Support Testing in the Huge Object Model

Tomer Adar* Eldar Fischer† Amit Levi‡

July 9, 2024

Abstract

The Huge Object model is a distribution testing model in which we are given access to independent samples from an unknown distribution over the set of strings $\{0,1\}^n$, but are only allowed to query a few bits from the samples. We investigate the problem of testing whether a distribution is supported on m elements in this model. It turns out that the behavior of this property is surprisingly intricate, especially when also considering the question of adaptivity.

We prove lower and upper bounds for both adaptive and non-adaptive algorithms in the one-sided and two-sided error regime. Our bounds are tight when m is fixed to a constant (and the distance parameter ε is the only variable). For the general case, our bounds are at most $O(\log m)$ apart. In particular, our results show a surprising $O(\log \varepsilon^{-1})$ gap between the number of queries required for non-adaptive testing as compared to adaptive testing. For one sided error testing, we also show that a $O(\log m)$ gap between the number of samples and the number of queries is necessary. Our results utilize a wide variety of combinatorial and probabilistic methods.

*Technion - Israel Institute of Technology, Israel. Email: tomer-adar@campus.technion.ac.il.

†Technion - Israel Institute of Technology, Israel. Email: eldar@cs.technion.ac.il. Research supported by an Israel Science Foundation grant number 879/22.

‡University of Haifa and Technion - Israel Institute of Technology, Israel. Email: alevi@ds.haifa.ac.il.

1 Introduction

Property testing [RS96, GGR98] is a framework concerned with analyzing global properties of an input while reading only a small part thereof, in the form of queries. Over the past few decades property testing has become an active field of study in theoretical computer science (see e.g, [Gol17]). The study of distribution property testing was first implicitly explored in [GR11], and explicitly formulated in [BFF+01] and [BFR+00]. In the standard model of distribution testing, an algorithm can access a sequence of independent sampled elements drawn from an unknown input distribution μ, and it either accepts or rejects the input based on this sequence. An ε-testing algorithm for a property of distributions is required to accept every input distribution that satisfies the property with high probability (e.g., $\frac{2}{3}$), and to reject with high probability (e.g., $\frac{2}{3}$) every input distribution whose variation distance from every distribution satisfying the property is greater than ε.

The standard model of distribution testing assumes that the elements drawn from the distribution are fully accessible, which might be unreasonable if they are "huge". The Huge Object model, whose study was initiated in [GR22], treats the sampled elements as long strings that have to be queried. In this model, for example, it is possible that the algorithm has two non-identical samples without being able to distinguish them. This "two-phase" characteristic of the Huge Object model ("sample then query", rather than only taking samples or only querying a string) exhibits rich behavior with respect to adaptive querying, as studied in detail in [AF23].

In the standard model of distribution testing, [VV11] and [VV17] show a tight bound of $\Theta(m/\log m)$ samples for two-sided error ε-testing of having a support size bounded by m in the standard model, for every fixed ε. An upper bound of $O(\varepsilon^{-1}m)$ samples for one-sided algorithms is implicitly shown in [AF23], and here we show that it is tight (Proposition 4.6). Based on these tight bounds, the bounded support property is considered to be fully understood in the standard model for one-sided testing, and mostly understood in the two-sided case (for every fixed m there is still a gap between $\Omega(\varepsilon^{-1})$ and $O(\varepsilon^{-2})$ for two-sided testing).

One would expect that having bounded support, which is arguably the simplest of distribution properties, would have simple and easily understood testing bounds also in the Huge Object model. As in the standard model, it is the only label-invariant property that is testable using one-sided error algorithms (Proposition 5.11). However, it turns out that the behaviour of this property under the Huge Object model is surprisingly intricate. One unexpected feature that we show here is a gap between the number of queries required for non-adaptively testing for this property as compared to adaptive testing. Indeed there is no adaptivity in the standard distribution testing model, but one would not expect the label-invariant (and even mapping-invariant as per the definition in [GR22]) property of having bounded support to exhibit such a gap.

1.1 Definition of the model

The Huge Object model differs from the standard sampling model in its distance measure and in the way that the algorithm gathers information about the input distribution.

Algorithmic model

A probabilistic algorithm $\mathcal{A}$ with q queries and s samples, whose input is a distribution P over $\{0,1\}^n$ accessible via the Huge Object model, is an algorithm that acts in the following manner:

at every stage, the algorithm may ask for a new sample v that is provided by drawing it according to P, independently of all prior samples, or may ask to query a coordinate $j \in \{1, \ldots, n\}$ of an old sample u (the algorithm may use internal coin tosses to make its decisions). When this query is made, the algorithm is provided with $u_j \in \{0, 1\}$ as its answer. The algorithm has no access to the sampled vectors apart from queries. At the end, after taking not more than a total of s samples and making a total of not more than q queries, the algorithm provides its output.

We say that the algorithm is *non-adaptive* if it makes all its sampling and querying decisions in advance, prior to receiving all query answers in bulk. Only the final output of a non-adaptive algorithm may depend on the received answers. The formal definition of adaptivity appears in Subsection 5.2.

Distances

Here we define some measures of distance. Note that we usually use $d(\cdot, \cdot)$ without mentioning the measure that we use, if its context is unambiguous. For distributions over $\{0, 1\}^n$, $d(\cdot, \cdot)$ usually refers to the earth mover's distance.

Definition 1.1 (String distance)**.** Let $u, v \in \{0, 1\}^n$ be two strings. We define their *distance* as the normalized Hamming distance,

$$d_H(u, v) = \frac{1}{n} \left|\{1 \leq i \leq n \mid u_i \neq v_i\}\right| = \Pr_{i \sim \{1, \ldots, n\}}[u_i \neq v_i]$$

We define the distance of $u \in \{0, 1\}^n$ from a set $A \subseteq \{0, 1\}^n$ as $d_H(u, A) = \min_{v \in A} d_H(u, v)$.

Definition 1.2 (Transfer distribution)**.** Let P and Q be distributions over finite sets Ω_1 and Ω_2, respectively. A distribution T over $\Omega_1 \times \Omega_2$ is a *transfer distribution* from P to Q if for every $a \in \Omega_1$, $\Pr_{(u,v) \sim T}[u = a] = P(a)$, and for every $b \in \Omega_2$, $\Pr_{(u,v) \sim T}[v = b] = Q(b)$. The set of transfer distributions from P to Q is denoted by $\mathcal{T}(P, Q)$. Note that this is a compact set when considered as a set of real-valued vectors.

Definition 1.3 (Variation distance)**.** Let μ and ν be two distributions over a finite set Ω. Their *variation distance* is defined as:

$$d_{\mathrm{var}}(\mu, \nu) = \frac{1}{2} \sum_{u \in \Omega} |\mu(u) - \nu(u)| = \max_{E \subseteq \Omega} \left| \Pr_{\mu}[E] - \Pr_{\nu}[E] \right| = \min_{T \in \mathcal{T}(\mu, \nu)} \Pr_{(u,v) \sim T}[u \neq v]$$

Definition 1.4 (Earth mover's distance)**.** Let P and Q be two distributions over $\{0, 1\}^n$. Their *earth mover's distance* is defined as:

$$d_{\mathrm{EMD}}(P, Q) = \min_{T \in \mathcal{T}(P, Q)} \mathrm{E}_{(u,v) \sim T}[d_H(u, v)]$$

The above minimum exists since it is in particular the minimum of a continuous function over a compact set.

Testing model

Definition 1.5 (A property)**.** A *property* $\mathcal{P}$ is a sequence $\mathcal{P}_1, \mathcal{P}_2, \ldots$ such that for every $n \geq 1$, $\mathcal{P}_n$ is a compact subset of the set of all distributions over $\{0, 1\}^n$.

Definition 1.6 (Distance of a distribution from a property)**.** Let $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, \ldots)$ be a property and P be a distribution over $\{0,1\}^n$ for some n. The *distance of P from $\mathcal{P}$* is defined as $d_{\mathrm{EMD}}(P, \mathcal{P}) = \min_{Q \in \mathcal{P}_n}\{d_{\mathrm{EMD}}(P, Q)\}$.

Definition 1.7 (ε-test)**.** Let $\mathcal{P}$ be a property of distributions over $\{0,1\}^n$. We say that a probabilistic algorithm $\mathcal{A}$ is an *ε-test* for $\mathcal{P}$ if:

- For every $P \in \mathcal{P}$, $\mathcal{A}$ accepts with probability higher than $\frac{2}{3}$.
- For every probability distribution P over $\{0,1\}^n$ that is ε-far from $\mathcal{P}$ (satisfying $d(P, \mathcal{P}) > \varepsilon$), $\mathcal{A}$ rejects with probability higher than $\frac{2}{3}$

Definition 1.8 (one-sided and two-sided ε-test)**.** Consider the setting of the above definition. If additionally for every input $P \in \mathcal{P}$, $\mathcal{A}$ accepts $\mathcal{P}$ with probability 1 (rather than "higher than $\frac{2}{3}$"), then we say that $\mathcal{A}$ is a *one-sided ε-test* for $\mathcal{P}$. Otherwise, we say that $\mathcal{A}$ has *two-sided error*.

1.2 Summary of our results

Table of results The following is a table summarizing the bounds presented here for ε-testing for being supported by at most m elements, along with previously known ones provided for reference (Section 4 contains a sketch on how to derive them). The hidden coefficients in the $O(\cdot)$ and the $\Omega(\cdot)$ notations are global numerical constants. The new results appear in purple.

Model	One-sided Error	Two-sided Error
Standard model (Sample complexity)	$\Theta(\varepsilon^{-1}m)$ Folklore, see [AF23]	$\Omega(\varepsilon^{-1}m/\log m)$ [VV11] $O(\varepsilon^{-2}m/\log m)$ [VV17]
Huge Object Non-adaptive	$\Omega(\varepsilon^{-1}m(\log\varepsilon^{-1} + \log m))$ $O(\varepsilon^{-1}m\log\varepsilon^{-1}\log m)$	$\Omega(\varepsilon^{-1}\log\varepsilon^{-1})$ $O(\varepsilon^{-3}m\log\varepsilon^{-1})$ [VV17] + [GR22]
Huge Object Adaptive	$\Omega(\varepsilon^{-1}m\log m)$ $O(\varepsilon^{-1}m\log m \cdot \min\{\log\varepsilon^{-1}, \log m\})$	$\Omega(\varepsilon^{-1}m/\log m)$ [VV11]

The overview in Section 3 provides an informal guide on deriving our results, whose proofs appear in the sections following it.

The following are some conclusions to be drawn from the bounds given above. We use $\mathcal{S}_m$ to denote the property of being supported by at most m elements (formally defined in Definition 5.4).

Adaptive vs. non-adaptive two-sided asymptotic gap The most surprising result is that testing a distribution for being supported by at most two elements cannot be done using a number of queries linear in ε^{-1}, even with two-sided error. This result applies for every $m \geq 2$, and the exact bound is $\Omega(\varepsilon^{-1}\log\varepsilon^{-1})$ (with the implicit coefficient being independent of m). To the best of our knowledge, combined with the $O(\varepsilon^{-1})$ adaptive upper bound of [AF23], "being supported by at most two elements" is the first explicit example of a property that is closed under mapping (and in particular is label-invariant) which has different asymptotic bounds for the number of queries for adaptive algorithms and non-adaptive ones in the Huge Object model (see Theorem 6.1).

A possible explanation for this is that being label-invariant in the Huge Object model is different from being so in the standard model, because applying a permutation on the labels may change their distinguishability, and in particular it may change the distance from the property.

In this paper we provide a thorough investigation of $\mathcal{S}_m$ utilizing a variety of methods. In particular, we show several gaps such as the above mentioned one. However, the behaviour of the bounded support property in the Huge Object model, especially when considering it as a problem with two variables (namely the maximal support sized m and the distance parameter ε) is still not completely understood. We do have tight bounds for the fixed constant m cases (where only ε is variable) for all algorithm types, and bounds up to logarithmic factors for the more general cases.

One-sided bounds and a gap from the standard model We have tight bounds for ε-testing of $\mathcal{S}_m$ for every fixed m (and variable ε) for both non-adaptive algorithms and adaptive ones. These bounds are also tight for every fixed ε (and variable m). Additionally, our bounds show a gap between the standard model (considering sample complexity) and the Huge Object model (considering query complexity). Consider the bounded support property as a sequence of individual properties, where for every $m \geq 2$, the m-th property is $\mathcal{S}_m$. We show that, if we only allow one-sided error tests, there is an $O(\log m)$ gap between the standard model of distribution testing and the Huge Object model. In the standard model, there exists a one-sided test for $\mathcal{S}_m$ at the cost of $O(\varepsilon^{-1}m)$ samples. In the Huge Object model, there is a lower bound of $\Omega(\varepsilon^{-1}m \cdot \log m)$ many queries for every one-sided ε-test, even if it is adaptive. Note that the gap is between the number of *samples* in the standard model and the number of *queries* in the Huge Object model, which is the natural measure of complexity in this model.

New tools

A new algorithmic paradigm For the adaptive one-sided upper bound, we define a standalone algorithmic primitive, the "fishing expedition" paradigm, that repeatedly executes a subroutine until it reaches a predefined goal or when it finds out that it is no longer cost-effective (even if it did not reach the goal). We believe that this primitive will also be useful in future endeavors.

A hybrid probabilistic-extremal analysis We define a concept of "valid composition". Loosely speaking, it is an ordered subset of samples that become closer to each other as the sequence progresses, but are still ε-far from each other. We use a hybrid probabilistic-extremal argument to show that for an input distribution that is ε-far from m-support, with high probability, there exists a valid composition with at least $m+1$-elements.

The hybrid probabilistic-extremal argument works as follows: we define some rank of valid compositions. If for every individual valid composition with at most m elements, there is a high probability that it is not maximal (according to the rank), then globally there is a high probability that none of them is maximal. Hence, with high probability, the maximally-ranked valid composition within our samples must have at least $m+1$ elements.

A new use for an old combinatorial result For the adaptive one-sided lower bound, we use an old combinatorial result, that a biclique covering of the m-clique must have at least $m \log_2 m$ vertices [Han64, KS67], to show that the every witness against m-support is at least $m \log m$ bits long, which makes it a lower bound to the number of queries. To apply a multiplicative factor of ε^{-1}, which is pretty easy for non-adaptive algorithms, we analyze the effectivity of a decision tree that incrementally constructs a witness based on the queries.

1.3 Open problems

One-sided non-adaptive bounds We have an $\Omega(\varepsilon^{-1}m(\log\varepsilon^{-1}+\log m))$ lower bound for one-sided ε-testing of $\mathcal{S}_m$, as well as an $O(\varepsilon^{-1}m\log\varepsilon^{-1}\log m)$ upper bound for one-sided ε-testing of $\mathcal{S}_m$. We believe that the upper bound is tight, but we do not have the corresponding lower bound. What is the true complexity of ε-testing $\mathcal{S}_m$?

Non-trivial two-sided bounds Is there a lower bound of $\omega(m/\log m)$ queries for two-sided testing of $\mathcal{S}_m$ (noting that [VV11] only gives $\Omega(m/\log m)$), even for non-adaptive algorithms? We believe that $\Omega(m)$ should be this lower bound, based on the $\log m$-gap in the one-sided case (a $\Theta(m)$ tight bound in the standard model, and a $\Theta(m\log m)$ tight bound in the Huge Object model).

One-sided adaptive bounds Our results for one-sided adaptive ε-testing of $\mathcal{S}_m$ are tight with respect to m, but have a logarithmic gap with respect to ε (more precisely, with respect to $\min\{\varepsilon^{-1},m\}$). Closing this gap is an open problem.

The tradeoffs between sample and query complexity Our bounds apply to the query complexity of the tests. The lower bounds adapted from previous works on the traditional model clearly apply for the sample complexity here, even if we allow a higher query complexity. As for our new upper bounds, most of them have a polylogarithmic average queries per sample ratio. It would be interesting to investigate whether the sample complexity can be reduced if we allow a much higher (but still sub-linear in n) number of queries per sample.

2 Preliminaries

2.1 Algorithmic model

As observed by Yao [Yao77], every probabilistic algorithm can be seen as a distribution over a set of deterministic algorithms. Hence we can analyze probabilistic query-making algorithms by analyzing the deterministic algorithms they are supported on.

We observe that we can assume that all samples are drawn before the first query is made, since they are fully independent: the distribution of every sample made does not depend at all on any calculation or queries that occurred before it was taken, and so we can assume that it was taken before any calculation was performed. Based on this observation we can represent our algorithms using a $\{0,1\}$-valued matrix (whose rows are sampled from the distribution), from which the algorithms are allowed to query.

Definition 2.1 (Matrix representation of input access)**.** Considering an algorithm with s samples and q queries, we assume that the samples are all taken at the beginning of the algorithm and are used to populate a matrix $M\in\{0,1\}^{s\times n}$. Then, during the run of the algorithm, each of its queries is represented as a pair $(i,j)\in\{1,\ldots,s\}\times\{1,\ldots,n\}$, for which the answer is $M_{i,j}$.

Definition 2.2 (Adaptive algorithm)**.** Every deterministic algorithm in the Huge Object model with q queries over s samples is equivalent to a pair (T,A), where T is a decision tree of height q in which every internal node contains a query (i,j) (where $1\le i\le s$ is the index of a sample and $1\le j\le n$ is the index to query), and A is the set of accepting leaves.

Definition 2.3 (Non-adaptive algorithm)**.** A deterministic algorithm (T, A) with q queries is *non-adaptive* if, for every $0 \leq i < q$, all internal nodes at the i-th level consist of the exact same query. Every non-adaptive algorithm can be represented as a pair (Q, A), where $Q \subseteq \{1, \ldots, s\} \times \{1, \ldots, n\}$ is a *set* of queries and $A \subseteq \{Q \mapsto \{0, 1\}\}$ is the set of accepted answer vectors.

2.2 Technical components

Fishing expedition

We define an algorithmic primitive that allows us to repeat an execution of a probabilistic subroutine until it is no longer effective. Consider for example a "coupon-collector" type process, but one in which the number of distinct elements is not known to us. The goal is to collect a preset number of elements, but we also want to stop early if we believe that there are no more elements to be effectively collected.

Consider a (probabilistic) subroutine $\mathcal{A}$ that can either fail or succeed. We denote the outcome of an execution of $\mathcal{A}$ by R. In this discussion the outcome includes both the explicit output of the execution and its side effects, which may affect the probabilities for future executions of $\mathcal{A}$. We thus analyze a *sequence* of executions $R_1, \ldots, R_N$, where R_1 is performed over the initial state. We define two behaviors of "coupon collection" that such an $\mathcal{A}$ must present.

