frac{1}{\varepsilon^4}$	$\frac{1}{\varepsilon^4} < 2^\ell \leq \frac{k^{2/3}}{\varepsilon^{4/3}}$	$\frac{k^{2/3}}{\varepsilon^{4/3}} < 2^\ell \leq k$
$1 \leq 2^{s_0} \leq (k\varepsilon^4)^{1/3}$	s_0	s_0	$s_0 \wedge \log_2 \frac{k}{2^\ell}$
$(k\varepsilon^4)^{1/3} \leq 2^{s_0} \leq k\varepsilon^4$	$\begin{cases} s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq k \\ s_0 & \text{if } m > k \end{cases}$	$\begin{cases} s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}} \\ s_0 \wedge \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$	$\log_2 \frac{k}{2^\ell}$
$2^{s_0} > k\varepsilon^4$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq k \\ s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } k < m \leq \frac{k}{\varepsilon^2 2^{\ell/2}} \\ s_0 \wedge \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$	$\log_2 \frac{k}{2^\ell}$

When $k\varepsilon^4 > 1$ ***and*** $m < k^{2/3}/\varepsilon^{4/3}$***:***

	$1 \leq 2^\ell \leq \frac{1}{\varepsilon^4}$	$\frac{1}{\varepsilon^4} < 2^\ell \leq \frac{k^{2/3}}{\varepsilon^{4/3}}$
$1 \leq 2^{s_0} \leq (k\varepsilon^4)^{1/3}$	$s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k}$	$\begin{cases} s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}} \\ s_0 \wedge \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$
$(k\varepsilon^4)^{1/3} \leq 2^{s_0} \leq k\varepsilon^4$	$\log_2 \frac{m^2\varepsilon^4}{k}$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$
$2^{s_0} > k\varepsilon^4$	$\log_2 \frac{m^2\varepsilon^4}{k}$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$

When $k\varepsilon^4 \leq 1$***:***

$$s = \begin{cases} s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ s_0 \wedge \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}}. \end{cases}$$

Note that Theorem 8 immediately follows from Theorem 9 by setting the "randomness budget" $s_0 = \infty$.

Proof of Theorem 9

Recall that our overarching goal is to bound n (number of players required), given m (number of samples with the center), ℓ (communication budget per player) and s_0 (number of common random bits available).

Constraints on use of common randomness. Suppose the algorithm uses s bits of common randomness for domain compression. Let the induced distribution on the new domain be p' (corresponding to distribution p of players) and q' (corresponding to distribution q of center). Center then outputs its verdict based on testing closeness between p' and q' with parameters $k' = k/2^s$ and $\varepsilon' = \varepsilon/\sqrt{2^s}$. This requires (information theoretically) at least $\frac{\sqrt{k'}}{\varepsilon'^2}$ samples from both p' and q'. In particular, this implies that $m \geq \sqrt{k2^s}/\varepsilon^2$, or, equivalently,

$$2^s \leq \frac{m^2\varepsilon^4}{k}. \tag{1}$$

Further, since each player has a communication budget of ℓ bits, we do not compress domain to size smaller than 2^ℓ – i.e., $k' \geq 2^\ell$, or

$$2^s \leq \frac{k}{2^\ell}. \tag{2}$$

Finally, since the number of available common random bits is at most s_0, we have

$$2^s \leq 2^{s_0}. \tag{3}$$

Overall, constraints on the number of random bits s can thus be encapsulated as

$$1 \leq 2^s \leq \min\left\{\frac{m^2\varepsilon^4}{k}, \frac{k}{2^\ell}, 2^{s_0}\right\}. \tag{4}$$

A.1. Proof of correctness.

The correctness of our protocol follows from the correctness, with high (constant) probability, of the four main subroutines – DOMAIN COMPRESSION (see Theorem 6 and the text immediately after it), DISTRSIM$_\ell$ (see Theorem 5), DOMAIN EXPANSION (see Theorem 4), and ℓ_2-CLOSENESS-TESTER (see Theorem 3) – and taking a union bound (in order to upper bound the total probability of failure by a constant strictly less than $1/2$). Given any target probability of success $1-\delta$, $\delta \in (0,1]$, we can simply run our protocol $O(\log(1/\delta))$ times and take the majority of the outcomes.

