On Sample-Based Testers[*]

Oded Goldreich[†] Dana Ron[‡]

July 3, 2014

Abstract

The standard definition of property testing endows the tester with the ability to make arbitrary queries to "elements" of the tested object. In contrast, sample-based testers only obtain independently distributed elements (a.k.a. labeled samples) of the tested object. While sample-based testers were defined by Goldreich, Goldwasser, and Ron (*JACM* 1998), with few exceptions, most research in property testing is focused on query-based testers.

In this work, we advance the study of sample-based property testers by providing several general positive results as well as by revealing relations between variants of this testing model. In particular:

- We show that certain types of query-based testers yield sample-based testers of sublinear sample complexity. For example, this holds for a natural class of proximity oblivious testers.
- We study the relation between distribution-free sample-based testers and one-sided error sample-based testers w.r.t the uniform distribution.

While most of this work ignores the time complexity of testing, one part of it does focus on this aspect. The main result in this part is a sublinear-*time* sample-based tester in the dense graphs model for k-Colorability, for any $k \geq 2$.

Keywords: Property Testing, Graph Properties, One-Sided Error

[*] Partially supported by the Israel Science Foundation (grant No. 671/13).

[†] Department of Computer Science, Weizmann Institute of Science, Rehovot, ISRAEL. E-mail: oded.goldreich@weizmann.ac.il

[‡] Department of EE–Systems, Tel-Aviv University, Ramat-Aviv, ISRAEL. E-mail: danar@eng.tau.ac.il

Contents

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., [11, 29, 30]). Loosely speaking, property testing typically refers to sub-linear complexity probabilistic algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by performing queries; that is, the object is seen as a function and the testers get oracle access to this function (and thus may be expected to work in time that is sub-linear in the length of the object).

The standard definition of property testing, which is aluded to above, endows the tester with the ability to make queries. This is the definition presented in [31], and it is also the main definition studied in [14] as well as in most research on property testing. Nevertheless, a weaker notion, where the tester is only provided with uniformly distributed labeled samples (or, equivalently, is only allowed to make uniformly and *independently distributed* queries)[1] was also presented (and briefly studied) in [14]. We call such testers *sample-based*.

While sample-based testers were studied in [14][2] and in some subsequent works (see, e.g., [13]), these studies tend to be negative in nature: Their focus is on showing why the query ability utilized in the main positive results is necessary for efficiently testing the property that is being considered, which amounts to presenting high lower bounds on the sample complexity of sample-based testers for that property. A few positive results on sample-based testers have been obtained in previous work [13, 24, 3], and they are further discussed in Subsection 1.3. We also disucss the relation to testing properties of *distributions* in Subsection 1.2.1.

In this work we aim at a broader study of sample-based property testing, and we obtain several general positive results as well as reveal relations between variants of this testing model. Although sample-based testers are typically much less efficient than general (query-based) testers, in many applications, random labeled samples are easier to obtain than a full query capacity. Hence, we believe that sample-based testers may be of practical value.

1.1 Sample-based testers of sublinear sample complexity

The complexity of testers (of various types) is measured as a function of the size of the object, denoted n, and the proximity parameter, denoted ϵ. (Both these parameters are given as input to standard testers.) When we say that a complexity measure is *sublinear*, we mean that it is sublinear as a function of n, when fixing ϵ. (Typically, the query and sample complexities are at least linear in $1/\epsilon$.) With one exception (i.e., Section 5), all our positive results assert sublinear sample complexity for properties that are impossible to learn within such complexity (under the uniform distribution).

One of our results transforms certain Proximity Oblivious Testers (POTs) into sample-based testers of sublinear sample complexity, where the "level of sublinearity" depends on the (constant) query complexity of the POT. Loosely speaking, Proximity Oblivious Testers (POTs) are "basic testers" that do not get a proximity parameter as input, and make only a constant number of queries to the tested object [18]. They are required to reject objects with probability that is lower

[1] We stress that the canonical testers of graph properties, which are used in many works and are studied explicitly in [20, Sec. 4], do *not* make uniformly and independently distributed queries. Indeed, they select at random a uniformly distributed set of vertices, but their queries (which correspond to all vertex pairs) are dependent, although each is uniformly distributed. In general, many testers make uniformly distributed queries, but these queries are typically not fully independent of one another. See further discussion following Theorem 1.1.