Definition 2.4 (Fail stability)**.** Let $\mathcal{A}$ be a subroutine that may succeed or fail. Specifically let $R_1, \ldots, R_N$ be random variables that detail the outputs of the first N executions of $\mathcal{A}$. We say that $\mathcal{A}$ is *fail stable* with respect to a set G of outcomes indicating success, if for every $2 \leq i \leq N$ and every result sequence $(r_1, \ldots, r_{i-1}) \in \mathrm{supp}(R_1, \ldots, R_{i-1})$ for which $r_{i-1} \notin G$:

$$\Pr\left[R_i \in G \mid R_1 = r_1, \ldots, R_{i-2} = r_{i-2}, R_{i-1} = r_{i-1}\right] = \Pr\left[R_{i-1} \in G \mid R_1 = r_1, \ldots, R_{i-2} = r_{i-2}\right]$$

In other words, a failure does not affect the probability of further executions to succeed.

Definition 2.5 (Diminishing returns)**.** Let $\mathcal{A}$ and $R_1, \ldots, R_N$ be as in Definition 2.4. We say that $\mathcal{A}$ *has diminishing returns* with respect to a set G of successful outcomes, if for every $2 \leq i \leq N$ and every result sequence $(r_1, \ldots, r_{i-1}) \in \mathrm{supp}(R_1, \ldots, R_{i-1})$:

$$\Pr\left[R_i \in G \mid R_1 = r_1, \ldots, R_{i-2} = r_{i-2}, R_{i-1} = r_{i-1}\right] \leq \Pr\left[R_{i-1} \in G \mid R_1 = r_1, \ldots, R_{i-2} = r_{i-2}\right]$$

That is, if $\mathcal{A}$ has diminishing returns, then a success in a single execution never increases, but may decrease, the probability of further executions to succeed.

Recall the coupon-collecting example. We expect it to have both fail stability and diminishing returns (with respect to a common set G of outcomes indicating success). If we look for a coupon and do not find it in a single try, nothing happens. Further tries will have the same probability to succeed. On the other hand, if we collect a coupon, then in further tries, there are less uncollected coupons left and it is slightly harder to find an additional one.

The fishing expedition paradigm seeks to collect a goal of k coupons, but "gives up" if it believes that the probability to find an additional coupon is less than some parameter p.

The desired algorithm has three parameters: a threshold p, a confidence q and a goal $k \geq 1$. The input is a subroutine $\mathcal{A}$ with diminishing returns and fail stability (with respect to some common

set G). Informally, the goal of the algorithm is to have k successful executions of $\mathcal{A}$, but also to terminate earlier if the probability of $\mathcal{A}$ to succeed becomes lower than p. Since the algorithm has no actual access to the success probability of $\mathcal{A}$, it should terminate early only if it is confident enough that the success probability of further executions is too low for them to be effective.

Lemma 2.6. *Consider a black box subroutine $\mathcal{A}$ with fail stability (Definition 2.4) and diminishing returns (Definition 2.5) with respect to a common set G of outcomes indicating success.*

For an algorithm that repeatedly executes $\mathcal{A}$, we define the following random variables:

- N *– the number of executions.*
- $R_1, \ldots, R_N$ *– their outcomes.*
- $X_1, \ldots, X_N$ *– indicators of success (that is, $X_i = 1$ if and only if $R_i \in G$).*
- $H = \sum_{i=1}^{N} X_i$ *– the number of successful executions.*
- $\hat{p} = \Pr[X_{N+1} = 1 | R_1, \ldots, R_N]$ *– the success probability of a possible extra execution of $\mathcal{A}$.*

Considering the parameters $p > 0$ (threshold), $q > 0$ (confidence), and $k \geq 1$ (goal), there exists an algorithm that repeatedly executes $\mathcal{A}$ for which $N \leq p^{-1}(4H + 5(\log q^{-1} + \log(\log k + 1))) + 1$ and $H \leq k$, such that with probability higher than $1 - q$, either $H = k$ or $\hat{p} \leq p$ (or both).

Contradiction graph

We define here what it means to be a "counter-example" for having a bounded support size m.

Definition 2.7 (Contradiction graph)**.** Let $x_1, \ldots, x_s \in \{0,1\}^n$ be a sequence of strings. Let $Q \subseteq \{1 \ldots, s\} \times \{1, \ldots, n\}$ be a set of queries. We define *the contradiction graph* of $(x_1, \ldots, x_s; Q)$ as $G(V, E)$ with $V = \{1, \ldots, s\}$, and for every $1 \leq i_1, i_2 \leq s$:

$$\{i_1, i_2\} \in E \iff \exists 1 \leq j \leq n : (x_{i_1})_j \neq (x_{i_2})_j \wedge ((i_1, j), (i_2, j) \in Q)$$

Note that the graph is undirected since the definition of the edges is commutative. It is also clearly without self-loops.

Definition 2.8 (Witness against m-support)**.** Let P be a distribution that is supported by a set of more than m elements. We say that $(x_1, \ldots, x_s; Q)$ is a *witness against m-support* (of P) if $x_1, \ldots, x_s$ are all drawn from P, and their contradiction graph is not m-colorable.

We prove in Lemma 5.14 that calling the above a witness is indeed justified, in the sense that a distribution P has m-support if and only if there is zero probability to draw a tuple $x_1, \ldots, x_s$ for which one can provide a query set Q that makes it a witness.

Definition 2.9 (Explicit witness against m-support)**.** Let P be a distribution that is supported by a set of more than m elements. We say that $(x_1, \ldots, x_s, Q)$ is an *explicit witness against m-support* (of P) if $x_1, \ldots, x_s$ are all drawn from P, and their contradiction graph contains a clique with $m+1$ vertices as a subgraph.

Note that an explicit witness is in particular a witness against m-support, but the converse does not generally hold.

3 Overview of our proofs

Two-sided, non-adaptive lower-bound (Theorem 6.1)

We first describe our lower bound for $\mathcal{S}_2$, which holds the main ideas also for $\mathcal{S}_m$. We begin by analyzing a restricted form of non-adaptive algorithms, which we call *rectangle algorithms*. A rectangle algorithm is characterized by the number of samples s and a set I of indices. Every sample is queried at the indices of I, hence the query complexity is $s \cdot |I|$. We say that $|I|$ is the "width" of the rectangle and that the number of samples is its "height".

Consider the following $O(\varepsilon^{-1})$-query rectangle algorithm: for some hard-coded parameter $\beta > 0$, it chooses a set I of $O(\beta^{-1})$ indices, and then it takes $O(\beta\varepsilon^{-1})$ samples, and then queries every sample on all indices of I.

Now consider the following form of inputs. For some $\alpha > 0$ and two strings a and b for which $d(0,a), d(0,b), d(a,b) = \Theta(\alpha)$, let P be the following distribution. The string 0 is picked with probability $1 - c\alpha^{-1}\varepsilon$, the string a with probability $\frac{c}{2} \cdot \alpha^{-1}\varepsilon$ and the string b with probability $\frac{c}{2} \cdot \alpha^{-1}\varepsilon$, where $c > 1$ is some global constant.

Intuitively, the algorithm finds a witness against 2-support if there is a query common to a and b, at an index j that is not always zero (we call such j a *non-zero index*). That is, there are two necessary conditions to reject: the algorithm must get both a and b as samples, and it must query at an index j for which $(a)_j \neq (b)_j$.

The expected number of non-zero samples that the algorithm gets is $O(\alpha^{-1}\beta)$. If α is much greater than β, then with high probability the algorithm only gets all-zero samples and cannot even distinguish the input distribution from the deterministic all-zero one.

The expected number of non-zero indices that the algorithm chooses is $O(\alpha\beta^{-1})$. If α is much smaller than β, then with high probability all queries are made in "zero indices" and the algorithm again cannot even distinguish the input distribution from the deterministic all-zero one. Thus, the algorithm can reject the input with high probability only if $\alpha \approx \beta$.

Our construction of D_{no} chooses $\alpha = 2^k$ where k is distributed uniformly over its relevant range, to ensure that a rectangle algorithm (with a fixed β) "misses" α with high probability. Intuitively, the idea is that a non-adaptive algorithm must accommodate a large portion of the possible values of α, which would lead to an additional $\log \varepsilon^{-1}$ factor. Then, we show that given an input drawn from D_{no}, if the algorithm did not distinguish two non-zero elements, then the distribution of runs looks exactly the same as the distribution of runs of the same algorithm given an input drawn from D_{yes}, which is supported over 0 and a single a.

To show that the above distributions defeat any non-adaptive algorithm (not just rectangle algorithms), we analyze every index $1 \leq j \leq n$ according to the number of samples which are queried in that index. If few samples are queried, then this index has a high probability of not hitting two non-zero samples, rendering it useless (we gain an important advantage by noting that actually querying j from at least two non-zero samples is required for it to be useful). If many samples are queried then this index may hit many samples, but only few indices can host many queries, which gives us a high probability of all of them together not containing a non-zero index among them.

To extend this result to $m \geq 2$, for every $t \geq 2$ we define a distribution D^t_{no} over inputs that are

supported by $t+1$ elements (one of them being the zero vector), and also ε-far from being supported by m elements (for every $m \leq t/2+1$). As before, we define D_{yes} as a distribution over inputs supported by 2 elements, which is identical to D^1_{no} (including the all-zero elements), and then we proceed with the same argument as before.

One-sided, non-adaptive upper bound (Theorem 7.12)

Let us first consider a "reverse engineering" algorithm: for every $\ell = 2^0, 2^1, \ldots, 2^{\log \varepsilon^{-1}}$, we query $\Theta((\varepsilon^{-1}/\ell) \cdot \log m)$ indices that are common to at least $\ell \cdot m$ samples. Intuitively, according to the analysis of the two-sided lower bound, the algorithm should have roughly $\Omega(m \log m)$ indices that distinguish pairs of elements, which suffice for a contradiction graph that contains an $m+1$-clique.

This intuition appears to be lacking when it comes to showing the correctness of this construction for inputs that lack the special form of D^t_{no}. To be able to handle distance combinations (instead of just one "α" as above), we use a concept of "valid compositions". A valid composition is an ordered combination of samples $(x_1, \ldots, x_k)$ and a sequence of non-decreasing scales $(a_2, \ldots, a_k)$, for which the distances are bounded by $d(x_i, \{x_1, \ldots, x_{i-1}\}) > 2^{-a_i - 1}$ (see Definitions 7.2, 7.3, 7.5).

Querying according to index sets whose random choice follows the prescribed distances distinguishes all elements in a composition with high probability. Our goal is to show the existence of valid compositions of $m+1$ elements in order to ensure that we find an explicit witness, and thus establish the upper bound. However, it is not clear that "long" valid compositions even exist.

We use an extremal probabilistic argument, and show that if the input is ε-far from having support size m, then with high probability no ranked composition with at most m elements is maximal. This implies that the maximally ranked composition (with respect to an order that we define) cannot have less than $m+1$ elements, leading with high probability to finding an explicit witness against m-support.

One-sided, adaptive upper bound (Theorem 9.8)

We adaptively construct a distinguishing sequence that resembles a valid composition, but at some point we decide to "give up" and change phase to another way of querying that is more efficient under some conditions. Luckily, the condition that makes us give up implies them.

For every distance scale, from $\Omega(1)$ to $\frac{1}{m}$, we use the "fishing expedition" paradigm to extend our sequence with as many elements as we can until we are certain enough that it is no longer effective to look for them (or until we find a witness against m-support). Unfortunately, it is possible that at some point the algorithm is certain enough that it is no longer effective to look for elements in any of these scales. At this point, we observe that the contribution of elements with small distance scale to the distance of the input from $\mathcal{S}_m$ is still $\Omega(\varepsilon)$ (that is, we can safely ignore the "rare large-distance elements"). To make use of this observation, the algorithm shifts to the second phase of looking for elements with small distances in a more general way which does not necessarily follow the "theme" of looking for valid compositions.

In the small distance scale phase we construct and maintain a "decision tree" data structure over the existing elements, so that for every element that we need to compare to the existing elements, we can rule out in advance, using only $O(m)$ many queries, all but one of them. This allows us

to save queries, since the smaller distances require the querying of relatively many indices for a comparison, which would have been very inefficient to perform for all existing elements.

One-sided lower-bounds (Theorem 6.6, Corollary 10.11)

We prove that an algorithm obtains a witness against m-support if and only if the contradiction graph (Definition 2.7) is not m-colorable (Lemma 5.14). Hence we look for the lower bound on the number of queries needed to construct a non-m-colorable contradiction graph.

We observe that, given a query set, every index j describes a biclique contradiction graph whose classes are "all samples queried at j for which $x_j = 0$" and "all samples queried at j for which $x_j = 1$". The contradiction graph is the union of these graphs. Then we extend our analysis in two ways, one of which applies to non-adaptive algorithms (giving a $\log \varepsilon^{-1}$ factor) and the other also applies to adaptive ones (giving a $\log m$ factor).

For non-adaptive algorithms, we extend the analysis of the two-sided bound to show that a one-sided algorithm for $\mathcal{S}_m$ requires $\Omega(\varepsilon^{-1} m \log \varepsilon^{-1})$ many queries. The following shows the hardness of "gathering a witness against $\mathcal{S}_m$", which allows for a more versatile argument as compared to the indistinguishability argument that we use for the lower bound of Theorem 6.1.

We use D^t_{no} using $t = 4m/3$. For a non-adaptive algorithm that makes less than $O(\varepsilon^{-1} m \log \varepsilon^{-1})$ queries, the probability that it distinguishes two specific non-zero elements is $\frac{1}{16}$. Considering the contradiction graph, excluding the vertex corresponding to the zero vector, we show that the expected number of edges is at most $\frac{1}{16}\binom{t}{2}$. By Markov's inequality, with probability higher than $\frac{2}{3}$, there are less than $\binom{3t/4-1}{2} = \binom{m-1}{2}$ edges, meaning that this subgraph is colorable using $m-1$ colors. Combined with the vertex corresponding to the zero vector, the contradiction graph is colorable by m colors, hence it cannot be a witness against being supported on only m-support.

For the other bounds we extend a result of [Han64, KS67], providing a lower bound on bi-clique covers of cliques, to show that every biclique cover of a non-m-colorable graph requires at least $(m+1)\log_2(m+1)$ vertices (Lemma 10.4). To show another lower bound against non-adaptive algorithms, we construct a distribution in which a single, "anchor" element is drawn with probability $1 - \Theta(\varepsilon)$. This way, for every non-adaptive algorithm that makes only $o(\varepsilon^{-1} m \log m)$ many queries, the expected number of queries applied to other elements is $o(m \log m)$. By Markov's inequality, with probability $\frac{2}{3}$, only $o(m \log m)$ queries are made in non-zero elements, and in this case, there cannot be a witness against $m-1$ other elements.

This construction cannot immediately be applied to adaptive algorithms, since they can use adaptivity to avoid wasting queries on the anchor element. To overcome this issue, we use two additional methods. The first one is using very short strings, that is, we focus on distributions over $\{0,1\}^{O(\log m)}$ that are ε-far from having m elements in their support (later we prove that the bound also holds for arbitrarily large n using a simple repetition technique). The second method involves using shared-secret code ensembles [BEFLR20] that guarantee, in an appropriate setting, that if the algorithm makes less than $O(\log m)$ queries in an individual sample, then it gathers no information at all. This way, for every individual sample, the algorithm either behaves similarly to a non-adaptive algorithm or makes at least a fixed portion of the maximum number of queries. The exact argument requires a careful analysis of the decision tree of the algorithm.

4 Quick bounds from previous results

We recall some known results for the standard model and use them to derive initial bounds on testing $\mathcal{S}_m$.

Observe that, without loss of generality, we can assume that every sample is queried at least once. Using distributions over sets of of vectors that are mutually 0.499-far, lower bounds for the standard model can be converted to to the Huge Object model, implying in particular the following.

Proposition 4.1 (Proposition 2.8 in [GR22])**.** *Every two-sided error ε-test for $\mathcal{S}_m$ makes at least $\Omega(m/\log m)$ queries (for some fixed ε).*

In the Huge Object model, different samples may be indistinguishable, hence standard-model algorithms cannot be immediately converted to Huge Object model ones. However, we can use the following reduction.

Lemma 4.2 (Theorem 2.2 in [GR22])**.** *Suppose that $\mathcal{P}$ is testable with sample complexity $s(n,\varepsilon)$ in the standard model, and that $\mathcal{P}$ is closed under mapping (note that bounded support properties are closed under mapping). Then for every $\varepsilon > 0$ there exists a non-adaptive ε-test for $\mathcal{P}$ in the Huge Object model that uses $3 \cdot s(m,\varepsilon)$ samples and $O(\varepsilon^{-1}\log(\varepsilon^{-1}s(m,\varepsilon/2)))$ queries per sample.*

Proposition 4.3 (combining [VV17] and [GR22])**.** *There exists a two-sided ε-test for $\mathcal{S}_m$ whose query complexity is $O(\varepsilon^{-3}m\log\varepsilon^{-1})$.*

Proof. In [VV17] there is a two-sided algorithm for m-support testing that uses $O(\varepsilon^{-2}m/\log m)$ samples in the standard model of distribution testing. Lemma 4.2 implies that there exists a non-adaptive ε-testing algorithm for m-support that uses $O(\varepsilon^{-1}m\log\varepsilon^{-1})$ queries per sample, which gives $O(\varepsilon^{-3}m\log\varepsilon^{-1})$ queries in total. □

In the above we used [VV17] rather than the more recent [WY19], since we needed a statement that holds for all values of ε (including those smaller than $1/m$). Proposition 4.3 implies that for every fixed ε and variable m, there exists an $O(m)$ non-adaptive two-sided error ε-test for $\mathcal{S}_m$. In this context we also note the following known bounds.

Theorem 4.4 ([GR22], Corollary 2.3)**.** *For every $\varepsilon > 0$ and $m \geq 2$, there exists a non-adaptive one-sided ε-testing algorithm for $\mathcal{S}_m$ that takes $O(\varepsilon^{-1}m)$ samples and makes $O(\varepsilon^{-2}m\log(m/\varepsilon))$ queries.*

Theorem 4.5 ([AF23], Theorem 6.1)**.** *For every $\varepsilon > 0$ and $m \geq 2$, there exists an adaptive one-sided ε-testing algorithm for $\mathcal{S}_m$ that takes $O(\varepsilon^{-1}m)$ samples and makes $O(\varepsilon^{-1}m^2)$ queries.*

This immediately implies an upper bound of $O(\varepsilon^{-1}m)$ samples for ε-testing $\mathcal{S}_m$ in the standard model of distribution testing. As can be expected, this is tight. The following proposition is considered common knowledge, but as we are not aware of any reference proof, we put one here.

Proposition 4.6. *Every one-sided ε-test for $\mathcal{S}_m$ takes at least $\Omega(\varepsilon^{-1}m)$ samples in the standard model.*

Proof. For $\varepsilon < \frac{1}{24}$, let

$$\mu : \begin{cases} 1 & \text{with probability } 1-2\varepsilon \\ i & \text{with probability } \frac{1}{m}\varepsilon, 2 \le i \le 2m+1 \end{cases}$$

The variation distance of μ from $\mathcal{S}_m$ is greater than ε, since for every set of m elements, there are additional $m+1$ elements in the support of μ whose combined probability is at least $\frac{m+1}{m}\varepsilon$.