A.2. Sample complexity analysis assuming $k\varepsilon^4 > 1$

Before delving into a case-wise analysis, we note that, since $k' = k/2^s$ and $\varepsilon' = \varepsilon/\sqrt{2^s}$, it always holds that

$$\frac{k'}{\varepsilon'^2} = \frac{k}{\varepsilon^2} \tag{5}$$

which, in particular, implies that

$$\frac{k'^{2/3}}{\varepsilon'^{4/3}} = \frac{k^{2/3}}{\varepsilon^{4/3}}.$$

Further, when $k\varepsilon^4 > 1$, we have

$$\frac{\sqrt{k}}{\varepsilon^2} \leq \frac{k^{2/3}}{\varepsilon^{4/3}} \leq k \leq \frac{k}{\varepsilon^2}. \tag{6}$$

Finally, after domain compression, the ordering of important quantities plays out as follows depending on the number of random bits s used.

If $1 \leq 2^s \leq (k\varepsilon^4)^{1/3}$:

$$\frac{\sqrt{k}}{\varepsilon^2} \leq \frac{\sqrt{k'}}{\varepsilon'^2} \leq \frac{k'^{2/3}}{\varepsilon'^{4/3}} \leq k' \leq k. \tag{7}$$

If $(k\varepsilon^4)^{1/3} \leq 2^s \leq k\varepsilon^4$:

$$k' \leq \frac{k'^{2/3}}{\varepsilon'^{4/3}} \leq \frac{\sqrt{k'}}{\varepsilon'^2} \leq k. \tag{8}$$

If $2^s \geq k\varepsilon^4$:

$$k' \leq \frac{k'^{2/3}}{\varepsilon'^{4/3}} \leq k \leq \frac{\sqrt{k'}}{\varepsilon'^2}. \tag{9}$$

With regard to setting s satisfying the constraint in Eq. (4), the following will come in handy:

$$\min\left\{\frac{m^2\varepsilon^4}{k}, \frac{k}{2^\ell}\right\} = \begin{cases} \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases} \tag{10}$$

and

$$1 \le 2^\ell \le \frac{1}{\varepsilon^4} \implies (k\varepsilon^4)^{1/3} \le \frac{k}{2^\ell}, \quad k\varepsilon^4 \le \frac{k}{2^\ell}; \tag{11}$$

$$\frac{1}{\varepsilon^4} \le 2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}} \implies (k\varepsilon^4)^{1/3} \le \frac{k}{2^\ell}, \quad k\varepsilon^4 \ge \frac{k}{2^\ell}; \tag{12}$$

$$2^\ell \ge \frac{k^{2/3}}{\varepsilon^{4/3}} \implies (k\varepsilon^4)^{1/3} \ge \frac{k}{2^\ell}, \quad k\varepsilon^4 \ge \frac{k}{2^\ell}. \tag{13}$$

A.2.1. CASE 1: $m \ge k^{2/3}/\varepsilon^{4/3}$.

By (5), we have $m \ge k'^{2/3}/\varepsilon'^{4/3}$. Thus, according to Algorithm 3, center does domain expansion using $\mathrm{Poi}(\bar{m})$ samples from the multiset available with it, where $\bar{m} \asymp \min\{m, k'\}$. Furthermore, for closeness testing, center obtains $n' \asymp k'/(\sqrt{\bar{m}}\varepsilon'^2)$ samples using DISTSIM with $n = n'\big(k'/2^\ell\big)$ players. Therefore,

$$n \asymp \frac{n'k'}{2^\ell} = \frac{k'^2}{\varepsilon'^2 2^\ell \sqrt{\bar{m}}} = \frac{k'^2}{\varepsilon'^2 2^\ell \sqrt{m \wedge k'}} = \frac{k^2}{\varepsilon^2 2^\ell 2^s \sqrt{m \wedge (k/2^s)}} = \frac{k^{3/2}}{\varepsilon^2 2^\ell 2^{s/2}} \left(\sqrt{\frac{k}{m2^s}} \vee 1\right). \tag{14}$$

This, in particular, tells us that, we should set s to be the maximum possible while satisfying Eq. (4). We now see what is the best possible value of s for different ranges that s can lie in.