[2] In fact, the main definition in [14] refers to (distribution-free) sample-based testers (cf. [14, Def. 2.1]).

bounded by a function of the object's distance to the property (see Definition 2.2).[3] This function is called the *detection probability function.*

Theorem 1.1 (Proximity Oblivious Testers with uniformly distributed queries imply sample-based testers of sublinear sample complexity): *Suppose that* Π *has a* q*-query POT with detection probability function* ϱ *that makes uniformly distributed queries. Then,* Π *has a sample-based tester of sample complexity* $s(n,\epsilon) = O(n^{1-(1/q)}/\varrho(\epsilon)^{2+(3/q)})$*. Furthermore, if the POT has one-sided error, then so does the sample-based tester.*

We stress that the premise of Theorem 1.1 is merely that each of the queries made by the POT is uniformly distributed, although the queries may be dependent on one another and may even be chosen adaptively. A typical example is the celebrated `linearity` tester of [5], which is based on a three-query POT. This three-query POT selects x and y uniformly and independently in the function's domain (which is a group), and makes the queries x, y and $x + y$.

Indeed, Theorem 1.1 implies that `linearity` has a sample-based tester of sample complexity $O(n^{2/3}\text{poly}(1/\epsilon))$, but a better (and tight) result can be obtained directly (see Theorem 5.1). In general, POTs that satisfy the "uniform query distribution" condition are quite common; in fact, in the dense graph model, any POT can be converted to one that makes uniformly distributed queries, while at most squaring the number of queries (see [18, Sec. 4.2], following [20, Sec. 4]).[4] Actually, we can get stronger results for the dense graph model.

Theorem 1.2 (on sample-based testers in the dense graph model): *Let* n *denote the size of the adjacency matrix of a graph, that is, the number of vertex-pairs.*

1. (From quasi-canonical testers to sample-based testers): *Let* Π *be a graph property and consider the task of testing* Π *in the dense graph model. If* Π *has a POT with detection probability function* ϱ *that inspects the subgraph induced by a random set of* ν *vertices, then* Π *has a sample-based tester of sample complexity* $s(n,\epsilon) = O(n^{1-(1/(\nu-1))}/\varrho(\epsilon)^2)$*. Furthermore, if the POT has one-sided error, then so does the sample-based tester.*

2. *In the dense graph model, for every* $k \geq 2$*,* k*-Colorability has a sample-based tester of sample complexity* $s(n,\epsilon) = O(1/\epsilon) \cdot \sqrt{n}$*. Furthermore, this tester has one-sided error.*

Note that Bipartitness does not have a POT (cf. [18]). Hence, while Item 1 merely presents a quantitative improvement over Theorem 1.1, Item 2 goes beyond the scope of Theorem 1.1. Two more comments are in order:

1. The square-root dependence on n of the sample complexity in the second item of the theorem is tight for (sample-based) testing of bipartiteness.

2. The tester referred to in the second item of the theorem is not computationally efficient. We address this issue in Subsection 1.4.

1.2 Perspective on other testing models

Sample-based testing offers interesting perspectives on several notions.

[3] In contrast, standard testers do get a proximity parameter, denoted ϵ, as input, make a number of queries that depend on ϵ, and are required to reject any object that is ϵ-far from the property with probability at least 2/3.

[4] In fact, the resulting POT inspects the subgraph induced by a random set of $O(q)$ vertices, where q is the query complexity of the original POT.

1.2.1 Testing symmetric properties and testing distributions

As argued in [23] (see also [16]), natural families of properties are defined in terms of invariances, where each property in the family is invariant under a group of permutations acting on the functions' domain. (Indeed, testing graph properties in the dense graph model (as in Theorem 1.2) is one famous example, and linearly invariant properties [23] is another.) At the very extreme, one may consider symmetric properties, which are properties that are invariant under the symmetric group acting on the functions' domain. We observe that *when testing symmetric properties, samples are essentially as good as queries* (see Theorem 6.1).

We also articulate Sudan's observation that *testing distributions* (for any property of distributions, cf. [4]) *can be tightly reduced to testing symmetric properties* [32, Sec. 2.1]. In particular, Theorem 6.4 formally relates the model of testing distributions[5] to the standard model of property testing (i.e., testing properties of functions w.r.t the uniform distribution over their domains). While establishing this relation is technically simple, to the best of our knowledge, this is the first time that the model of testing distributions has been formally related to the standard model of property testing, which relates to testing functions (w.r.t the uniform distribution over their domains).