Assume that we draw infinitely many independent samples $x_1, x_2, \ldots$. Let $\mathcal{B}$ be the event for $x_1 = 1$. For every $1 \le k \le m+1$, let T_k be the index of the first k-th distinct element. Conditioned on $\mathcal{B}$, for every $2 \le k \le m+1$, $T_k - T_{k-1}$ distributes as a geometric variable with success probability $(2 - \frac{k-1}{2m})\varepsilon \ge \varepsilon$, hence its expected value is at least ε^{-1}, and its variance is at most $\frac{1-\varepsilon}{\varepsilon^2}$. By linearity of expectation,

$$\mathrm{E}[T_{m+1} - T_1] \ge \Pr[\mathcal{B}]\,\mathrm{E}[T_{m+1} - T_1|\mathcal{B}] = \Pr[\mathcal{B}] \sum_{k=2}^{m+1} \mathrm{E}[T_k - T_{k-1}|\mathcal{B}] \ge m\varepsilon^{-1}$$

The differences $T_k - T_{k-1}$ are independent, even if conditioned on $\mathcal{B}$, hence $\mathrm{Var}[T_{m+1} - T_1|\mathcal{B}] = \sum_{k=2}^{m+1} \mathrm{Var}[T_k - T_{k-1}|\mathcal{B}] \le \frac{(1-\varepsilon)m}{\varepsilon^2}$.

For $m \ge 16$, $\sqrt{\mathrm{Var}[T_{m+1} - T_1|\mathcal{B}]} \le \varepsilon^{-1}\sqrt{m} \le \frac{1}{4}\varepsilon^{-1}m \le \frac{1}{4}\,\mathrm{E}[T_{m+1} - T_1|\mathcal{B}]$. By Chebyshev's inequality ($\Pr[X < \mathrm{E}[X] - \lambda\sqrt{\mathrm{Var}[X]}] < \lambda^{-2}$), we obtain

$$\Pr[T_{m+1} - T_1 < \frac{1}{2}\varepsilon^{-1}m|\mathcal{B}] \le \Pr[T_{m+1} - T_1 < \mathrm{E}[T_{m+1} - T_1|\mathcal{B}] - 2\sqrt{\mathrm{Var}[T_{m+1} - T_1|\mathcal{B}]}] < \frac{1}{4}$$

Hence, for an algorithm that takes $\lfloor \frac{1}{2}\varepsilon^{-1}m \rfloor$ samples, the probability to draw $m+1$ distinct samples is less than $\Pr[\neg\mathcal{B}] + \frac{1}{4} \le 2\varepsilon + \frac{1}{4} < \frac{1}{3}$. Since a one-sided algorithm cannot reject without observing more than m distinct members in the support of μ, this concludes the proof. □

As with Proposition 4.1, the above can be immediately converted to a Huge Object model bound.

Proposition 4.7. *Every one-sided ε-test for $\mathcal{S}_m$ in the Huge Object model must make at least $\Omega(\varepsilon^{-1}m)$ queries as well.*

In this paper we improve this proposition, showing a gap between the standard model and the Huge Object model for one-sided error tests.

5 Additional preliminaries

5.1 Common notations

For an integer n and a set $A \subseteq \{1, \ldots, n\}$, we denote by 1_A the n-bit binary string for which the i-th bit is 1 if and only if $i \in A$. Given two sets $A, B \subseteq \{1, \ldots, n\}$, we let $A \Delta B$ be their symmetric difference. For a finite set Ω, we define $\mathcal{D}(\Omega)$ as the set of all distributions over Ω.

The following is a useful notation for analyzing expectations of random variables.

Definition 5.1 (Contribution of a random variable over an event)**.** Let μ be a probability distribution, X be a random variable and B be an event. We define the *contribution* $\mathrm{Ct}[X \mid B]$ *of* X *over* B *to be* 0 *if* $\Pr[B] = 0$*, and otherwise by*

$$\mathrm{Ct}\left[X \mid B\right] = \mathrm{E}[X \mid B] \cdot \Pr[B] = \sum_{x \in B} \mu(x) X(x).$$

Observation 5.2. *The following properties are immediate.*

- *Inclusion-Exclusion:* $\mathrm{Ct}\left[X|B_1 \vee B_2\right] = \mathrm{Ct}\left[X|B_1\right] + \mathrm{Ct}\left[X|B_2\right] - \mathrm{Ct}\left[X|B_1 \wedge B_2\right]$.
- *Total expectation: If* $\Pr\left[\bigvee_{i=1}^{k} B_i\right] = 1$ *and the events* $B_1, \ldots, B_k$ *are mutually disjoint, then* $\mathrm{E}\left[X\right] = \sum_{i=1}^{k} \mathrm{Ct}\left[X|B_i\right]$.

Definition 5.3 (Sample map)**.** Let P be a distribution over a finite set Ω_1 and let $f : \Omega_1 \to \Omega_2$ be a mapping to a finite set Ω_2 (possibly $\Omega_1 = \Omega_2$). The *sample map* of P according to f, denoted by $f(P)$, is the distribution Q over Ω_2 for which, for every $b \in \Omega_2$, $\Pr_Q[b] = \Pr_{a \sim P}[f(a) = b]$.

Definition 5.4 (The bounded support property)**.** We define the following variants of bounded support properties:

1. Let m be a fixed number. The property of distributions that are supported by at most m elements is denoted by $\mathcal{S}_m$.
2. Let A be a fixed set of elements. The property of distributions that are supported by a subset of A (possibly A itself) is denoted by $\mathcal{S}_A$.
3. Let $f : \mathbb{N} \to \mathbb{N}$ be a fixed function. The property of distributions over $\{0,1\}^n$ that are supported by at most $f(n)$ elements is denoted by $\mathcal{S}_f$.

5.2 Analysis of probabilistic algorithms

To be able to state and prove lower bounds, we use the notion of a "distribution of runs". Informally, it is the behavior of an algorithm $\mathcal{A}$ on an input that is drawn from a distribution. If we have a distribution over inputs in a property and another distribution over inputs that are ε-far from the property, and their distributions over runs (with respect $\mathcal{A}$) are indistinguishable, then $\mathcal{A}$ cannot be an ε-test for that property.

Definition 5.5 (Distribution of runs)**.** Let T be an s-sample, q-query decision tree and let D be a distribution over inputs. The *distribution of* T*-runs on* D is denoted by $\mathcal{R}(T, D)$, and is defined over $\{0,1\}^q$ as follows: first draw an input $P \sim D$. Then draw s independent samples from P, and make the queries of T (following the corresponding root to leaf path). The result is the vector of answers (of size q, padded with zeroes if necessary).

Note that for non-adaptive algorithms, the distribution of runs can be seen as a distribution over functions from the fixed query set Q to $\{0,1\}$, and can be obtained by drawing a distribution P, populating the matrix M using s independent samples from P, and then using the restriction $M|_Q$.

Definition 5.5 can be naturally generalized to probabilistic algorithms since they can be seen as a distribution over deterministic ones. That is, $\mathcal{R}(\mathcal{A}, D)$ distributes like "draw $T \sim \mathcal{A}$ and then draw the run according to $\mathcal{R}(T, D)$".

Definition 5.6 (Conditional distribution of runs)**.** Let T be an s-sample, q-query decision tree and let D be a distribution over inputs. Also, let B be some event. The *distribution of T-runs on D conditioned on B* is denoted by $\mathcal{R}(T, D|B)$, and is defined as the distribution $\mathcal{R}(T, D)$ restricted to B.

Lemma 5.7 (Lower bounds by Yao's principle)**.** *Let $\mathcal{T}$ be a class of deterministic decision trees (which in turn define a class of probabilistic algorithms) and let $q > 0$. Let D_1 and D_2 be two distributions over inputs. If, for every decision tree $T \in \mathcal{T}$ of size less than q, $d(\mathcal{R}(T, D_1), \mathcal{R}(T, D_2)) < \frac{1}{3}$, then every probabilistic algorithm that distinguishes D_1 and D_2 (with error less than $\frac{1}{3}$) must make make at least q queries (with positive probability).*

The simulation method, described in the lemma below (whose proof is trivial), is a useful "user interface" for Yao's principle.

Lemma 5.8 (The simulation method)**.** *Let T be an s-sample, q-query decision tree, and let D_1, D_2 be distributions of inputs. Assume that there exist two events B_1 and B_2 for which $\mathcal{R}(T, D_1|\neg B_1)$ is identical to $\mathcal{R}(T, D_2|\neg B_2)$. In this setting, $d(\mathcal{R}(T, D_1), \mathcal{R}(T, D_2)) \leq \Pr_{\mathcal{R}(T,D_1)}[B_1] + \Pr_{\mathcal{R}(T,D_2)}[B_2]$.*

5.3 Analysis of properties of distributions

Definition 5.9 (Being closed under mapping, [GR22])**.** A property $\mathcal{P}$ of distributions over $\{0,1\}^n$ is *closed under mapping* if for every function $f : \{0,1\}^n \to \{0,1\}^n$ and for every distribution $P \in \mathcal{P}$ we have $f(P) \in \mathcal{P}$.

Note that the bounded support properties ($\mathcal{S}_m$ for fixed sizes and $\mathcal{S}_f$ for functions) are closed under mapping.

Definition 5.10 (Label-invariance)**.** A property $\mathcal{P}$ of distributions over $\{0,1\}^n$ is *label-invariant* if for every distribution $P \in \mathcal{P}$ and for every permutation $\sigma : \{0,1\}^n \to \{0,1\}^n$, $\sigma(P) \in \mathcal{P}$ as well.

Note that every property that is closed under mapping is also label-invariant.

The following proposition informally states that a specific strong constraint fully characterizes the bounded support property. We prove it in Appendix B.

Proposition 5.11. *Consider any label-invariant property of distributions $\mathcal{P}$ that has a one-sided ε-test for every $\varepsilon > 0$ (with any number of samples and queries). There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathcal{P} = \mathcal{S}_f$.*

5.4 Analysis of the bounded support property

We start by stating simple observations that we use throughout our work.

Lemma 5.12 ([AF23])**.** *Let P be a distribution over $\{0,1\}^n$ and A be a subset of $\{0,1\}^n$. Let $f : \mathrm{supp}(P) \to A$ be the following function: for every $x \in \mathrm{supp}(P)$, $f(x) = \arg\min_{y \in A}\{d(x, y)\}$ (ties are broken arbitrarily). Then $d(P, f(P)) = d(P, \mathcal{S}_A)$.*

Observation 5.13 ([AF23])**.** *Let $A \subseteq \{0,1\}^n$ be a set of elements, and let P be a distribution that is ε-far from being supported by any subset of A. Then $d(P, \mathcal{S}_A) = \mathrm{E}_{x \sim P}[d(x, A)] > \varepsilon$.*

The following lemma shows the correctness of Definition 2.8 (witness against m-support).

Lemma 5.14. *Let $x_1, \ldots, x_s \in \mathrm{supp}(P)$ be a set of samples and let $Q \subseteq \{1, \ldots, s\} \times \{1, \ldots, n\}$ be a query set. Let $Q_1, \ldots, Q_s$ be the sample-specific query sets, that is, $Q = \bigcup_{i=1}^{s} (\{i\} \times Q_i)$, and let G be the contradiction graph as per Definition 2.7. If G is not colorable by m colors, then $|\{x_1, \ldots, x_s\}| > m$. And if G is colorable by m colors, then there exists $\hat{P}$ with $|\mathrm{supp}(\hat{P})| \leq m$ and a sequence $y_1, \ldots, y_s \in \mathrm{supp}(\hat{P})$ such that for every $1 \leq i \leq s$, $x_i|_{Q_i} = y_i|_{Q_i}$.*

Proof. Let $A \subseteq \{0,1\}^n$, and let $f : \{x_1, \ldots, x_m\} \to A$ be a mapping such that for every $1 \leq i \leq s$, $x_i|_{Q_i} = (f(i))|_{Q_i}$. If G is not colorable by m colors, then f cannot be a valid coloring unless $|A| > m$. Specifically, $|\{x_1, \ldots, x_s\}| > m$, since with $A = \{x_1, \ldots, x_s\}$ we have the coloring $f(i) = x_i$.

Now assume that G is colorable by m colors. Let $f : \{1, \ldots, s\} \to \{1, \ldots, m\}$ be a valid coloring. Let $\hat{A} = \{y_1, \ldots, y_m\}$ be the following set: for every $1 \leq k \leq m$, $y_k \in \{0,1\}^n$ is a string for which $y_k|_{Q_i} = x_i|_{Q_i}$ for every i for which $f(i) = k$. A concrete example for y_k would be: the j-th bit is 1 if and only if there exists $1 \leq i \leq s$ such that $f(i) = k$, $j \in Q_i$ and $x_i|_j = 1$. Let $\hat{P}$ be the uniform distribution over this $\hat{A}$. There exist $y_1, \ldots, y_s \in \mathrm{supp}(\hat{P})$ such that for every $1 \leq i \leq s$, $x_i|_{Q_i} = y_i|_{Q_i}$. Finally, note that $|\mathrm{supp}(\hat{P})| \leq m$. □

The following lemma is a counterpart of Lemma 5.7 for the special case of one-sided error. It follows from the same observation by Yao that a probabilistic algorithm can be viewed as a distribution over deterministic algorithms, along with the observation that without loss of generality a one-sided error algorithm rejects if and only if it finds a witness against the property.

Lemma 5.15 (Lower bounds by Yao's principle for one-sided algorithms)**.** *Let $\mathcal{P}$ be a property, $\mathcal{T}$ be a class of deterministic decision trees and let $q > 0$. Let D be a distribution over inputs that always draws an input distribution that is ε-far from $\mathcal{P}$.*

Consider a decision tree $T \in \mathcal{T}$, and let $\mathcal{B}_T$ be the set of witnesses against $\mathcal{P}$ (that is, the set of unreachable leaves, or runs, for any input $P \in \mathcal{P}$). If $\Pr_{\mathcal{R}(T,D)}[\mathcal{B}_T] < \frac{1}{3}$, for every $T \in \mathcal{T}$ of size less than q, then every one-sided probabilistic algorithm for $\mathcal{P}$ conforming to $\mathcal{T}$ must make at least q queries (with positive probability).

6 Superlinear lower-bound for non-adaptive 2-support test

We show an $\Omega(\varepsilon^{-1} \log \varepsilon^{-1})$ lower bound for non-adaptive 2-support tests, even with two-sided error. Generalizing the construction, we show a bound of $\Omega(\varepsilon^{-1} \log \varepsilon^{-1})$ for m-support tests with any $m \geq 2$ (note that a single $f(\varepsilon) = \Omega(\varepsilon^{-1} \log \varepsilon^{-1})$ holds simultaneously for all $m \geq 2$, rather than having an implicit coefficient that depends on m), and a one-sided bound of $\Omega(\varepsilon^{-1} \log \varepsilon^{-1} \cdot m)$.

Theorem 6.1. *Every non-adaptive ε-test for $\mathcal{S}_m$ must make $\Omega(\varepsilon^{-1} \log \varepsilon^{-1})$ queries, even if it has two-sided error.*

We prove this theorem in this section. To do so, we first define distributions over inputs (which are in themselves distributions over $\{0,1\}^n$) and analyze them.

Definition 6.2 (D^t_{no} for ε)**.** Draw α such that $\log_2 \alpha^{-1}$ is uniform over $\{2, \ldots, \lfloor \log_2 \varepsilon^{-1} \rfloor - 2\}$. Draw a set $D \subseteq \{1, \ldots, n\}$ such that for every $1 \leq j \leq n$, $\Pr[j \in D] = 4\alpha$, independently. Then, for every $1 \leq k \leq t$, draw a set $A_k \subseteq D$ such that for every $j \in D$, $\Pr[j \in A_k | j \in D] = \frac{1}{2}$,

independently. The resulting input is defined as the following distribution over $\{0,1\}^n$:

$$P: \begin{cases} 0 & \text{with probability } 1-2\alpha^{-1}\varepsilon \\ 1_{A_1} & \text{with probability } 2\alpha^{-1}\varepsilon/t \\ \vdots & \\ 1_{A_t} & \text{with probability } 2\alpha^{-1}\varepsilon/t \end{cases}$$

Definition 6.3 (D_{yes} for ε)**.** We simply define D_{yes} as D^1_{no}. An equivalent definition of D_{yes} is the following: draw α such that $\log_2 \alpha^{-1}$ is uniform over $\{2,\ldots,\lfloor \log_2 \varepsilon^{-1} \rfloor - 2\}$, and then draw a set $A \subseteq \{1,\ldots,n\}$ such that for every $1 \le j \le n$, $\Pr[j \in A] = 2\alpha$, independently. The resulting input is defined as the following distribution over $\{0,1\}^n$:

$$P: \begin{cases} 0 & \text{with probability } 1-2\alpha^{-1}\varepsilon \\ 1_A & \text{with probability } 2\alpha^{-1}\varepsilon \end{cases}$$

First we show that D^t_{no} and D_{yes} can be used to demonstrate lower bounds for ε-testing $\mathcal{S}_m$. Trivially, D_{yes} draws a distribution in $\mathcal{S}_2$ (and hence $\mathcal{S}_m$) with probability 1.

Observation 6.4. *If $a_1,\ldots,a_k$ are non-negative integers, then $\sum_{i=1}^k \lfloor a_i/2 \rfloor \ge \frac{\sum_{i=1}^k a_i - k + 1}{2}$.*

Lemma 6.5. *Let $t \ge 2$ and $P \sim D^t_{\text{no}}$. For sufficiently large n (as a function of t and ε) and for every $1 \le m \le t$, with probability at least $1 - \frac{3}{1000}$, the distance of P from $\mathcal{S}_m$ is more than $(2 - \frac{2m-2}{t})\varepsilon$.*

Concretely, D^2_{no} is ε-far from $\mathcal{S}_2$ with probability at least $1 - \frac{3}{1000}$, and for every $t \ge 3$, P is ε-far from $\mathcal{S}_{\lceil t/2 \rceil}$ with this probability.

Proof. Let $n > 480000\varepsilon^{-1}t^2(\ln t + 10)$. By the multiplicative Chernoff's bound (Lemma A.4), every individual event of one of the following forms happens with probability at least $1 - \frac{3}{1000t^2}$:

- $\left(4 - \frac{1}{100t}\right)\alpha n < |D| < \left(4 + \frac{1}{100t}\right)\alpha n$.
- $\left(2 - \frac{1}{100t}\right)\alpha n < |A_k| < \left(2 + \frac{1}{100t}\right)\alpha n$, for every $1 \le k \le t$.
- $\left(2 - \frac{1}{100t}\right)\alpha n < |A_{k_1} \Delta A_{k_2}| < \left(2 + \frac{1}{100t}\right)\alpha n$, for every $1 \le k_1 < k_2 \le t$.

By the union bound over $\binom{t}{2} + t + 1 \le t^2$ events, with probability $1 - \frac{3}{1000}$, all of the above events happen simultaneously.