Case 1.1: $1 \le 2^s \le (k\varepsilon^4)^{1/3}$**.**

Suppose $2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$. This implies that

$$(k\varepsilon^4)^{1/3} \le \frac{k}{2^\ell}.$$

Also, since $m \ge k^{2/3}/\varepsilon^{4/3}$, we have

$$(k\varepsilon^4)^{1/3} \le \frac{m^2\varepsilon^4}{k}.$$

Thus, s should be set to

$$s = \frac{1}{3}\log_2 k\varepsilon^4.$$

Suppose $2^\ell > \frac{k^{2/3}}{\varepsilon^{4/3}}$. Then

$$\frac{k}{2^\ell} \le (k\varepsilon^4)^{1/3} \le \frac{m^2\varepsilon^4}{k}.$$

Thus, s should be set to

$$s = \log_2 \frac{k}{2^\ell}.$$

Case 1.2: $(k\varepsilon^4)^{1/3} \le 2^s \le k\varepsilon^4$**.** For $2^s \ge (k\varepsilon^4)^{1/3}$ (see Eqs. (8) and (9)), we have that $k' = k/2^s \le m$, and thus (from Eq. (14)), we have

$$n \asymp \frac{k^{3/2}}{\varepsilon^2 2^\ell 2^{s/2}} \quad \text{since } k' \le m.$$

We now analyze how to choose s when $(k\varepsilon^4)^{1/3} \le 2^s \le k\varepsilon^4$.
Suppose $2^\ell \le \frac{1}{\varepsilon^4}$. Then

$$k\varepsilon^4 \le \frac{k}{2^\ell}.$$

Moreover,

$$(k\varepsilon^4)^{1/3} \le \frac{m^2\varepsilon^4}{k} \le k\varepsilon^4 \quad \text{if } m \le k, \tag{15}$$

$$(k\varepsilon^4)^{1/3} \le k\varepsilon^4 \le \frac{m^2\varepsilon^4}{k} \quad \text{if } m > k. \tag{16}$$

Thus, s should be set to

$$s = \begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le k, \\ \log_2 k\varepsilon^4 & \text{if } m > k. \end{cases}$$

Suppose $\frac{1}{\varepsilon^4} < 2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$. This implies that

$$\frac{k}{\varepsilon^2 2^{\ell/2}} < k.$$

We have

$$(k\varepsilon^4)^{1/3} \le \frac{m^2\varepsilon^4}{k} \le \frac{k}{2^\ell} \le k\varepsilon^4 \quad \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \tag{17}$$

$$(k\varepsilon^4)^{1/3} \le \frac{k}{2^\ell} \le \frac{m^2\varepsilon^4}{k} \le k\varepsilon^4 \quad \text{if } \frac{k}{\varepsilon^2 2^{\ell/2}} < m \le k \tag{18}$$

$$(k\varepsilon^4)^{1/3} \le \frac{k}{2^\ell} \le k\varepsilon^4 \le \frac{m^2\varepsilon^4}{k} \quad \text{if } m > k. \tag{19}$$

Thus, s should be set to

$$s = \begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}}. \end{cases}$$

Suppose $2^\ell > \frac{k^{2/3}}{\varepsilon^{4/3}}$. Then

$$\frac{k}{2^\ell} < (k\varepsilon^4)^{1/3}.$$

Thus, s has no solution in this range! (That is, we never need to set s in this range when $2^\ell > \frac{k^{2/3}}{\varepsilon^{4/3}}$.)

Case 1.3: $2^s > k\varepsilon^4$**.** As in the previous case, the expression for n is

$$n \asymp \frac{k^{3/2}}{\varepsilon^2 2^\ell 2^{s/2}} \quad \text{since } k' \le m.$$

If $m < k$, then

$$\frac{m^2\varepsilon^4}{k} < k\varepsilon^4$$

and thus s has no solution in this range.
Suppose $2^\ell \le \frac{1}{\varepsilon^4}$. Then

$$k\varepsilon^4 \le \frac{k}{2^\ell}$$

and

$$\frac{k}{\varepsilon^2 2^{\ell/2}} \ge k.$$

So, we have

$$k\varepsilon^4 \le \frac{m^2\varepsilon^4}{k} \le \frac{k}{2^\ell} \qquad \text{if } k < m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \tag{20}$$

$$k\varepsilon^4 \le \frac{k}{2^\ell} \le \frac{m^2\varepsilon^4}{k} \qquad \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}}. \tag{21}$$

Thus, s should be set to

$$s = \begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } k < m \le \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$$

and s has no solution if $m \le k$.
Suppose $2^\ell > \frac{1}{\varepsilon^4}$. Then

$$\frac{k}{2^\ell} < k\varepsilon^4.$$

Thus, s has no solution in this range!