Assume that this is the case, and consider a set A of size m. Let $f : \text{supp}(P) \to A$ be a mapping from every element in the support of P to a closest element in A (ties are broken arbitrarily). By Lemma 5.12, f realizes the distance from P to a closest distribution supported by A, that is, $d(P, f(P)) = d(P, \mathcal{S}_A)$. Consider some $a \in A$. For every $u, v \in f^{-1}(a)$, the contribution to the distance is at least $d(u,a) + d(v,a) \ge d(u,v) > \left(2 - \frac{1}{100t}\right)\alpha$. The probability weight of u, v is at least $2\alpha^{-1}\varepsilon/t$ individually, hence their contribution to the distance is more than $\left(4 - \frac{1}{50t}\right)\varepsilon/t$.

Considering $\lfloor |f^{-1}(a)|/2 \rfloor$ disjoint pairs of elements in $f^{-1}(a)$, the contribution of $f^{-1}(a)$ elements to the distance $d(P, \mathcal{S}_A)$ is at least $(4 - \frac{1}{50t})\frac{\varepsilon}{t}\lfloor |f^{-1}(a)|/2 \rfloor$. Summing over all $a \in A$ we have:

$$\begin{aligned} d(P, \mathcal{S}_A) > \frac{\left(4 - \frac{1}{50t}\right)\varepsilon}{t} \sum_{a \in A} \lfloor |f^{-1}(a)|/2 \rfloor &\underset{(*)}{\geq} \frac{(4 - \frac{1}{50t})\varepsilon}{t} \cdot \frac{|\mathrm{supp}(P)| - |A| + 1}{2} \\ &= \left(2 - \frac{1}{100t}\right)\frac{\varepsilon}{t}(t + 2 - m) \geq \left(2 - \frac{2m-2}{t}\right)\varepsilon \end{aligned}$$

The starred transition is implied by Observation 6.4. This holds for every A of size m, hence $d(P, \mathcal{S}_m) > (2 - \frac{2m-2}{t})\varepsilon$. □

We show that a non-adaptive algorithm that only uses $q < \frac{1}{170}\varepsilon^{-1}\log\varepsilon^{-1}$ queries cannot distinguish D^t_{no} from D_{yes} with error smaller than $\frac{2}{7} < \frac{1}{3} - \frac{3}{1000}$, for every $t \geq 2$ and sufficiently small ε (whose bound is independent of t).

Proof (of Theorem 6.1). Let Q be a query set of size $q < \frac{1}{300}\varepsilon^{-1}\log\varepsilon^{-1}$. Our D_{no} for this proof is D^t_{no} with $t = 2m$ (and $D_{\mathrm{yes}} = D^1_{\mathrm{no}}$). By Lemma 6.5, a distribution that is drawn from D^{2m}_{no} is $(2 - \frac{2m-2}{t})\varepsilon$-far from $\mathcal{S}_m$ with probability $\frac{3}{1000}$, which is ε-far for every $m \geq 2$.

Let R^Q be the following distribution over responses to the query set:

- Choose α such that $\log\alpha^{-1}$ is uniform in $\{2, \ldots, \lfloor \log\varepsilon^{-1} \rfloor - 2\}$.
- Choose a set $D \subseteq \{1, \ldots, n\}$ such that for every $1 \leq j \leq n$, $\Pr[j \in A] = 4\alpha$, independently.
- Choose a set $S \subseteq \{1, \ldots, s\}$ such that for every $1 \leq i \leq s$, $\Pr[i \in S] = 2\alpha^{-1}\varepsilon$, independently.
- The result is $f : Q \to \{0, 1\}$, where $f(i, j) = 0$ if $i \notin S$ or $j \notin D$, and for every $(i, j) \in S \times D$, $\Pr[f(i, j) = 1] = \frac{1}{2}$, independently.

We show that both $\mathcal{R}(Q, D_{\mathrm{yes}})$ and $\mathcal{R}(Q, D_{\mathrm{no}})$ are $\frac{1}{7}$-close to a distribution related to R^Q, hence they must be $\frac{2}{7}$-close to each other. By Yao's method (Lemma 5.7), this means that there is no non-adaptive ε-test for $\mathcal{S}_m$ making only q queries, as required.

Let $\mathcal{B}$ be the following bad event: there exists an index $j \in D$ at which two (or more) non-zero samples are queried. If $\mathcal{B}$ does not happen then the algorithm does not even have an opportunity to compare those non-zero samples (which in mathematical terms means having the same distribution over the query answers when conditioned on this event). In R^Q, $\mathcal{B}$ is defined correspondingly as the event that there exists $j \in D$, $i_1, i_2 \in S$ $(i_1 \neq i_2)$ for which $(i_1, j), (i_2, j) \in Q$.

The rest of the proof has two parts: the first is proving that the probability of $\mathcal{B}$ is less than $\frac{1}{3} - \frac{3}{1000}$; and the second is proving that D_{no} and D_{yes} are both identical to R^Q when those are conditioned on the negation of $\mathcal{B}$.

We decompose Q by its indices. That is, $Q = \bigcup_{j=1}^{n} \{j\} \times S_j$, where S_j are the samples that the algorithm queries at the j-th index. For every $2 \leq \ell \leq q$, let $w_\ell = |\{1 \leq j \leq n : |S_j| = \ell\}|$ be the number of indices that have exactly ℓ samples.

Consider some index j with ℓ samples. Given α, the probability to draw more than one non-zero sample among these ℓ samples is bounded by $\min\{1, (2\alpha^{-1}\varepsilon\ell)^2\}$. Note that the non-constant expression is effective only when $\log\alpha^{-1} \leq \log\varepsilon^{-1} - \log\ell - 1$.

For every $1 \leq j \leq n$, let X_j be an indicator for being queried in two (or more) non-zero samples, as well as belonging to D. If $X_j = 0$, then the algorithm cannot distinguish between any two non-zero samples using j-queries. Consider some index j with $|S_j| = \ell$ samples.

$$\begin{aligned}
\mathrm{E}[X_j] &= \Pr[X_j = 1] = \sum_{a=2}^{\lfloor \log \varepsilon^{-1} \rfloor - 2} \Pr\left[\alpha = 2^{-a}\right] \Pr\left[j \in D \middle| \alpha = 2^{-a}\right] \Pr\left[X_j = 1 \middle| \alpha = 2^{-a}, j \in D\right] \\
&\leq \frac{1}{\lfloor \log \varepsilon^{-1} \rfloor - 3} \sum_{a=2}^{\lfloor \log \varepsilon^{-1} \rfloor - 2} 4 \cdot 2^{-a} \min\{1, (2 \cdot 2^a \varepsilon \ell)^2\} \\
&= \frac{1}{\lfloor \log \varepsilon^{-1} \rfloor - 3} \left(\sum_{a=2}^{\lfloor \log \varepsilon^{-1} - \log \ell \rfloor - 1} 4 \cdot 2^{-a} (2 \cdot 2^a \varepsilon \ell)^2 + \sum_{a=\lfloor \log \varepsilon^{-1} - \log \ell \rfloor}^{\lfloor \log \varepsilon^{-1} \rfloor - 2} 4 \cdot 2^{-a} \right) \\
&= \frac{1}{\lfloor \log \varepsilon^{-1} \rfloor - 3} \left(\left(16 \varepsilon^2 \ell^2 \sum_{a=2}^{\lfloor \log \varepsilon^{-1} - \log \ell \rfloor - 1} 2^a \right) + \left(4 \sum_{a=\lfloor \log \varepsilon^{-1} - \log \ell \rfloor}^{\lfloor \log \varepsilon^{-1} \rfloor - 2} 2^{-a} \right) \right) \\
&\leq \frac{1}{\lfloor \log \varepsilon^{-1} \rfloor - 3} \left(\left(16 \varepsilon^2 \ell^2 \cdot \varepsilon^{-1} / \ell\right) + (4 \cdot 2\ell\varepsilon) \right) \leq \frac{24 \varepsilon \ell}{\lfloor \log \varepsilon^{-1} \rfloor - 3}
\end{aligned}$$

Let X be the number of indices that are queried in two (or more) non-zero samples, and also belong to D. Considering all indices, by linearity of expectation:

$$\begin{aligned}
\mathrm{E}[X] = \sum_{\ell} \sum_{j : |S_j| = \ell} \mathrm{E}[X_j] &\leq \sum_{\ell} \frac{24 \varepsilon \ell}{\lfloor \log \varepsilon^{-1} \rfloor - 3} w_\ell \\
&< \frac{24 \varepsilon}{\lfloor \log \varepsilon^{-1} \rfloor - 3} \cdot \frac{1}{300} \varepsilon^{-1} \log \varepsilon^{-1} = \frac{2}{25} \cdot \frac{\log \varepsilon^{-1}}{\lfloor \log \varepsilon^{-1} \rfloor - 3} \underset{\varepsilon < 2^{-12}}{<} \frac{1}{7}
\end{aligned}$$

To summarize the first part of the proof, $\Pr[\mathcal{B}] = \Pr[X \neq 0] \leq \mathrm{E}[X] < \frac{1}{7}$ for $\varepsilon < 2^{-12}$. Note that at this point we showed a lower bound of $\Omega(\varepsilon^{-1} \log \varepsilon^{-1})$ queries for a one-sided ε-test of $\mathcal{S}_m$.

In the following we show that if $\mathcal{B}$ did not happen then D_{no} and D_{yes} are identical to R^Q conditioned on $\neg\mathcal{B}$. Note that D_{yes} always draws a distribution in $\mathcal{S}_m$, since $\mathcal{S}_2 \subseteq \mathcal{S}_m$ for every $m \geq 2$.

Consider the answer function of the run. For every query $(i, j) \in Q$, if the i-th sample is a zero sample (that is, $i \notin S$), then $f(i, j) = 0$. Additionally, if $j \notin D$, then $f(i, j) = 0$ as well. For $(i, j) \in S \times D$, $\Pr[f(i, j) = 1 | i \in S, j \in D] = \Pr[(1_{A_{k_j}})_j = 1 | j \in D] = \frac{1}{2}$. Also, if $\mathcal{B}$ does not happen, then for every $(i, j) \neq (i', j')$ that are both in $S \times D$ we must have $j \neq j'$. Thus all the "$f(i, j)$" events for $Q \cap (S \times D)$ are mutually independent, making the distribution of the answers to the queries, when conditioned on $\neg\mathcal{B}$, identical to R^Q conditioned on $\neg\mathcal{B}$.

The above argument holds for both D_{yes} and D_{no}, hence $\mathcal{R}(Q, D_{\mathrm{yes}})$ and $\mathcal{R}(Q, D_{\mathrm{no}})$ are $\frac{2}{7}$-close to each other (since $2 \Pr[\mathcal{B}] \leq \frac{2}{7}$). Since D_{no} draws a distribution ε-far from $\mathcal{S}_m$ with probability $\frac{3}{1000}$, and D_{yes} always draws a distribution that belongs to $\mathcal{S}_m$, every non-adaptive ε-test for $\mathcal{S}_m$ must make at least $\Omega(\varepsilon^{-1} \log \varepsilon^{-1})$ queries. □

6.1 Composite lower bound for non-adaptive one-sided ε-tests of $\mathcal{S}_m$

Theorem 6.6. *Every one-sided non-adaptive ε-test of $\mathcal{S}_m$ must make $\Omega(\varepsilon^{-1} m \log \varepsilon^{-1})$ queries.*

Proof. Without loss of generality, assume that m is divisible by 3. Our D_{no} for this proof is D_{no}^t with $t = \frac{4}{3}m$ and $\hat{\varepsilon} = 2\varepsilon$. By Lemma 6.5, with probability $1 - o(1)$, D_{no} draws a distribution that is $(\frac{1}{2} + \frac{3}{4m})\hat{\varepsilon}$-far from $\mathcal{S}_m$, which is ε-far. For a bound against a one-sided test we will bound the probability of finding a witness against $\mathcal{S}_m$ under this distribution (and then use Lemma 5.15), so in particular there is no D_{yes}.

Let Q be a query set of size $q < \frac{m}{1000}\varepsilon^{-1} \log \varepsilon^{-1} \leq \frac{m}{500}\hat{\varepsilon}^{-1} \log \hat{\varepsilon}^{-1}$. Fix two indices $1 \leq k_1 < k_2 \leq \frac{4}{3}m$. For every $1 \leq j \leq n$, let X_{j,k_1,k_2} be an indicator for having queries in two (or more) samples in $\{1_{A_{k_1}}, 1_{A_{k_2}}\}$, as well as belonging to D. If $X_{j,k_1,k_2} = 0$, then the algorithm cannot distinguish between $1_{A_{k_1}}$ and $1_{A_{k_2}}$ using j-queries. Also, let $X_{k_1,k_2} = \sum_{j=1}^{n} X_{j,k_1,k_2}$ be the number of indices that have an opportunity to distinguish between $1_{A_{k_1}}$ and $1_{A_{k_2}}$ (note that it is similar to X in Theorem 6.1). Finally let $\mathcal{B}_{k_1,k_2}$ be the event for $X_{k_1,k_2} > 0$, and $Y_{k_1,k_2} \in \{0,1\}$ be the indicator for $\mathcal{B}_{k_1,k_2}$. The number of edges in the contradiction graph (Definition 2.7), excluding the vertex representing the zero vector, is bounded by $\sum_{1 \leq k_1 < k_2 \leq \frac{4}{3}m} Y_{k_1,k_2}$ (if $Y_{k_1,k_2} = 1$ then the algorithm has an *opportunity* to distinguish them, but it can still fail to do so).

Consider some $1 \leq k_1 < k_2 \leq \frac{4}{3}m$ and some index j of with ℓ samples. The probability to draw a single sample in $\{1_{A_{k_1}}, 1_{A_{k_2}}\}$ is $2 \cdot (2\alpha^{-1}\hat{\varepsilon}/(\frac{4}{3}m)) = 3\alpha^{-1}\hat{\varepsilon}/m$. The probability to draw more than one sample in $\{1_{A_{k_1}}, 1_{A_{k_2}}\}$ among these ℓ samples is bounded by $\min\{1, (3\alpha^{-1}\varepsilon\ell/m)^2\}$. As in the previous subsection, for every $2 \leq \ell \leq q$, let $w_\ell = |\{1 \leq j \leq n : |S_j| = \ell\}|$ be the number of indices that have exactly ℓ samples.

For convenience, let $h = \log \hat{\varepsilon}^{-1} + \log m - \log \ell$.

$$
\begin{aligned}
\mathrm{E}[X_{j,k_1,k_2}] &= \Pr[X_{j,k_1,k_2}=1] \\
&= \sum_{a=2}^{\lfloor \log \hat{\varepsilon}^{-1} \rfloor -2} \Pr\left[\alpha = 2^{-a}\right] \Pr\left[j \in D \middle| \alpha = 2^{-a}\right] \Pr\left[X_{j,k_1,k_2}=1 \middle| \alpha = 2^{-a}, j \in D\right] \\
&\leq \frac{1}{\lfloor \log \hat{\varepsilon}^{-1} \rfloor - 3} \sum_{a=2}^{\lfloor \log \hat{\varepsilon}^{-1} \rfloor -2} 4 \cdot 2^{-a} \min\{1, (3 \cdot 2^a \hat{\varepsilon} \ell / m)^2\} \\
&= \frac{1}{\lfloor \log \hat{\varepsilon}^{-1} \rfloor - 3} \left(\sum_{a=2}^{\lfloor h - \log 3 \rfloor} 4 \cdot 2^{-a} (3 \cdot 2^a \hat{\varepsilon} \ell / m)^2 + \sum_{a=\lfloor h - \log 3 \rfloor + 1}^{\lfloor \log \hat{\varepsilon}^{-1} \rfloor -2} 4 \cdot 2^{-a} \right) \\
&= \frac{1}{\lfloor \log \hat{\varepsilon}^{-1} \rfloor - 3} \left(\left(\frac{36}{m^2} \hat{\varepsilon}^2 \ell^2 \sum_{a=2}^{\lfloor h - \log 3 \rfloor} 2^a \right) + \left(4 \sum_{a=\lfloor h - \log 3 \rfloor + 1}^{\lfloor \log \hat{\varepsilon}^{-1} \rfloor -2} 2^{-a} \right) \right) \\
&\leq \frac{1}{\lfloor \log \hat{\varepsilon}^{-1} \rfloor - 3} \left(\left(\frac{36}{m^2} \hat{\varepsilon}^2 \ell^2 \cdot \frac{2}{3} \hat{\varepsilon}^{-1} m / \ell \right) + \left(\frac{8}{3} \cdot \ell \hat{\varepsilon} / m \right) \right) \\
&\leq \frac{27 \hat{\varepsilon} \ell}{m(\lfloor \log \hat{\varepsilon}^{-1} \rfloor - 3)}
\end{aligned}
$$

Considering all indices,

$$
\begin{aligned}
\mathrm{E}[X_{k_1,k_2}] &= \sum_{\ell} \sum_{j:|S_j|=\ell} \mathrm{E}[X_{j,k_1,k_2}] \\
&\leq \sum_{\ell} \frac{27 \hat{\varepsilon} \ell}{m(\lfloor \log \hat{\varepsilon}^{-1} \rfloor - 3)} w_\ell \\
&< \frac{27 \hat{\varepsilon}}{m(\lfloor \log \hat{\varepsilon}^{-1} \rfloor - 3)} \cdot \frac{m}{500} \hat{\varepsilon}^{-1} \log \hat{\varepsilon}^{-1} = \frac{27}{500} \cdot \frac{\log \hat{\varepsilon}^{-1}}{\lfloor \log \hat{\varepsilon}^{-1} \rfloor - 3} < \frac{1}{16}
\end{aligned}
$$

(The last transition is correct for every sufficiently small $\hat{\varepsilon}$). That is, $\Pr[Y_{k_1,k_2}=1] = \Pr[\mathcal{B}_{k_1,k_2}] < \frac{1}{16}$ for every $1 \leq k_1 < k_2 \leq \frac{4}{3}m$.

By linearity of expectation, the expected number of edges in the contradiction graph, excluding the vertex representing the zero vector, is bounded by $\frac{1}{16}\binom{4m/3}{2} < \frac{1}{4}\binom{2m/3}{2}$. By Markov's inequality, with probability at least $\frac{3}{4}$, this contradiction graph has less than $\binom{2m/3}{2}$ edges, hence it is colorable using $\frac{2}{3}m$ colors. Considering the zero vector as well, the contradiction graph can be colored using $\frac{2}{3}m + 1 \leq m$ colors. In this case, there is no witness against $P \in \mathcal{S}_m$. The lower bound is then implied by Lemma 5.15. □

7 Quasilinear non-adaptive one-sided m-support test

We show a one-sided non-adaptive ε-test algorithm for $\mathcal{S}_m$ using $O(\varepsilon^{-1} m \log \varepsilon^{-1} \log m)$ queries for every $m \geq 2$. Note that for every fixed constant ε this bound is tight, since in Section 10 we show

an $\Omega(m \log m)$ lower bound. The bound is tight for every fixed constant m as well, since we have a corresponding non-adaptive lower bound of $\Omega(\varepsilon^{-1} \log \varepsilon^{-1})$.

Let $\varepsilon > 0$ and $m \geq 2$. The algorithm looks for a set A for of size at least $m+1$ whose elements are fully distinguishable using queries. The algorithm is defined for ε that is a power of 2 (for other choices of ε, we can use $\hat{\varepsilon} = 2^{-\lceil \log_2 \varepsilon^{-1} \rceil}$ instead).

At first, the algorithm chooses $I_0 \subseteq I_1 \subseteq \cdots \subseteq I_{\log \varepsilon^{-1}} \subseteq \{1, \ldots, n\}$, where I_a consists of $\lceil 2^{a+2} \log(m+1) \rceil$ indices drawn uniformly and independently.

The algorithm takes $1 + 32\varepsilon^{-1}m$ samples. Except for the first sample, they are partitioned into $2m$ "blocks" of at most $16\varepsilon^{-1}$ samples each. For every $1 \leq k \leq 2m$ and $0 \leq a \leq \log \varepsilon^{-1}$, the algorithm takes a sequence $S_{a,k}$ of $2^{3-a}\varepsilon^{-1}$ new samples, and queries every sample in it at the indices of I_a.

The algorithm rejects if there exists a *distinguishable composition* of size $m+1$ (which in particular is also a witness against $\mathcal{S}_m$). We formally define this term (and others) below.

Definition 7.1 (K-composition)**.** For $K \subseteq \{1, \ldots, 2m\}$, a sequence $A = (u_1, a_2, u_2, \ldots, a_\ell, u_\ell)$ is called a *K-composition of length ℓ* if $u_1 = u$ (the first sample) and for every $2 \leq i \leq \ell$, $0 \leq a_i \leq \log \varepsilon^{-1}$ and $u_i \in S_{a_i,k_i}$ (for some $k_i \in K$). A $\{1, \ldots, 2m\}$-composition is called a *composition*.

Definition 7.2 (Soundness of a K-composition)**.** For some K, let $A = (u_1, a_2, u_2, \ldots, a_\ell, u_\ell)$ be a K-composition. We say that it is *sound*, if for every $1 \leq i < j \leq \ell$, $d(u_i, u_j) > 2^{-a_j - 1}$.

In other words, a composition is *sound* if for every i, the choice of a_i results in lower bound for the distance of u_i from all $\{u_1, \ldots, u_{i-1}\}$. Note that the algorithm cannot be certain about the soundness of a K-composition unless it makes $\Omega(\varepsilon n)$ queries for every individual element in the composition (which it does not).

Definition 7.3 (Monotonicity of a K-composition)**.** For some K, let $A = (u_1, a_2, u_2, \ldots, a_\ell, u_\ell)$ be a K-composition. We say that it is *monotone* if $a_2 \geq \ldots \geq a_\ell$.

Observe that the algorithm can easily verify the monotonicity of a K-composition.

Definition 7.4 (Distinguishability of a K-composition)**.** For some K, let $A = (u_1, a_2, u_2, \ldots, a_\ell, u_\ell)$ be a K-composition. We say that it is *distinguishable* if, for every $1 \leq i_1 < i_2 \leq \ell$, there exists a query $j \in I_{a_{i_1}} \cap I_{a_{i_2}}$ for which $(u_{i_1})_j \neq (u_{i_2})_j$. For this definition, we set $a_1 = \log \varepsilon^{-1}$ (since $u_1 = u$, and the algorithm queries u at $I_{\log \varepsilon^{-1}}$).

In other words, a sequence is distinguishable if for every two samples in the composition, there exists a common query that distinguishes them. Observe that the algorithm can always verify the distinguishability of a K-composition.

Definition 7.5 (Valid composition)**.** For some K, a K-composition is *valid* if it is both sound and monotone.

Definition 7.6 (Rank of a monotone composition)**.** Let $A = (u_1, a_2, u_2, \ldots, a_\ell, u_\ell)$ be a monotone K-composition (for some K). Its *rank* is defined as $\vec{r}(A) = (a_2, \ldots, a_\ell)$.

Definition 7.7 (Order of ranks)**.** Ranks are ordered lexicographically as strings in $\{0, \ldots, \log \varepsilon^{-1}\}^*$. In particular, if $\vec{r}(A_1)$ is a proper prefix of $\vec{r}(A_2)$, then $\vec{r}(A_1) < \vec{r}(A_2)$. That is, the "end-of-string" virtual character is considered smaller than every actual character.

Algorithm 1 Non-adaptive construction of a valid composition

choose indices $i_1, \ldots, i_{\lceil 4\varepsilon^{-1}\log(m+1)\rceil}$ uniformly and independently, with repetitions.
for $0 \le a \le \log \varepsilon^{-1}$ **do**
  let $I_a = \{i_1, \ldots, i_{\lceil 2^{a+2}\log(m+1)\rceil}\}$.
take a sample u.
query u at $I_{\log \varepsilon^{-1}}$.
for k **from** 1 **to** $2m$ **do**
  for a **from** 0 **to** $\log \varepsilon^{-1}$ **do**
    take $2^{3-a}\varepsilon^{-1}$ new samples, denoting the sequence by $S_{a,k}$.
    query all samples in $S_{a,k}$ at I_a.
if there exists a distinguishable composition of size $m+1$ **then**
  return REJECT
else
  return ACCEPT

Observation 7.8. *For every composition A of length ℓ there exists some $K \subseteq \{1, \ldots, 2m\}$ of size $\ell - 1$ for which A is a K-composition.*

Definition 7.9 (Bad events)**.** For every $1 \le \ell \le m$ and for every $K \subseteq \{1, \ldots, 2m\}$ of size ℓ, let $B_{K,\ell}$ be the following event: considering the maximum rank $\vec{r}$ of a valid K-composition $A = (u_1, a_2, u_2, \ldots, a_\ell, u_\ell)$ of length ℓ (for some $a_2, \ldots, a_\ell$), there is no valid composition $\tilde{A}$ (not necessarily a K-composition) with $\vec{r}(\tilde{A}) > \vec{r}(A)$ of size $2 \le \ell' \le \ell + 1$.

Informally, $B_{K,\ell}$ is the event that there exist some valid K-composition whose length is too short and yet has the maximal rank among all valid compositions.

Lemma 7.10. *Assume that for every $1 \le \ell \le m$ and for every $K \subseteq \{1, \ldots, 2m\}$ of size ℓ, $B_{K,\ell}$ does not happen. Then there exists a valid composition of size $m+1$ or more.*

Proof. Let A be a valid composition with the maximal rank among all valid compositions (A is not necessarily unique). Let $K \subseteq \{1, \ldots, 2m\}$ be the set of blocks that contain the elements of A ($|K| \le |A|$). If $|A| \le m$, then by maximality, $B_{K,|A|}$ must have happened. Since we assumed that it has not, the length of $|A|$ must be at least $m+1$. □

Lemma 7.11. *If the input distribution is ε-far from $\mathcal{S}_m$, then for every $1 \le \ell \le m$ and $K \subseteq \{1, \ldots, 2m\}$ of size ℓ, $\Pr[B_{K,\ell}] \le e^{-4m}$.*

Proof. There are at least $2m - |K| \ge m$ blocks that are entirely free from the conditions on A. For every $k \in \{1, \ldots, 2m\} \setminus K$, and for every $0 \le a \le \log \varepsilon^{-1}$, the expected number of samples in $S_{a,k}$ that are 2^{-a-1}-far from all elements A is at least

$$\Pr\left[2^{-a-1} < d(x, A) \le 2^{-a}\right] \cdot 2^{3-a}\varepsilon^{-1} \ge 2^a c_a^A \cdot 8 \cdot 2^{-a}\varepsilon^{-1} = 8\varepsilon^{-1} c_a^A$$

Where c_a^A is a short notation for $\mathrm{Ct}[d(x, A) | 2^{-a-1} < d(x, A) \le 2^{-a}]$.

Considering all possible values for a in the same block, the expected number of these "matches" is at least

$$\sum_{a=0}^{\log \varepsilon^{-1}} 8\varepsilon^{-1} c_a^A = 8\varepsilon^{-1} \operatorname{Ct}\left[d(x,A) \middle| d(x,A) > \frac{1}{2}\varepsilon\right] \geq 8\varepsilon^{-1}\left(\operatorname{E}[d(x,A)] - \frac{1}{2}\varepsilon\right) > 8\varepsilon^{-1} \cdot \frac{1}{2}\varepsilon = 4$$

Considering all $k \in \{1,\ldots,2m\} \setminus K$ as well (at least m of them), the expected number of these matches is at least $4m$. This is a sum of independent binomial variables, hence by Lemma A.2 the probability that there are no matches at all is bounded by e^{-4m}. That is, with probability at least $1 - e^{-4m}$, there exist $a \in \{0,\ldots,\log \varepsilon^{-1}\}$, $k \in \{1,\ldots,2m\} \setminus K$ and $v \in S_{a,k}$ for which $d(v, \{u_1,\ldots,u_\ell\}) > 2^{-a-1}$. Consider the valid composition: $\tilde{A} = (u_1, a_2, u_2, \ldots, a_{i_0}, u_{i_0}, a, v)$, where $i_0 = \max(\{1\} \cup \{i | a_i \geq a\})$ (possibly $i_0 = 1$, and in this case $\tilde{A} = (u_1, a, v)$). Comparing the ranks,

$$\vec{r}(A) = (a_2, \ldots, a_{i_0}, \mathbf{a_{i_0+1}}, \ldots) < (a_2, \ldots, a_{i_0}, \mathbf{a}) = \vec{r}(\tilde{A})$$

Note that possibly also $i_0 = \ell$, and in this case a_{i_0+1} is the virtual "end-of-string" character that is defined as smaller than every value of $\mathbf{a}$. Hence in all cases $\vec{r}(\tilde{A}) > \vec{r}(A)$ as desired. □

Theorem 7.12. *Algorithm 1 is a one-sided ε-test of $\mathcal{S}_m$ that makes $O(\varepsilon^{-1} \log \varepsilon^{-1} \cdot m \log m)$ queries.*

Proof. For the query complexity, note that for every $1 \leq k \leq 2m$ and for every $0 \leq a \leq \log \varepsilon^{-1}$, the algorithm makes $\lceil 2^{3-a}\varepsilon^{-1} \rceil \cdot \lceil 2^{a+2} \log(m+1) \rceil = O(\varepsilon^{-1} \log m)$ queries inside $S_{a,k}$. Since the number of pairs of (a,k) is $O(m \log \varepsilon^{-1})$, the query complexity is $O(\varepsilon^{-1} \log \varepsilon^{-1} \cdot m \log m)$.

Perfect completeness is trivial, since a distinguishable composition of length $m+1$ is in particular an explicit witness against $\mathcal{S}_m$. For soundness, recall Lemma 7.10. If none of the bad events $B_{K,\ell}$ happens, then there exists a valid composition A of size $m+1$. If the input is ε-far from $\mathcal{S}_m$, then by the union bound (of the complement events), the probability for this is at least

$$1 - \sum_{\ell=1}^{m} \cdot \binom{2m}{\ell} \cdot e^{-4m} \geq 1 - 2^{2m} e^{-4m} > \frac{99}{100} \text{ for } m \geq 2$$

Hence, with probability $\frac{99}{100}$, there exists a valid composition A of size $m+1$.

Assume that this happens with some valid composition $A = (u_1, a_2, u_2, \ldots, a_{m+1}, u_{m+1})$. Set $a_1 = \log \varepsilon^{-1}$ for convenience.

By the constraints of A, for every $1 \leq i < j \leq m+1$:

- $d(u_i, u_j) > 2^{-a_j - 1}$.
- u_i is queried in $I_{a_i} \supseteq I_{a_j}$.
- u_j is queried in I_{a_j}.

The probability that I_{a_j} distinguishes u_i and u_j is at least $1 - (1 - 2^{-a_j-1})^{\lceil 2^{a_j+2} \log(m+1) \rceil} > 1 - e^{-2\log(m+1)}$. The probability that this happens for every $1 \leq i < j \leq m+1$ is bounded by $1 - \binom{m+1}{2} e^{-2\log(m+1)}$, which is at least $\frac{5}{6}$ (for $m \geq 2$). Overall, with probability at least $\frac{99}{100} \cdot \frac{5}{6} > \frac{2}{3}$, there exists a valid composition A of size $m+1$ which is also distinguishable, and in this case the algorithm rejects. □

8 The fishing expedition paradigm

We construct and prove here the algorithm for the fishing expedition paradigm, Lemma 2.6.

The algorithm has three parameters: a threshold p, a confidence q and a goal $k \geq 1$. The input is a subroutine $\mathcal{A}$ with diminishing returns and fail stability. Informally, the goal of the algorithm is to have k successful executions of $\mathcal{A}$, but also to terminate earlier if the probability of $\mathcal{A}$ to succeed becomes lower than p. Since the algorithm has no actual access to the success probability of $\mathcal{A}$, it should terminate early only if it is confident enough that the success probability of further executions is too low for them to be effective.

In this section we construct the fishing expedition paradigm providing Lemma 2.6. The algorithm providing it is Algorithm 2. The observations and lemmas below show that this algorithm satisfies the corresponding components of Lemma 2.6.

Algorithm 2 repeatedly executes $\mathcal{A}$. Of course, if $\mathcal{A}$ was successful for the k-th time, the algorithm terminates immediately. At some predefined check points (which are determined by p, q and k), the algorithm considers an early termination. Concretely, for $t_{\max} = \lfloor \log k + 1 \rfloor$ and for every $2 \leq t \leq t_{\max}$, after $\left\lceil p^{-1} \cdot \max\{2^t, 5(\log q^{-1} + \log(\log k + 2))\} \right\rceil$ executions of $\mathcal{A}$, the algorithm terminates if the number of successful executions was less than a $\frac{1}{2}p$-portion of the total number of executions. The algorithm must terminate in one of these iterations, as stated in the following observation.

Algorithm 2 Fishing expedition

parameters $k \geq 1$ (goal), $p > 0$ (threshold), $q > 0$ (confidence).
input A subroutine $\mathcal{A}$ with output, given as a black box, where the output "0" means FAIL.
let $t_{\max} \leftarrow \lfloor \log k + 1 \rfloor$.
let $N_1 \leftarrow 0$.
set $H \leftarrow 0$.
for t **from** 2 **to** $t_{\max}$ **do**
  let $N_t \leftarrow \left\lceil p^{-1} \max\{2^t, 5(\log q^{-1} + \log(\log k + 1))\} \right\rceil$.
  for N **from** $N_{t-1} + 1$ **to** N_t ▷ possibly empty **do**
    run $\mathcal{A}$, let R_N be its outcome.
    let X_N be an indicator for success ($X_N = 1$ if $R_N \in G$, otherwise $X_N = 0$).
    set $H \leftarrow H + X_N$.
    if $H = k$ **then terminate** with N. ▷ goal is reached
  if $H < \frac{1}{2}pN_t$ **then**
    terminate with N_t. ▷ continuing is ineffective
unreachable point ▷ Observation 8.1.

Observation 8.1. *If Algorithm 2 has not terminated by the $t_{\max}$-th iteration, it must do so there.*

Proof. Note that $N_{t_{\max}} = N_{\lfloor \log k+1 \rfloor} \geq \left\lceil p^{-1} 2^{\lfloor \log k+2 \rfloor} \right\rceil \geq p^{-1} 2^{\log k+1} = 2p^{-1}k$. That is, at the end of the $t_{\max}$-th iteration, if $H < k$ then $H < \frac{1}{2}pN_{t_{\max}}$ and the iteration must terminate. □

For every $N \geq 0$, let H_N be the value of H after the N-th execution (that is, $H_N = \sum_{i=1}^{N} X_i$).

Lemma 8.2. *Algorithm 2 always terminates with* $N \leq p^{-1}(4H + 5(\log q^{-1} + \log(\log k + 1))) + 1$.

Proof. If the algorithm terminates in the first iteration ($t = 2$), then

$$N \leq N_2 = \left\lceil p^{-1} \max\{2^2, 5(\log q^{-1} + \log(\log k + 2))\} \right\rceil \leq 5p^{-1}(\log q^{-1} + \log(\log k + 2)) + 1$$

In terminations outside the first iteration, observe that $N_t \leq 2N_{t-1}$. Since the algorithm did not terminate in the previous iteration, $H_{N_{t-1}} \geq \frac{1}{2}pN_{t-1} \geq \frac{1}{4}pN_t$. Since $H_{N_t} \geq H_{N_{t-1}}$ as well, we have $H_{N_t} \geq \frac{1}{4}pN_t$ and thus $N_t \leq 4p^{-1}H_{N_t}$.

By Observation 8.1, the algorithm must have terminated in one of the iterations. This completes the proof. □

Lemma 8.3. *Let* $p > 0$, $q > 0$, $k \geq 1$, *and let* $\mathcal{A}$ *be a subroutine with diminishing returns and fail stability. Let* N *be the number of executions of* $\mathcal{A}$ *done by Algorithm 2 and let* H *be the number of successful executions. Let* $\hat{p} = \Pr[X_{N_t+1} = 1|R_1, \ldots, R_{N_t}]$ *be the probability that an additional, hypothetical execution of* $\mathcal{A}$ *is successful (note that* $\hat{p}$ *is a random variable that depends on the outcomes of the* N *executions of* $\mathcal{A}$*). In this setting, with probability higher than* $1 - q$, $H = k$ *or* $\hat{p} \leq p$ *(or both).*

Proof. Consider the following equivalent algorithm: we simulate Algorithm 2, but ignore the termination requests. When reaching what was the "unreachable point", we stop and choose the output (N, H) according to the first termination request. Clearly, this algorithm always returns the same N, H as would be in a run of Algorithm 2 for the same outcome sequence, but it is non-adaptive (more precisely, it always makes $N_{t_{\max}}$ executions of $\mathcal{A}$ regardless of their outcomes and then chooses the output).

For every $0 \leq n \leq N_{t_{\max}}$, let $H_n = \sum_{i=1}^{n} X_i$. For every $2 \leq t \leq t_{\max}$, let $\mathcal{B}_t$ be the following bad event: $(H_{N_t} < k) \wedge (H_{N_t} < \frac{1}{2}pN_t) \wedge (\Pr[X_{N+1} = 1|R_1, \ldots, R_N] \geq p)$. Note that if no bad event happens, then the output satisfies $H \geq k$ or $\Pr[X_{N_t+1} = 1|R_1, \ldots, R_{N_t}] < p$. We will use a variant of Chernoff's bound that is proved in the appendix (Lemma A.5) to bound the probabilities of the bad events.

Let $\mathcal{G} = \{(R_1, \ldots, R_N) \in \mathrm{supp}(R_1, \ldots, R_N) : |\{R_i \neq 0\}| \geq k \vee \Pr[R_{N+1} \neq 0|R_1, \ldots, R_N] < p\}$ be the set of outcome sequences where a termination would be justifiable (note that it is not exactly the same as the set of termination conditions). Note that if $(R_1, \ldots, R_{N-1}) \in \mathcal{G}$, then $(R_1, \ldots, R_N) \in \mathcal{G}$ as well, since the number of non-zero elements cannot decrease and the probability of the next trial cannot increase.