Summary of the case when $m \ge k^{2/3}/\varepsilon^{4/3}$**.** The following table summarizes the values of optimal s in different ranges

	$1 \le 2^\ell \le \frac{1}{\varepsilon^4}$	$\frac{1}{\varepsilon^4} < 2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$	$\frac{k^{2/3}}{\varepsilon^{4/3}} < 2^\ell \le k$
$1 \le 2^s \le (k\varepsilon^4)^{1/3}$	$\frac{1}{3}\log_2 k\varepsilon^4$	$\frac{1}{3}\log_2 k\varepsilon^4$	$\log_2 \frac{k}{2^\ell}$
$(k\varepsilon^4)^{1/3} \le 2^s \le k\varepsilon^4$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le k \\ \log_2 k\varepsilon^4 & \text{if } m > k \end{cases}$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$	-
$2^s > k\varepsilon^4$	$\begin{cases} - & \text{if } m \le k \\ \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } k < m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$	-	-

Now, we add the constraint that the parties may have a randomness budget of at most s_0 bits. The following table summarizes the optimal value of s that the protocol should use:

	$1 \le 2^\ell \le \frac{1}{\varepsilon^4}$	$\frac{1}{\varepsilon^4} < 2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$	$\frac{k^{2/3}}{\varepsilon^{4/3}} < 2^\ell \le k$
$1 \le 2^{s_0} \le (k\varepsilon^4)^{1/3}$	s_0	s_0	$s_0 \wedge \log_2 \frac{k}{2^\ell}$
$(k\varepsilon^4)^{1/3} \le 2^{s_0} \le k\varepsilon^4$	$\begin{cases} s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le k \\ s_0 & \text{if } m > k \end{cases}$	$\begin{cases} s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ s_0 \wedge \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$	$\log_2 \frac{k}{2^\ell}$
$2^{s_0} > k\varepsilon^4$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le k \\ s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } k < m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ s_0 \wedge \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$	$\log_2 \frac{k}{2^\ell}$

The sample complexity is obtained by plugging-in the appropriate value of s (depending on s_0, ℓ, m) from the above table to the expression given by Eq. (14).

A.2.2. CASE 2: $m < k^{2/3}/\varepsilon^{4/3}$.

In this case, center does domain expansion using $\mathrm{Poi}(\bar{n})$ samples from p' that it obtains using DISTSIM with $n_1 = \bar{n}\big(k'/2^\ell\big)$ players. Here, $\bar{n}$ is chosen so that ℓ_2 closeness testing done after domain expansion requires m samples each from $\bar{p}$ and $\bar{q}$. As in the analysis of the unconstrained setting, this is ensured by setting $\bar{n} \asymp k'^2/(m^2\varepsilon'^4)$. Center obtains m samples each from $\bar{p}$ using DISTSIM with $n_2 = m\big(k'/2^\ell\big)$ players. Thus

$$n = n_1 + n_2 = (\bar{n} + m)\frac{k'}{2^\ell} = \left(\frac{k'^2}{m^2\varepsilon'^4} + m\right)\frac{k'}{2^\ell} \asymp \frac{k'^2}{m^2\varepsilon'^4}\frac{k'}{2^\ell} = \frac{k^3}{\varepsilon^4 2^{\ell+s} m^2} \tag{22}$$

Case 2.1: $1 \le 2^s \le (k\varepsilon^4)^{1/3}$**.**

Suppose $2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$. Then

$$(k\varepsilon^4)^{1/3} \le \frac{k}{2^\ell}.$$

Since $m < k^{2/3}/\varepsilon^{4/3}$, we have

$$\frac{m^2\varepsilon^4}{k} \le (k\varepsilon^4)^{1/3}.$$

Thus, s should be set to

$$s = \log_2 \frac{m^2\varepsilon^4}{k}.$$

Suppose $2^\ell > \frac{k^{2/3}}{\varepsilon^{4/3}}$. Then

$$\frac{k}{2^\ell} \le (k\varepsilon^4)^{1/3}.$$

We still have

$$\frac{m^2\varepsilon^4}{k} \le (k\varepsilon^4)^{1/3}.$$

Eq. (10) tells us that

$$\min\left\{\frac{m^2\varepsilon^4}{k}, \frac{k}{2^\ell}\right\} = \begin{cases} \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}}. \end{cases}$$

Thus, s should be set to

$$s = \begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}}. \end{cases}$$

Case 2.2: $(k\varepsilon^4)^{1/3} \le 2^s \le k\varepsilon^4$**.**

Suppose $2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$. Then

$$\frac{m^2\varepsilon^4}{k} \le (k\varepsilon^4)^{1/3}.$$

Thus, s has no solution in this range!
Suppose $2^\ell > \frac{k^{2/3}}{\varepsilon^{4/3}}$. Then

$$\frac{k}{2^\ell} \le (k\varepsilon^4)^{1/3}.$$

Thus, again, s has no solution in this range!