For every $2 \leq t \leq t_{\max}$, by Lemma A.5, for $\delta = \frac{1}{2}$, $m = N_t$ and $X = H_{N_t}$

$$\begin{aligned}
\Pr[\mathcal{B}_t] &= \Pr\left[(H_{N_t} < k) \wedge (H_{N_t} < \frac{1}{2}pN_t) \wedge (\Pr\left[R_{N_t+1} \notin G|R_1, \ldots, R_{N_t}\right] \geq p\right] \\
&= \Pr\left[((R_1, \ldots, R_{N_t}) \notin \mathcal{G}) \wedge (H_{N_t} < \frac{1}{2}pN_t)\right] \\
&< (\sqrt{2/e})^{pN_t} \leq 0.86^{pN_t} \leq 0.86^{5(\log q^{-1} + \log(\log k + 2))} < 0.5^{\log q^{-1} + \log(\log k + 2)} = \frac{q}{\log k + 2}
\end{aligned}$$

By the union bound, $\Pr[\bigvee_{t=2}^{t_{\max}} \mathcal{B}_t] < \lfloor \log k + 2 \rfloor \cdot \frac{q}{\log k+2} \leq q$. With probability greater than $1-q$, no bad event happens, and the algorithm terminates with $H \geq k$ or $\Pr[R_{N+1} \notin G | R_1, \ldots, R_N] < p$. □

At this point we can prove the fishing expedition lemma.

Proof of Lemma 2.6. This lemma follows immediately from Lemma 8.2 (number of executions) and Lemma 8.3 (the algorithm reaches one of its goals with probability at least $1-q$). □

9 Adaptive m-support test

We construct here an adaptive one-sided error algorithm using $O(\varepsilon^{-1} m \log m \cdot \min\{\log \varepsilon^{-1}, \log m\})$ many queries.

The advantage of being adaptive The non-adaptive algorithm considers $\Omega(\log \varepsilon^{-1})$ buckets of distance ranges at the cost of $\Omega(\varepsilon^{-1} \cdot m \log m)$ queries per bucket, and we believe that we cannot do much better (the $\Omega(\log \varepsilon^{-1})$ buckets are required according to the non-adaptive lower bound, and the $\Omega(\varepsilon^{-1} m \log m)$ queries per bucket are required according to the adaptive lower-bound in Section 10). For $\varepsilon < \frac{1}{m}$, the number of distance buckets is more than $\log m + 1$.

An adaptive algorithm can do better. Initially, we consider $\log m + 1$ distance buckets. Then, using a decision tree constructed incrementally as more distinct elements are found, we avoid the need to consider the rest of the buckets. In particular, the "far buckets" phase is the bottleneck of the algorithm ($\Theta(m^3 \log m + \varepsilon^{-1} m \log^2 m)$ queries), where the second phase is extremely cheap: $O(\varepsilon^{-1} m)$ queries, below the lower bound for adaptive algorithms (see Section 10). This means that further improvements of the query complexity must address the first phase (which only considers distinct elements that are $\frac{1}{2m}$-far from each other). We later get rid of the $\Theta(m^3 \log m)$ term by using the non-adaptive algorithm when ε is too large.

9.1 Additional building blocks

The "fishing expedition" paradigm (Algorithm 2) is an important building block in our algorithm. Here we define some additional algorithmic building blocks.

Building Block 9.1 (Using a decision tree)**.** Let $A = \{x_1, \ldots, x_k\}$ be a set of k distinct strings, and let $\mathcal{T}$ be a query-based decision tree with exactly k leaves, such that every string in A corresponds to a different leaf in $\mathcal{T}$. For every input string x, we can algorithmically find an $1 \leq i \leq k$ such that $\mathcal{T}(x) = \mathcal{T}(x_i)$. The number of queries made by this procedure is bounded by the height of $\mathcal{T}$.

Proof. Trivially, start at the root and follow the path according to the queries and the answers of x to these queries. At some point we reach a leaf. This leaf must correspond to some $x_i \in A$, since $\mathcal{T}$ has exactly k leaves and they fully distinguish the k elements in A. □

Building Block 9.2 (Updating a decision tree)**.** Let $\mathcal{T}$ be a decision tree with k leaves that fully distinguishes $A = \{x_1, \ldots, x_k\}$. Given a string x, an i for which $\mathcal{T}(x) = \mathcal{T}(x_i)$, and an index j for which $x|_j \neq x_i|_j$, we can algorithmically construct a decision tree $\mathcal{T}'$ with $k+1$ leaves that fully distinguishes $A' = \{x_1, \ldots, x_k, x\}$, at the cost of no additional queries.

Proof. Just substitute the leaf $\mathcal{T}(x_i)$ by an internal node consisting of the query j and two children x and x_i. □

Building Block 9.3 (Construction of a decision tree)**.** Let $A = \{x_1, \ldots, x_k\}$ be a distinguishable set of strings, that is, for every $1 \leq i_1 < i_2 \leq k$, there exists an index j at which both x_{i_1}, x_{i_2} were queried and $(x_{i_1})_j \neq (x_{i_2})_j$. We can construct a decision tree $\mathcal{T}$ with exactly k leaves that distinguishes all $x_1, \ldots, x_k$, at the cost of at most k^2 queries. Moreover, at the end of the construction, for every $1 \leq i \leq k$, every x_i was queried at all indices in the search path of x_i in $\mathcal{T}$.

Proof. For $k = 1$ it is trivial. For $k > 1$, consider $A' = \{x_1, \ldots, x_{k-1}\}$ and construct a decision tree $\mathcal{T}'$ with $k-1$ leaves distinguishing $x_1, \ldots, x_{k-1}$ at the cost of at most $(k-1)^2$ queries. Use Building Block 9.1 (using a decision tree) to find $1 \leq i \leq k-1$ for which $\mathcal{T}'(x_k) = \mathcal{T}'(x_i)$, at the cost of at most $k-1$ queries (note that these additional queries are only done in x_k). According to the statement, there exists some index j at which both x_i and x_k were queried, and $(x_i)_j \neq (x_k)_j$. Use Building Block 9.2 (updating a decision tree) to insert x_k to the tree at the cost of no additional queries.

We used at most $(k-1)^2$ queries to construct $\mathcal{T}'$ and at most $k-1$ additional queries to insert x_k to it. The total number of queries is at most k^2, as required. □

9.2 The algorithm

If $\varepsilon \geq \frac{1}{m^2}$ then we just execute Algorithm 1. The query complexity of the algorithm is $O(\varepsilon^{-1} \log \varepsilon^{-1} \cdot m \log m)$ in this case, and it is the same as $O(\varepsilon^{-1} m \log m \cdot \min\{\log \varepsilon^{-1}, \log m\})$ since $\log \varepsilon^{-1} \leq \log m^2 \leq 2 \log m = O(\log m)$. If $\varepsilon < \frac{1}{m^2}$, then we use the adaptive algorithm below.

For $\varepsilon < \frac{1}{m^2}$, the algorithm consists of two phases: the first one is intended to find distinct samples that are $\frac{1}{2m}$-far from each other, and the second one uses a decision tree to reduce the number of queries required to find additional distinct elements that are $\frac{1}{2m}$-close to those already found.

Batches Assume that we know (or guess) that $\Pr[d(x, A) > \alpha] > \frac{\alpha^{-1}\varepsilon}{\log m}$. If we draw $\alpha\varepsilon^{-1} \log m$ samples, then with high probability there is a sample Y that is α-far from A. In this case, a set of $2^a \log m$ indices should distinguish Y from all $X \in A$. That is, under this assumption, we can find an additional distinct element with probability greater than some global positive constant. This subroutine is called a *batch*.

Concretely, every a-batch chooses a set J of $\lceil 2^{a+2} \log m \rceil$ indices (uniformly and independently) and queries all samples in A at the indices of J. Then it draws additional $\lceil 2^{2-a}\varepsilon^{-1} \log m \rceil$ samples and queries all of them at the indices of J. If there exists a sample Y for which $Y|_J \neq X|_J$ for every $X \in A$, the batch is considered successful, and we add Y to A.

Observation 9.4. *Algorithm 3a has diminishing returns and fail stability as per Definition 2.4 and Definition 2.5, where for formality's sake we use some fixed mapping of the set of possible non-failing output values to distinct positive natural numbers.*

Lemma 9.5. *Algorithm 3a uses $O(m^2 \log m + \varepsilon^{-1} \log^2 m)$ queries, and if $\Pr[d(x, A) > 2^{-a-1}] > \frac{2^a \varepsilon}{4 \log m}$, then the success probability of Algorithm 3a is at least $\frac{1}{3}$.*

Algorithm 3a Adaptive one-sided ε-test for $\mathcal{S}_m$, a single batch

parameters $\varepsilon > 0$, A, $m \geq 2$, $0 \leq a \leq \lceil \log m \rceil$ where $|A| \leq m$.
input A distribution P.
choose a set J of $\left\lceil 2^{a+2} \log m \right\rceil$ indices uniformly and independently.
query X at J for every $X \in A$.
take $\left\lceil 2^{2-a} \varepsilon^{-1} \log m \right\rceil$ samples.
query each new sample at J.
if there exists a sample Y for which $Y|_J \neq X_J$ for every $X \in A$ **then**
    set $A \leftarrow A \cup \{Y\}$.
    return SUCCESS with (Y, J).
else
    return FAIL

Proof. The query complexity of Algorithm 3a is $(|A| + \left\lceil 2^{-a} \varepsilon^{-1} \log m \right\rceil)|J| \leq (m + 2^{-a} \varepsilon^{-1} \log m + 1)(2^a \log m + 1) = O(m^2 \log m + \varepsilon^{-1} \log^2 m)$ (since $a \leq \log m + 1$).

If $\Pr[d(x, A) > 2^{-a-1}] > \frac{2^a \varepsilon}{4 \log m}$, then the expected number of samples 2^{-a-1}-far from A within the new $\left\lceil 2^{2-a} \varepsilon^{-1} \log m \right\rceil$ samples is at least 1. Hence the probability that there is at least one them is at least $1 - e^{-1} > \frac{3}{5}$ (by Lemma A.2).

If this happens, let Y be such a sample. With probability at least $1 - m(1 - 2^{-a-1})^{2^{a+2} \log m} > \frac{7}{10}$, $Y|_J \neq X|_J$ for every $X \in A$. Overall, the probability to have a sample Y for which $Y|_J \neq X|_J$ for all $X \in A$ is at least $\frac{3}{5} \cdot \frac{7}{10} > \frac{1}{3}$. □

The first phase The batches are not standalone, since they must have some parameter a. The first phase of the algorithm consists of $\lceil \log m \rceil + 1$ iterations. For every $0 \leq a \leq \lceil \log m \rceil$, the a-th iteration consists of batches with parameter a. To make sure that the batches are only executed when it is cost-effective, we use the "fishing expedition" paradigm (Algorithm 2).

Algorithm 3b Adaptive one-sided ε-test for $\mathcal{S}_m$, first phase

parameters $\varepsilon > 0$, $m \geq 2$.
input A distribution P, a set $A \subseteq \mathrm{supp}(P)$ of distinguishable elements.
for a **from** 0 **to** $\lceil \log m \rceil$ **do**
    let $k_a = m + 1 - |A|$.
    run Algorithm 2 ("fishing expedition") with parameters $k = k_a$, $q = \frac{1}{4\lceil \log m + 1 \rceil}$, $p = \frac{1}{3}$,
        input $\mathcal{A}$ = Algorithm 3a (a single batch).
    if $|A| \geq m + 1$ **then**
        return REJECT
Proceed to the second phase with A.

Lemma 9.6. *Algorithm 3b makes $O(m^3 \log m + \varepsilon^{-1} m \log^2 m)$ queries. With probability at least $\frac{3}{4}$, either $|A| \geq m + 1$ or $\mathrm{Ct}[d(x, A) | d(x, A) > \frac{1}{2m}] \leq \frac{1}{2}\varepsilon$.*

Proof. For every $0 \leq a \leq \lceil \log m \rceil$, let H_a be the number of successful executions of Algorithm 3a within the a-th iteration. Also, let N_a be the total number of executions. Lemma 8.2 guarantees

that

$$N_a \leq 3\left(4H_a + 5\left(\log\left(4\lceil \log m + 1\rceil\right) + \log\left(\log m + 2\right)\right)\right) + 1 \leq 12H_a + O(\log m)$$

Note that $\sum_{a=0}^{\lceil \log m\rceil} H_a = |A| - 1$. Either $|A| \geq m+1$ or $\sum_{a=0}^{\lceil \log m\rceil} H_a < m$. Considering all iterations,

$$\begin{aligned}\sum_{a=0}^{\lceil \log m\rceil} N_a &\leq 12 \sum_{a=0}^{\lceil \log m\rceil} H_a + (\lceil \log m\rceil + 1)O(\log m) \\ &< 12m + O(\log^2 m) = O(m)\end{aligned}$$

Every iteration makes at most $O(m^2 \log m + \varepsilon^{-1}\log^2 m)$ queries, hence the total number of queries is $O(m^3 \log m + \varepsilon^{-1} m \log^2 m)$.

For every $0 \leq a \leq \lceil \log m\rceil$, by Lemma 8.3, with probability at least $1 - \frac{1}{4\lceil \log m\rceil}$, either $H_a = k_a$ or the probability to find an additional distinct element is less than $\frac{1}{3}$. In the first case, $|A| = m+1$ at the end of the iteration (since $k_a = m + 1 - |A|$, considering the size of A at the beginning of the iteration). In the second case, $\Pr[d(x,A) > 2^{-a-1}] \leq \frac{2^a \varepsilon}{4 \log m}$ (by Lemma 9.5).

With probability at least $1 - \frac{\lceil \log m\rceil + 1}{4\lceil \log m + 1\rceil} = \frac{3}{4}$, this happens for all values of a. That is, either $|A| \geq m+1$ (once) or $\Pr[d(x,A) > 2^{-a-1}] \leq \frac{2^a\varepsilon}{4\log m}$ for all $0 \leq a \leq \lceil \log m\rceil$. In the first case we reject, and in the second case,

$$\begin{aligned}\mathrm{Ct}\left[d(x,A) | d(x,A) > \frac{1}{2m}\right] &\leq \sum_{a=0}^{\lceil \log m\rceil} 2^{-a} \Pr[2^{-a-1} < d(x,A) \leq 2^{-a}] \\ &\leq \sum_{a=0}^{\lceil \log m\rceil} 2^{-a} \Pr[d(x,A) > 2^{-a-1}] \\ &\leq \sum_{a=0}^{\lceil \log m\rceil} 2^{-a} \cdot \frac{2^a \varepsilon}{4\log m} = \frac{\lceil \log m\rceil + 1}{4\log m}\varepsilon \leq \frac{1}{2}\varepsilon\end{aligned}$$

□

The second phase The second phase of the algorithm handles the case where $|A| \leq m$ and also $\mathrm{Ct}[d(x,A) | d(x,A) \geq \frac{1}{2m}] \leq \frac{1}{2}\varepsilon$. If the input distribution is ε-far from being supported by any subset of A, the contribution of "small distances", $\mathrm{Ct}[d(x,A) | d(x,A) < \frac{1}{2m}]$, is strictly greater than $\frac{1}{2}\varepsilon$.

First, we construct a decision tree $\mathcal{T}$ for the elements in A, according to Building Block 9.3, at the cost of at most m^2 queries. After the decision tree is constructed, the algorithm is iterative. It tracks a set $A = \{X_0, \ldots, X_{|A|-1}\}$ of distinguishable samples (initialized with the A supplied by the first phase) and a decision tree $\mathcal{T}$ with $|A|$ leaves corresponding to A's elements.

In every iteration, the algorithm draws a new sample $Y \sim P$ and executes the decision tree $\mathcal{T}$ on Y, resulting in an index $0 \leq i \leq |A| - 1$ for which $\mathcal{T}(Y) = \mathcal{T}(X_i)$. Then it queries both X_i and Y at a brand new query set J of size m. If Y is $\frac{1}{2m}$-close to A, then with high probability (proportionally to the distance), $Y|_J \neq X_i|_J$. If this happens, then we add Y to A and to the decision tree.

Algorithm 3c Adaptive one-sided ε-test for $\mathcal{S}_m$, a single iteration of the second phase

input A sample $Y \in \text{supp}(P)$, $A \subseteq \text{supp}(P)$, a decision tree $\mathcal{T}$; $|A| \geq 1$.
invariant $\mathcal{T}$ has $|A|$ leaves corresponding to A's elements.
choose a set J of m indices uniformly, independently, with repetitions.
let $X \in A$ for which $\mathcal{T}(Y) = \mathcal{T}(X)$. ▷ Building Block 9.1
query X, Y at J.
if $Y|_J \neq X|_J$ **then**
  set $A \leftarrow A \cup \{Y\}$.
  add Y to $\mathcal{T}$. ▷ Building Block 9.2, no extra queries

Lemma 9.7. *Algorithm 3c makes at most $3m$ queries and keeps its invariants. If $|A| \leq m$ and the input distribution P is ε-far from $\mathcal{S}_m$, then the algorithm adds Y to A with probability at least $\frac{1}{4}m\varepsilon$.*

Proof. There exists some $\hat{A} \subseteq A$ of size at most m for which $\text{Ct}[d(x,\hat{A})|d(x,\hat{A}) > \frac{1}{2m}] \leq \frac{1}{2}\varepsilon$ ($\hat{A}$ is the output of the first, non-adaptive phase). Since A contains $\hat{A}$, $\text{Ct}[d(x,A)|d(x,A) > \frac{1}{2m}] \leq \frac{1}{2}\varepsilon$ as well. Since $|A| \leq m$, $\text{E}[d(x,A)] \geq d(P,\mathcal{S}_m) > \varepsilon$, and $\text{Ct}[d(x,A)|d(x,A) \leq \frac{1}{2m}] > \frac{1}{2}\varepsilon$.

For every $Y \in \text{supp}(P)$, consider $X(Y) \in A$ for which $\mathcal{T}(Y) = \mathcal{T}(X(Y))$. For a sample Y drawn from P, by Lemma A.1, The probability to distinguish between $X(Y)$ and Y is:

$$\begin{aligned}
\mathop{\text{E}}_{Y\sim P}\left[1-(1-d\left(Y,X(Y)\right))^{|J|}\right] &\geq \mathop{\text{Ct}}_{Y\sim P}\left[1-(1-d\left(Y,X(Y)\right))^{|J|}\middle| d(Y,A)\leq\frac{1}{2m}\right] \\
&\geq \mathop{\text{Ct}}_{Y\sim P}\left[1-(1-d(Y,A))^{m}\middle| d(Y,A)\leq\frac{1}{2m}\right] \\
&\geq \frac{1}{2}m\mathop{\text{Ct}}_{Y\sim P}\left[d(Y,A)\middle| d(Y,A)\leq\frac{1}{2m}\right] > \frac{1}{4}m\varepsilon
\end{aligned}$$

If the algorithm distinguishes between Y and $X(Y)$, then the invariant is kept by the constraints of Building Block 9.2. If the algorithm fails to distinguish between Y and $X(Y)$, then the invariant is kept trivially. □

Theorem 9.8. *Algorithm 3d is a one-sided ε-test of $\mathcal{S}_m$ using $O(\varepsilon^{-1}m\log m\cdot\min\{\log\varepsilon^{-1},\log m\})$ many queries.*

Proof. If $\varepsilon \geq \frac{1}{m^2}$ then the correctness is implied by Theorem 7.12, and the query complexity is $O(\varepsilon^{-1}\log\varepsilon^{-1}\cdot m\log m)$. Since $\varepsilon \geq \frac{1}{m^2}$, $\log\varepsilon^{-1} \leq 2\log m$ and the query complexity is bounded by $O(\varepsilon^{-1}m\log m\cdot\min\{\log\varepsilon^{-1},\log m\})$.

If $\varepsilon < \frac{1}{m^2}$, the query complexity of the first phase is $O(m^3\log m+\varepsilon^{-1}m\log^2 m)$ (Lemma 9.6). The query complexity of constructing $\mathcal{T}$ for the first time (between the phases) is m^2 queries, which is at most ε^{-1} since $\varepsilon < \frac{1}{m^2}$. The query complexity of the second phase is $O(\varepsilon^{-1})\cdot 3m = O(\varepsilon^{-1}m)$ (Lemma 9.7). Overall, the query complexity of the algorithm is $O(m^3\log m+\varepsilon^{-1}m\log^2 m)$. Since $m^2 < \varepsilon^{-1}$, $m^3 < \varepsilon^{-1}m$ and thus the query complexity is bounded by $O(\varepsilon^{-1}m\log^2 m)$. Since $\log m \leq \log\varepsilon^{-1}$, it is bounded by $O(\varepsilon^{-1}m\log m\cdot\min\{\log\varepsilon^{-1},\log m\})$ as well.

Algorithm 3d Adaptive one-sided ε-test for $\mathcal{S}_m$

input A distribution P.
if $\varepsilon \geq \frac{1}{m^2}$ **then**
  run Algorithm 1 and **return** its answer.
take the first sample u.
set $A \leftarrow \{u\}$.
run Algorithm 3b (possibly modifying A, possibly rejecting).
construct a decision tree $\mathcal{T}$ based on A. ▷ Building Block 9.3
invariant $\mathcal{T}$ has $|A|$ leaves corresponding to A's elements.
for $\lceil 48\varepsilon^{-1} \rceil$ **times do**
  draw another sample Y.
  run Algorithm 3c with $(Y, A, \mathcal{T})$ (note that A, $\mathcal{T}$ may have been modified).
  if $|A| \geq m+1$ **then**
    return REJECT
return ACCEPT

Perfect completeness is trivial, since the algorithm rejects only if $|A| \geq m+1$, where $|A|$ is a set of fully distinguishable samples.

For soundness, consider an input distribution P that is ε-far from $\mathcal{S}_m$.

By Lemma 9.6, with probability $\frac{3}{4}$, either $|A| \geq m+1$ or $\mathrm{Ct}\left[d(x,A) | d(x,A) > \frac{1}{2m}\right] \leq \frac{1}{2}\varepsilon$. In the first case the algorithm rejects immediately. Otherwise, we analyze the second phase.

Assume that the second phase of the algorithm had infinitely many iterations. By Lemma 9.7, as long as $|A| \leq m$, every iteration of the second phase extends A with probability at least $\frac{1}{4}m\varepsilon$. The number of iterations until A has $m+1$ elements is a sum of at most m geometric random variables, each having success probability at least $\frac{1}{4}m\varepsilon$. The expected number of iterations until A has $m+1$ elements is thus bounded by $m \cdot 4\varepsilon^{-1}/m = 4\varepsilon^{-1}$. By Markov's inequality, with probability at least $\frac{11}{12}$, this number of iterations is at most $48\varepsilon^{-1}$.

To conclude: with probability at least $\frac{3}{4}$, there exists a distinguishable set A for which $|A| \geq m+1$ or $\mathrm{Ct}[d(x,A) | d(x,A) > \frac{1}{2m}] \leq \frac{1}{2}\varepsilon$. In the first case the algorithm rejects the input immediately, and in the second case, with probability at least $\frac{11}{12}$, the algorithm rejects the input within the first $\lceil 48\varepsilon^{-1} \rceil$ samples of the second phase. Overall, the probability to reject an ε-far input is at least $\frac{3}{4} \cdot \frac{11}{12} > \frac{2}{3}$. □

10 Superlinear lower-bound for one-sided adaptive m-support test

10.1 Lower bound on the size of witnesses against $\mathcal{S}_m$

We show that every witness against $\mathcal{S}_m$ must be of size at least $\Omega(m \log m)$, hence every one-sided ε-test algorithm for $\mathcal{S}_m$ must use $\Omega(m \log m)$ queries as well. We use this to show an $\Omega(\varepsilon^{-1} m \log m)$ lower bound for non-adaptive algorithms, and after a short discussion we extend this result to adaptive algorithms as well.

Definition 10.1 (Capacity of an edge cover)**.** Let G be a graph over a set V vertices and let

$\mathcal{G} = (G_1, \ldots, G_k)$ be a sequence of graphs over $V_1, \ldots, V_k \subseteq V$ such that $G = \bigcup_{i=1}^{k} G_i$. We define the *capacity* of $\mathcal{G}$ as $\mathrm{cap}(\mathcal{G}) = \sum_{i=1}^{k} |V_k|$.

The following observation follows directly from the definition of capacity.

Observation 10.2. *Let P be a distribution over $\{0,1\}^n$, $x_1, \ldots, x_s \in \mathrm{supp}(P)$ be a set of samples and $Q \subseteq \{1, \ldots, s\} \times \{1, \ldots, n\}$ be a query set. Let $S_1, \ldots, S_n$ be the index-specific query sets, that is, $Q = \bigcup_{j=1}^{n}(S_j \times \{j\})$. In other words, for every j, all samples in S_j are queried at the index j. Let $\mathcal{G} = (G_1, \ldots, G_n)$ be the edge cover of the contradiction graph (Definition 2.7) implied by $(x_1, \ldots, x_s; Q)$: for every $1 \le j \le n$, G_j is the complete bipartite graph whose vertices are S_j and the sides are:*

$$L_j = \{i \in S_j | (x_i)_j = 0\}, \qquad R_j = \{i \in S_j | (x_i)_j = 1\}$$

In this setting, $\mathrm{cap}(\mathcal{G}) = |Q|$.

Lemma 10.3 ([Han64, KS67, Alo23])**.** *Let V be a set of vertices, and let $\mathcal{G} = (G_1, \ldots, G_k)$ be an edge cover of the V-clique such that all graphs $G_1, \ldots, G_k$ are bipartite. Then* $\mathrm{cap}(\mathcal{G}) \ge |V| \log_2 |V|$.

Lemma 10.4. *Let G be a graph over a set V of vertices that is not m-colorable, and let $\mathcal{G} = (G_1, \ldots, G_k)$ be an edge cover of G such that all graphs $G_1, \ldots, G_k$ are bipartite. Then* $\mathrm{cap}(\mathcal{G}) \ge (m+1)\log_2(m+1)$.

Proof. Without loss of generality we assume that G is exactly $m+1$-colorable (otherwise we can just omit vertices one by one until it is). Let $U_1, \ldots, U_{m+1}$ be a coloring of G using $m+1$ colors (that is, U_i is an independent set for every $1 \le i \le m+1$ and $\bigcup_{i=1}^{k} U_i = V$).

Let $\hat{G}$ be a graph over $\{1, \ldots, m+1\}$ such that the edge $\{i, j\}$ exists if and only if there is an edge between a vertex in U_i and a vertex in U_j. We define the edge cover $\hat{\mathcal{G}} = (\hat{G}_1, \ldots, \hat{G}_k)$ similarly: the vertex i belongs to $\hat{G}_j$ if some vertex in U_i belongs to G_j. Note that the sides are implied since all vertices of U_i must be on the same side. Note that $\mathrm{cap}(\hat{\mathcal{G}}) \le \mathrm{cap}(\mathcal{G})$, since every vertex in $\hat{\mathcal{G}}$ represents a (disjoint) set of vertices in $\mathcal{G}$.

Observe that $\hat{G}$ must be a clique: if an edge $\{i, j\}$ is missing, then $U_i \cup U_j$ is an independent set and hence G is m-colorable. Hence $\hat{\mathcal{G}}$ is an edge cover of $m+1$-clique using bipartite graphs, and by Lemma 10.3 its capacity must be at least $(m+1)\log_2(m+1)$. Hence $\mathrm{cap}(\mathcal{G}) \ge \mathrm{cap}(\hat{\mathcal{G}}) \ge (m+1)\log_2(m+1)$. □

Proposition 10.5. *Every witness against belonging to $\mathcal{S}_m$ must be at least $m \log_2 m$-bits long. In particular, every one-sided ε-testing algorithm for $\mathcal{S}_m$ must make at least $m \log_2 m$ queries.*

Proof. By Lemma 5.14, there exists a witness against $\mathcal{S}_m$ if and only if the contradiction graph is not m-colorable. By Lemma 10.4, the capacity of every bipartite cover of the contradiction graph is at least $(m+1)\log_2(m+1)$. By Observation 10.2, the number of queries must be at least $(m+1)\log_2(m+1)$ as well. □

10.2 An improved bound for non-adaptive algorithms

The lower bound on the size of a witness implies a trivial bound of $\Omega(m \log m)$ queries for every one-sided ε-test of $\mathcal{S}_m$ for any $0 < \varepsilon < 1$. To extend this result to a slightly better $\Omega(\varepsilon^{-1} m \log m)$ bound for non-adaptive adaptive algorithms, we first need the following almost-trivial observation.

Observation 10.6. *Let G be a graph over $V = V_0 \cup V_1$ (where $V_0 \cap V_1 = \emptyset$). If the subgraph of G induced by V_1 is k-colorable, then G is $(|V_0| + k)$-colorable.*

Based on this observation we can improve the $\Omega(m \log m)$ bound for non-adaptive algorithms.

Proposition 10.7. *Every one-sided non-adaptive ε-testing algorithm for $\mathcal{S}_m$ must make at least $\Omega(\varepsilon^{-1} m \log m)$ many queries.*

Proof. Consider the following D_{no} distribution: first, we choose $a_1, \ldots, a_{2m} \sim \{0,1\}^n$ uniformly and independently, and then we return

$$P \sim \begin{cases} 0 & \text{with probability } 1 - 10\varepsilon \\ a_i & \text{with probability } \frac{5\varepsilon}{m}, 1 \le i \le 2m \end{cases}$$

With probability $1 - o(1)$, all a_is are 0.49-far from each other and from the zero vector, hence the distance of P from $\mathcal{S}_m$ is at least $0.24 \cdot m \cdot \frac{5\varepsilon}{m} > \varepsilon$.

Let Q be a query set with $|Q| < \frac{1}{100}\varepsilon^{-1} m \log_2 m$ queries over s samples. For every $1 \le i \le s$, let q_i be the number of queries in the i-th sample. Note that $q = \sum_{i=1}^{s} q_i$, by definition. This is where we use the assumption that the algorithm is non-adaptive, since $q_1, \ldots, q_s$ may not be pre-determined for adaptive algorithms.

By linearity of expectation, the expected number of queries applied to non-zero samples is bounded by $10\varepsilon \sum_{i=1}^{s} q_i = 10\varepsilon \cdot |Q| < \frac{1}{10} m \log_2 m$. With probability higher than $\frac{4}{5}$, this number of queries is smaller than $\frac{1}{2} m \log_2 m$, and in this case, there cannot be a witness for m distinct elements among $a_1, \ldots, a_{2m}$. This means that, with the same probability, there is no witness for $m+1$ elements among $0, a_1, \ldots, a_{2m}$. The lower bound follows from Lemma 5.15. □

10.3 Extending the bound to adaptive algorithms

Proposition 10.7 only applies to non-adaptive algorithms, and may also apply to the class of locally-bounded adaptive algorithms defined in [AF23] (but we do not show it here). The bottleneck of the proof is the need of having a "good" upper bound on the maximum number of queries per sample as compared to the expected number of queries per sample, which is impossible for adaptive algorithms. To extend the proof, we must make the algorithm completely clueless unless it queries every individual sample within a fixed portion of the maximum. To achieve these goals we use two concepts: very short strings (to reduce the maximum number of queries per sample – but the extension for arbitrarily long strings is then proved as a simple corollary) and secret sharing (see below).

Lemma 10.8 (Definition and existence of secret-sharing code ensembles, [BEFLR20])**.** *There exist some constants $\delta, \zeta > 0$ and $\eta > 1$, for which: for every $k \ge 1$, there exist $m(k) \le 2k$ divisible*

by 3, and some $n(k)$ with $n(k) \leq \eta \log_2 m(k)$ for which there exists a code-ensemble $\mathcal{H} : \{0,1\} \to \mathcal{P}(\{0,1\}^{n(k)})$ with the following properties:

- *Sufficiently large: $|\mathcal{H}(0)| = |\mathcal{H}(1)| = \frac{2}{3}m(k)$.*
- *Fixed lower bound on the distance: for every $u, v \in \mathcal{H}(0) \cup \mathcal{H}(1)$ either $u = v$ or $d(u, v) > \delta$.*
- *Shared secret: for every set $I \subseteq \{1, \ldots, n(k)\}$ of size $|I| \leq \zeta n(k)$, and for every $w \in \{0,1\}^{|I|}$,*

$$\Pr_{u \sim \mathcal{H}(0)}[u|_I = w] = \Pr_{u \sim \mathcal{H}(1)}[u|_I = w]$$

By the specific construction of [BEFLR20], we can have $\delta = \frac{1}{30}$, $\zeta = \frac{1}{12}$ and $\eta = 4$, and also that every restriction of $\mathcal{H}(0)$ (or $\mathcal{H}(1)$) to a set I of at most $\zeta n(k)$ bits is uniform over $\{0,1\}^I$. Note that the following does not rely on the exact values of δ, ζ, η, only on the constraints guaranteed by the lemma.

Theorem 10.9. *Consider δ, ζ, η of Lemma 10.8. For every $\varepsilon < \frac{1}{4}\delta$ and every $k \geq 1$ there exist $m(k) \geq 2k$, $n(k) \leq \eta \log_2 m(k)$ and a distribution P over $\{0,1\}^{n(k)}$ that is ε-far from $\mathcal{S}_{m(k)}$ for which every ε-testing adaptive algorithm that makes less than $\frac{1}{96}\zeta\delta\varepsilon^{-1}m(k)\log_2 m(k)$ queries cannot find a witness against $P \in \mathcal{S}_{m(k)}$ with probability $\frac{1}{3}$.*

Proof. Let $k \geq 1$ and let $m = m(k) \geq 2k$ and $\log_2 m \leq n(k) \leq \eta \log_2 m$ as per Lemma 10.8. For every $0 < \varepsilon < \frac{1}{4}\delta$, we define the following distribution:

$$P : \begin{cases} a & \text{with probability } \frac{1-4\delta^{-1}\varepsilon}{2m/3},\ \forall a \in \mathcal{H}(0) \\ b & \text{with probability } \frac{4\delta^{-1}\varepsilon}{2m/3},\ \forall b \in \mathcal{H}(1) \end{cases}$$

By the secret sharing property, for every set Q of at most $\zeta n(k)$ indices and for every $w \in \{0,1\}^{|Q|}$,

$$\Pr_{x \sim P}[x \in \mathcal{H}(1) | x|_Q = w] = \Pr_{x \sim P}[x \in \mathcal{H}(1)] = 4\delta^{-1}\varepsilon$$

Observe that P is supported by exactly $\frac{4}{3}m$ elements. For every set A of m elements, at least $\frac{1}{3}m$ elements in the support of P are $\frac{1}{2}\delta$-far from every element in A. The minimum probability mass of an individual element in P is at least $\frac{4\delta^{-1}\varepsilon}{2m/3}$, hence

$$d(P, \mathcal{S}_A) > \frac{1}{3}m \cdot \frac{4\delta^{-1}\varepsilon}{2m/3} \cdot \frac{1}{2}\delta = \varepsilon$$

This holds for every A of size m, hence $d(P, \mathcal{S}_m) > \varepsilon$.

Assume that $m \geq \max\left\{2^{2\zeta^{-1}}, 6\right\}$. Let $n = n(k)$ be the length of the string and $n' = \lfloor \zeta n(k) \rfloor$ be the number of bits whose reading supplies no information about $x \sim P$ coming from $\mathcal{H}(0)$ or $\mathcal{H}(1)$. Consider a deterministic adaptive algorithm T with $q < \frac{1}{96}\zeta\delta\varepsilon^{-1}m\log_2 m$ queries. The number of queries per sample is trivially bounded by $n = O(\log_2 m)$, since this is the bit length of a sample.

We color every query in T with either red or green. In every path in the tree, for every $1 \leq i \leq s$, the first n' queries of the i-th sample are red and the others are green. Note that the color of a

query is unambiguous even if it is common to multiple paths, since the coloring only takes the path from the root into account. Note that in every path in T, the number of queries is bounded by n/n'-times the number of red queries, which is:

$$\frac{n}{n'} = \frac{n}{\lfloor \zeta n \rfloor} \leq \frac{n}{\zeta n - 1} \leq \frac{\log_2 m}{\zeta \log_2 m - 1} \leq \frac{2\zeta^{-1}}{\zeta \cdot 2\zeta^{-1} - 1} = 2\zeta^{-1}$$

We would like to bound the expected number of queries applied to $\mathcal{H}(1)$-samples to apply Proposition 10.5. For every $1 \leq i \leq q$, let R_i be an indicator for "the i-th query is red" and X_i be an indicator for "the i-th query is red, and it applies to $\mathcal{H}(1)$-sample". Let $X = \sum_{i=1}^{q} X_i$ be the number of red queries applied to $\mathcal{H}(1)$-samples and Y be the total number of queries (of any color) applied to $\mathcal{H}(1)$-samples.

By the secret sharing property, and the definition of "red queries" as the first $\lfloor \zeta n \rfloor$ queries, $\mathrm{E}[X_i |\ R_i = 1] = 4\delta^{-1}\varepsilon$, since the restriction of a sample drawn from P to at most ζn indices distributes exactly the same regardless of whether it belongs to $\mathcal{H}(0)$ or to $\mathcal{H}(1)$. Hence

$$\mathrm{E}[X_i] = \Pr[R_i = 0] \underbrace{\mathrm{E}[X_i \mid R_i = 0]}_{=0} + \underbrace{\Pr[R_i = 1]}_{\leq 1} \underbrace{\mathrm{E}[X_i \mid R_i = 1]}_{=4\delta^{-1}\varepsilon} \leq 4\delta^{-1}\varepsilon$$

By linearity of expectation, $\mathrm{E}[X] = \sum_{i=1}^{q} \mathrm{E}[X_i] \leq 4\delta^{-1}\varepsilon q < \frac{1}{24}\zeta m \log_2 m$. Recall that $Y \leq 2\zeta^{-1}X$ for every sufficiently large m, hence $\mathrm{E}[Y] \leq \frac{1}{12} m \log_2 m$. By Markov's inequality, $\Pr[Y > \frac{1}{8} m \log_2 m] < \frac{2}{3}$.