Case 2.3: $2^{s_0} > k\varepsilon^4$**.** From Case 2.2 above and the fact that $k\varepsilon^4 \ge (k\varepsilon^4)^{1/3}$, we get that s has no solution in this range!

Summary of the case when $m < k^{2/3}/\varepsilon^{4/3}$**.** The following table summarizes the values of optimal s in different ranges

	$1 \le 2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$	$\frac{k^{2/3}}{\varepsilon^{4/3}} < 2^\ell \le k$
$1 \le 2^s \le (k\varepsilon^4)^{1/3}$	$\log_2 \frac{m^2\varepsilon^4}{k}$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$
$(k\varepsilon^4)^{1/3} \le 2^s \le k\varepsilon^4$	-	-
$2^s > k\varepsilon^4$	-	-

Now, we add the constraint that the parties may have a randomness budget of at most s_0 bits. The following table summarizes the optimal value of s that the protocol should use:

	$1 \le 2^\ell \le \frac{1}{\varepsilon^4}$	$\frac{1}{\varepsilon^4} < 2^\ell \le \frac{k^{2/3}}{\varepsilon^{4/3}}$
$1 \le 2^{s_0} \le (k\varepsilon^4)^{1/3}$	$s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k}$	$\begin{cases} s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ s_0 \wedge \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$
$(k\varepsilon^4)^{1/3} \le 2^{s_0} \le k\varepsilon^4$	$\log_2 \frac{m^2\varepsilon^4}{k}$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$
$2^{s_0} > k\varepsilon^4$	$\log_2 \frac{m^2\varepsilon^4}{k}$	$\begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \le \frac{k}{\varepsilon^2 2^{\ell/2}} \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$

The sample complexity is obtained by plugging-in the appropriate value of s (depending on s_0, ℓ, m) from the above table to the expression given by Eq. (22).

A.3. Sample complexity analysis assuming $k\varepsilon^4 \leq 1$

When $k\varepsilon^4 \leq 1$, we have

$$k' < k < \frac{k^{2/3}}{\varepsilon^{4/3}} < \frac{\sqrt{k}}{\varepsilon^2} < \frac{\sqrt{k'}}{\varepsilon'^2} < \frac{k}{\varepsilon^2}$$

which implies that the information theoretic lower bound of $\sqrt{k'}/\varepsilon'^2$ on closeness testing becomes a very strong condition. In particular, for successful testing, it must be the case that $m \geq k^{2/3}/\varepsilon^{4/3}$ $(= k'^{2/3}/\varepsilon'^{4/3})$. Thus, according to Algorithm 3, center does domain expansion using $\mathrm{Poi}(\bar{m})$ samples from the multiset available with it, where $\bar{m} \asymp \min\{m, k'\} = k'$. Furthermore, for closeness testing, center obtains $n' \asymp k'/(\sqrt{\bar{m}}\varepsilon'^2) = \sqrt{k'}/\varepsilon^2$ samples using DISTSIM with $n = n'\big(k'/2^\ell\big)$ players. Therefore,

$$n \asymp \frac{n'k'}{2^\ell} = \frac{k'^{3/2}}{\varepsilon'^2 2^\ell} = \frac{k^{3/2}}{\varepsilon^2 2^\ell 2^{s/2}}. \tag{23}$$

Note that, since $k\varepsilon^4 \leq 1$, we must have that

$$2^s \geq k\varepsilon^4$$

and so we do not have multiple sub-cases on s that we had when $k\varepsilon^4 > 1$. The only consideration with regard to setting s would be Eq. (10), which says that

$$\min\left\{\frac{m^2\varepsilon^4}{k}, \frac{k}{2^\ell}\right\} = \begin{cases} \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}} \end{cases}$$

and thus, s should be set to

$$s = \begin{cases} \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}}. \end{cases}$$

When we add the constraint that the parties may have a randomness budget of at most s_0 bits, the protocol should use s given by

$$s = \begin{cases} s_0 \wedge \log_2 \frac{m^2\varepsilon^4}{k} & \text{if } m \leq \frac{k}{\varepsilon^2 2^{\ell/2}}, \\ s_0 \wedge \log_2 \frac{k}{2^\ell} & \text{if } m > \frac{k}{\varepsilon^2 2^{\ell/2}}. \end{cases}$$