Consider a witness against $P \in \mathcal{S}_m$. By Observation 10.6, since $\mathrm{supp}(P) = \mathcal{H}(0) \cup \mathcal{H}(1)$ and $|\mathcal{H}(0)| = \frac{2}{3}m$, there must exist a "sub-witness" against $|\mathcal{H}(1)| \leq \frac{1}{3}m$, meaning that the restriction of the queries and answers to $\mathcal{H}(1)$-samples forms a witness against having support at most $\frac{1}{3}m$. By Proposition 10.5, this sub-witness must consist of at least $\frac{1}{3} m \log_2(\frac{1}{3}m)$ bits of $\mathcal{H}(1)$-samples. For $m \geq 6$, this bound requires more than $\frac{1}{8} m \log_2 m$ bits.

With probability greater than $\frac{1}{3}$, there are fewer than $\frac{1}{8} m \log_2 m$ queries in $\mathcal{H}(1)$-samples, hence they cannot form a witness against $|\mathcal{H}(1)| \leq \frac{1}{3}m$. In particular, in this case, there is no witness against $P \in \mathcal{S}_m$. For every $k \geq 1$ there exists $m \geq 2k$ for which this result applies for every $\varepsilon > 0$ smaller than some global constant. The lower bound follows from Lemma 5.15. □

Theorem 10.9 shows a lower bound of $\Omega(\varepsilon^{-1} m \log m)$ queries for every one-sided adaptive ε-testing of $\mathcal{S}_m$ for some fixed n (that depends on m). However, a "full" property testing lower bound would need to apply for at least an infinite set of value for n. The following corollary uses a simple repeating technique to provide this.

Lemma 10.10. *Let P be a distribution over $\{0,1\}^n$. Let $f_\ell : \{0,1\}^n \to \{0,1\}^{\ell n}$ be defined by the ℓ-fold repetition, that is $f_\ell(x_1, \ldots, x_n) = y$ where $y_{i \cdot n + j} = x_j$ for $1 \leq j \leq n$ and $0 \leq i \leq \ell - 1$, and consider the distribution $\hat{P} = f_\ell(P)$. Then for every $m \geq 1$, $d(P, \mathcal{S}_m) \leq d(\hat{P}, \mathcal{S}_m)$.*

Proof. Let $\hat{A} = \{\hat{x}_1, \ldots, \hat{x}_m\}$ be a set that realizes the distance $d(\hat{P}, \mathcal{S}_m)$. Let $g : \mathrm{supp}(\hat{P}) \to \hat{A}$ be the mapping provided by Lemma 5.12. For every $0 \leq i \leq \ell - 1$, let $h_i : \{0,1\}^{\ell n} \to \{0,1\}^n$ be the mapping $h_i(x) = x_{\{i \cdot n + 1, \ldots, i \cdot n + n\}}$. Choose $0 \leq i \leq \ell - 1$ uniformly, and let $h = h_i$. Let $A = \{h(\hat{x}_1), \ldots, h(\hat{x}_m)\}$.

$$
\begin{aligned}
d(P,\mathcal{S}_m) &\leq \mathop{\mathrm{E}}_{i\sim\{1,\ldots,n\}}[d(P,\mathcal{S}_A)] = \mathop{\mathrm{E}}_{x\sim P, i\sim\{1,\ldots,n\}}[d(x,A)] = \sum_{x\in\mathrm{supp}(P)} \Pr_P[x]\,\mathop{\mathrm{E}}_i[d(x,\mathcal{S}_A)] \\
&\underset{(*)}{\leq} \sum_{x\in\mathrm{supp}(P)} \Pr_P[x]\,\mathop{\mathrm{E}}_i[d(x,h(g(f_\ell(x))))] \\
&\underset{(\dagger)}{=} \sum_{x\in\mathrm{supp}(P)} \Pr_P[x] d(f_\ell(x),g(f_\ell(x))) \\
&= \sum_{x\in\mathrm{supp}(P)} \Pr_{\hat{P}}[f_\ell(x)] d(f_\ell(x),g(f_\ell(x))) \\
&= \sum_{\hat{x}\in\mathrm{supp}(\hat{P})} \Pr_{\hat{P}}[\hat{x}] d(\hat{x},g(\hat{x})) = d(\hat{P},\mathcal{S}_{\hat{A}}) = d(\hat{P},\mathcal{S}_m)
\end{aligned}
$$

The starred transition is correct since $h(g(f_\ell(x))) \in A$ by its definition. The daggered transition is correct since:

$$
\begin{aligned}
\mathop{\mathrm{E}}_{i\sim\{1,\ldots,n\}}[d(x,h(g(f_\ell(x))))] &= \frac{1}{\ell}\sum_{i=0}^{\ell-1} d(x,h_i(f(x))) \\
&= d(x\cdots x, h_0(g(f_\ell(x)))\cdots h_{\ell-1}(g(f_\ell(x)))) = d(f_\ell(x),g(f_\ell(x)))
\end{aligned}
$$

□

Corollary 10.11. *Every one-sided non-adaptive ε-testing algorithm for $\mathcal{S}_m$ must make at least $\Omega(\varepsilon^{-1} m \log m)$ many queries, for infinitely many values of the string length n.*

Proof. For a proper choice of $m \geq 2$, $q \geq 1$ and $\varepsilon > 0$, let P be a distribution over $\{0,1\}^n$ (for some n that may depend on m and ε) that is ε-far from $\mathcal{S}_m$ for which, for every deterministic adaptive algorithm T that makes at most q queries, the probability that it finds a witness against $P \in \mathcal{S}_m$ is less than $\frac{1}{3}$. For every $\hat{n} \geq n$, let $\hat{P} = f_{\lceil \hat{n}/n \rceil}(P)$ (as per Lemma 10.10). Note that P is ε-far from $\mathcal{S}_m$ as well.

Every bit of a sample drawn from $\hat{P}$ corresponds to a concrete bit of a sample drawn from P, hence every witness against $\hat{P} \in \mathcal{S}_m$ is also a witness against $P \in \mathcal{S}_m$. The probability to find this witness using less than q queries is less than $\frac{1}{3}$, hence every one-sided adaptive ε-testing algorithm for $\mathcal{S}_m$ must use at least q queries, even for distributions over arbitrarily long strings. This completes the proof since $q = \Omega(\varepsilon^{-1} m \log m)$. □

As a final remark, note that the corollary actually holds for sets of values of m and n that are not very sparse – all large enough integers of the form $3 \cdot 2^{4k}$ for m, and a set of positive density (that depends on m) for n.

Acknowledgement

We thank Noga Alon for pointing out the results of [Han64] and [KS67] that we use in Section 10.

References

[AF23] Tomer Adar and Eldar Fischer. Refining the adaptivity notion in the huge object model, 2023.

[Alo23] Noga Alon. On bipartite coverings of graphs and multigraphs. *arXiv preprint arXiv:2307.16784*, 2023.

[BEFLR20] Omri Ben-Eliezer, Eldar Fischer, Amit Levi, and Ron D Rothblum. Hard properties with (very) short pcpps and their applications. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, 2020.

[BFF+01] Tugkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 442–451. IEEE, 2001.

[BFR+00] Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 259–269. IEEE, 2000.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[Gol17] Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.

[GR11] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 68–75, 2011.

[GR22] Oded Goldreich and Dana Ron. Testing distributions of huge objects. 2022.

[Han64] Georges Hansel. Nombre minimal de contacts de fermeture nécessaires pour réaliser unefonction booléenne symétrique de n variables. *COMPTES RENDUS HEBDOMADAIRES DES SEANCES DE L ACADEMIE DES SCIENCES*, 258(25):6037, 1964.

[KS67] Gyula Katona and Endre Szemerédi. On a problem of graph theory. *Studia Scientiarum Mathematicarum Hungarica*, 2:2328, 1967.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252—-271, 1996.

[VV11] Gregory Valiant and Paul Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new clts. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 685–694, 2011.

[VV17] Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. *Journal of the ACM (JACM)*, 64(6):1–41, 2017.

[WY19] Yihong Wu and Pengkun Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *The Annals of Statistics*, 47(2):857–883, 2019.

[Yao77] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 222–227, 1977.

A Probabilistic Bounds

We prove and recall here some technical probabilistic bounds that are used in our proofs.

Lemma A.1. *Let X be a non-negative random variable and let $f, g : \mathbb{R} \to \mathbb{R}$ be two non-negative functions. Assume that $\Pr[g(X) \geq f(X)|B] = 1$ for some event B. Then $\mathrm{E}[g(X)|B] \geq \mathrm{Ct}[f(X)|B]$.*

Proof. $\mathrm{E}[g(X)|B] \geq \mathrm{Ct}[g(X)|B] = \Pr[B]\,\mathrm{E}[g(X)|B] \geq \Pr[B]\,\mathrm{E}[f(X)|B] = \mathrm{Ct}[f(X)|B]$ □

Lemma A.2 (A technical bound)**.** *Let X be a sum of independent variables $X_1, \ldots, X_n$, where each evaluates to 1 with probability p_i and evaluates to 0 otherwise. Then $\Pr[X = 0] \leq e^{-\mathrm{E}[X]}$.*

Proof. $\Pr[X = 0] = \prod_{i=1}^{n}(1 - p_i) \leq e^{-\sum_{i=1}^{n} p_i} = e^{-\mathrm{E}[X]}$ □

Observation A.3. *For $n \geq 1$ and $0 \leq p < \frac{1}{2n}$, $1 - (1-p)^n \geq \frac{1}{2}np$.*

Proof. Since $np < 1/2$, we have $(1-p)^n \leq 1 - np + (np)^2 \leq 1 - np + \frac{1}{2}np = 1 - \frac{1}{2}np$. □

Lemma A.4 (Multiplicative Chernoff's Bound)**.** *Let $X_1, \ldots, X_m$ be independent variables in $\{0, 1\}$. Let $X = \sum_{i=1}^{m} X_i$, then for every $\delta > 0$:*

$$\begin{aligned} \Pr[X < (1-\delta)\,\mathrm{E}[X]] &< \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mathrm{E}[X]} \\ \Pr[X > (1+\delta)\,\mathrm{E}[X]] &< \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mathrm{E}[X]} \end{aligned}$$

Lemma A.5 (Multiplicative Chernoff for well-dependent variables with a goal)**.** *Let $\mathcal{G} \subset \mathbb{R}^*$ be a set of goal sequences, satisfying that if u is a prefix of v and $u \in \mathcal{G}$ then $v \in \mathcal{G}$. Additionally let $R_1, \ldots, R_m$ be a set of random variables and $p_1, \ldots, p_m$ be values in $[0, 1]$, such that for every $1 \leq i \leq m$ and $v = (r_1, \ldots, r_{i-1}) \in \mathbb{R}^{i-1} \setminus \mathcal{G}$ (that can happen with positive probability) we have $\Pr[R_i \neq 0 | R_1 = r_1, \ldots, R_{i-1} = r_{i-1}] \geq p_i$. For every $1 \leq i \leq m$, let $X_i \in \{0, 1\}$ be an indicator for $R_i \neq 0$ and $X = \sum_{i=1}^{m} X_i$. Under these premises, for every $0 < \delta < 1$,*

$$\Pr\left[((R_1, \ldots, R_m) \notin \mathcal{G}) \wedge \left(X < (1-\delta)\sum_{i=1}^{m} p_i\right)\right] < \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\sum_{i=1}^{m} p_i}$$

Proof. We first define auxiliary random variables $Y_1, \ldots, Y_m \in \{0, 1\}$ that will depend on $R_1, \ldots, R_m$ but will be independent of each other. To draw the value of Y_i, we consider $R_1, \ldots, R_i$. Considering their respective values $r_1, \ldots, r_i$, if $(r_1, \ldots, r_{i-1}) \in \mathcal{G}$ we take Y_i to be equal to 1 with probability p_i and to 0 with probability $1 - p_i$, independently of all other choices so far. If $(r_1, \ldots, r_{i-1}) \notin \mathcal{G}$,

then we set $\alpha_i = \Pr[R_i \neq 0 | R_1 = r_1, \ldots, R_{i-1} = r_{i-1}]$ and choose Y_i according to r_i: If $r_i = 0$ (meaning in particular that $X_i = 0$) then we choose $Y_i = 0$. If $r_i \neq 0$, we choose $Y_i = 1$ with probability $\frac{p_i}{\alpha_i}$ and choose $Y_i = 0$ with probability $\frac{\alpha_i - p_i}{\alpha_i}$. This last choice is drawn independently of previous choices (note that in particular these are indeed probabilities between 0 and 1, since by the assumptions of the lemma $p_i \leq \alpha_i \leq 1$). We also define the sum $Y = \sum_{i=1}^{m} Y_i$.

It is not hard to see that $\Pr[Y_i = 1] = p_i$ for every $1 \leq i \leq m$. To conclude the proof, it remains to show that $Y_1, \ldots, Y_m$ are indeed independent (when not conditioning on the other random variables defined over our probability space), and that it is always the case that $Y \leq X$ or $(R_1, \ldots, R_m) \in \mathcal{G}$ (or both), since then we can use the multiplicative Chernoff bound to conclude that

$$\begin{aligned}
& \Pr\left[((R_1, \ldots, R_m) \notin \mathcal{G}) \wedge \left(X < (1-\delta)\sum_{i=1}^{m} p_i\right)\right] \\
\leq\ & \Pr\left[((R_1, \ldots, R_m) \notin \mathcal{G}) \wedge \left(Y < (1-\delta)\sum_{i=1}^{m} p_i\right)\right] \\
\leq\ & \Pr\left[Y < (1-\delta)\,\mathrm{E}[Y]\right] < \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\sum_{i=1}^{m} p_i}
\end{aligned}$$

For the independence assertion, we need to show that for every sequence of values $(b_1, \ldots, b_{i-1}) \in \{0,1\}^{i-1}$ we have $\Pr[Y_i = 1 | Y_1 = b_1, \ldots, Y_{i-1} = b_{i-1}] = p_i$. We note that it is enough to show that for every sequence $(r_1, \ldots, r_{i-1}) \in \mathrm{supp}(R_1, \ldots, R_{i-1})$ we have $\Pr[Y_i = 1 | R_1 = r_1, \ldots, R_{i-1} = r_{i-1}] = p_i$, since the choices of $Y_1, \ldots, Y_{i-1}$ depend only on the values of $R_1, \ldots, R_{i-1}$ (and possible additional independent coin tosses). To show the latter, we go over the cases. If $(r_1, \ldots, r_{i-1}) \in \mathcal{G}$ then Y_i was explicitly defined to be 1 with probability exactly p_i. If $(r_1, \ldots, r_{i-1}) \notin \mathcal{G}$, we write

$$\begin{aligned}
& \Pr\left[Y_i = 1 | R_1 = r_1, \ldots, R_{i-1} = r_{i-1}\right] \\
=\ & \Pr\left[Y_i = X_i = 1 | R_1 = r_1, \ldots, R_{i-1} = r_{i-1}\right] \\
=\ & \Pr\left[Y_i = 1 | X_i = 1, R_1 = r_1, \ldots, R_{i-1} = r_{i-1}\right] \cdot \Pr\left[X_i = 1 | R_1 = r_1, \ldots, R_{i-1} = r_{i-1}\right] \\
=\ & \frac{p_i}{\alpha_i} \cdot \alpha_i = p_i
\end{aligned}$$

To show the conditional inequality assertion, note that $(R_1, \ldots, R_m) \notin \mathcal{G}$ implies in particular that $(R_1, \ldots, R_{i-1}) \notin \mathcal{G}$ for every $1 \leq i \leq m$. Hence, all the choices of Y_i in this case are made so that $Y_i \leq X_i$ for $1 \leq i \leq m$, and in particular $Y \leq X$. □

B Proof of Proposition 5.11

To prove Proposition 3.26, we need the following lemma.

Lemma B.1. *Consider a property $\mathcal{P}$ of distributions over $\{0,1\}^n$ that has a one-sided ε-test for every $\varepsilon > 0$, and consider some $P \in \mathcal{P}$. For every distribution Q for which $\mathrm{supp}(Q) \subseteq \mathrm{supp}(P)$, $Q \in \mathcal{P}$ as well.*

Proof. Let $\mathcal{A}_{\varepsilon,n}$ be an ε-testing algorithm for $\mathcal{P}$ in the Huge Object model that draws $s(\varepsilon, n)$ samples and makes $q(\varepsilon, n)$ queries. For every ε, if $\mathcal{A}_{\varepsilon,n}$ rejects Q with positive probability, then there must

be a sequence of samples $X_1, \ldots, X_{s(\varepsilon,n)}$ and a sequence of random query choices $Y_1, \ldots, Y_{q(\varepsilon,n)}$ for which the algorithm rejects. If we run $\mathcal{A}_{\varepsilon,n}$ on P as its input, there is a positive probability to draw exactly the same sequence $X_1, \ldots, X_{s(\varepsilon,n)}$ (since $\text{supp}(Q) \subseteq \text{supp}(P)$), and to make the exact same random query choices $Y_1, \ldots, Y_{q(\varepsilon,n)}$. In this case, $\mathcal{A}_{\varepsilon,n}$ rejects P, a contradiction to its one-sideness. Hence $\mathcal{A}_{\varepsilon,n}$ must always accept Q and this holds for every $\varepsilon > 0$, which implies $Q \in \mathcal{P}$. □

We now recall the proposition to be proved.

Proposition 5.11. *Consider any label-invariant property of distributions $\mathcal{P}$ that has a one-sided ε-test for every $\varepsilon > 0$ (with any number of samples and queries). There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathcal{P} = \mathcal{S}_f$.*

Proof. Let $f(n)$ be as follows:

$$f(n) = \begin{cases} 0 & \mathcal{P} \cap \mathcal{D}(\{0,1\}^n) = \emptyset \\ \max_{P \in \mathcal{P}} |\text{supp}(P)| & \text{otherwise} \end{cases}$$

In the first case, the property is empty for n-bit strings. In the second case, a maximum must exist, and it cannot be more than 2^n. For every n for which $f(n) > 0$, we also define P_n as one of the distributions that demonstrate the maximum. By the definition of $f(n)$, $\mathcal{P}$ does not contain any distribution that is supported by more than $f(n)$ elements.

Consider some n for which $f(n) > 0$, and consider some distribution P over $\{0,1\}^n$ that is supported by at most $f(n)$ elements. Let $\sigma : \{0,1\}^n \to \{0,1\}^n$ be a permutation for which $\text{supp}(P) \subseteq \text{supp}(\sigma(P_n))$. By the label invariance of $\mathcal{P}$, $\sigma(P_n) \in \mathcal{P}$. By Lemma B.1, $P \in \mathcal{P}$, because its support is a subset of the support of $\sigma(P_n)$ which belongs to $\mathcal{P}$. Hence $\mathcal{P}$ contains all distributions that are supported by at most $f(n)$ elements, as desired. □