1.2.2 Distribution-Free Testing and One-Sided Error

The focus on sample-based testers brings us closer to the area of computational learning theory (cf. [33, 25]). Within this mind-frame, it is natural to consider distribution-free testing, which was also defined in [14] but received little attention so far. In distribution-free testing, the sample that the testing algorithm receives is distributed according to an arbitrary and unknown distribution, and distance between functions (and hence distance to having the property) is defined according to the same distribution. Our initial feeling was that there may be a relation between *one-sided error* sample-based testing (under the uniform distribution) and *distribution-free* (sample-based) testing. This feeling was partially confirmed by a general upper bound on the sample complexity of the former in terms of the sample complexity of the latter, but beyond this upper bound the two complexity measures may exhibit different relationships. Our results are summarized in the following theorem, where $\texttt{OSE}(\Pi)$ denotes the sample complexity of one-sided error sample-based testing Π (under the uniform distribution), and $\texttt{DF}(\Pi)$ denotes the sample complexity of distribution-free (sample-based) testing Π.

Theorem 1.3 (distribution-free sample-based testers versus one-sided error sample-based testers under the uniform distribution):

1. *For every property* Π*, it holds that* $\texttt{OSE}(\Pi) = \widetilde{O}(\texttt{DF}(\Pi)^2)$.
2. *There exists a property* Π *such that* $\texttt{OSE}(\Pi) = \Omega(n)$ *but* $\texttt{DF}(\Pi) = \text{poly}(1/\epsilon) \cdot \frac{n}{\log n}$.
3. *There exists a property* Π *such that* $\texttt{OSE}(\Pi) = \Theta(\texttt{DF}(\Pi))$.
4. *There exists a property* Π *such that* $\texttt{OSE}(\Pi) = \widetilde{\Theta}(1/\epsilon) \cdot \log \texttt{DF}(\Pi)$.
5. *There exists a property* Π *such that* $\texttt{OSE}(\Pi) = \widetilde{O}(1/\epsilon)$ *but* $\texttt{DF}(\Pi) = \Omega(n)$.

[5] In this model the tested object is a distribution D over some domain X and the algorithm receives points $x \in X$ that are distributed according to D. The goal is to decide whether D has a particular property or is far from any distribution that has the property (where the distance measure between distributions is usually the variation distance).

Hence, there exists a general upper bound of $\mathtt{OSE}$ in terms of $\mathtt{DF}$ (i.e., Item 1), but in specific cases we may see quite different relations ranging from $\mathtt{OSE} = \omega(\mathtt{DF})$, to $\mathtt{OSE} = \Theta(\mathtt{DF})$, to $\mathtt{OSE} = o(\mathtt{DF})$ (cf., Items 2-5). We mention that the properties used in Items 2-4 are natural ones.

1.3 Related work

As mentioned upfront, sample-based testers were considered in several prior works, starting with the work of Goldreich, Goldwasser, and Ron [14]. Ironically, the main definition in their work (i.e. [14, Def. 2.1]), refers to (distribution-free) sample-based testers, whereas the now-standard definition of query-based testers (with respect to the uniform distribution) is presented there as a variant (see Item 3 in [14, Sec. 2]). However, the bulk of [14] is devoted to the study of query-based testers, and this notion became the standard in the area.

Nevertheless, sample-based testers were considered also in subsequent works, which focused on query-based testers. Typically, the perspective is negative; that is, the focus is on lower bounds on sample complexity, which are presented as a justification for the use of queries in the main positive results. For example, the first study of testing monotonicity [13] focuses on query-based testers, but also provides a lower bound on the sample complexity of sample-based testers, which is shown to be tight. Indeed, the latter lower bound (i.e., [13, Thm. 5]) is used as justification for the use of queries in the main result (i.e., [13, Thm. 1]), and the sample-based tester (of [13, Thm. 6]) is viewed as indicating that this lower bound is tight.

Two notable exceptions appear in [24, 3], Kearns and Ron [24] consider sample-based testing (under the uniform distribution) for decision trees of a bounded size s over $[0,1]^d$ (for constant d) and for a special class of neural networks with s hidden units. They design testers whose sample complexity is significantly lower than that required for learning the corresponding class of functions. However, their testers are only required to reject functions that are far from a super-class of the tested class (determined by a larger size parameter s'). For the special case of interval functions ($d = 1$), Blum *et al.* [3] showed how this relaxation of the rejection requirement can be removed, and they obtained optimal (in terms of the dependence on s) sample-based testers. Blum *et al.* [3] also present sample-based testers (which out-perform learning algorithms) for linear threshold functions under the Gaussian distribution.

Actually, the work of Blum *et al.* [3] puts forward a more refined notion, called active testers. These may be viewed as a generalization of sample-based testers, where the testers are provided with *unlabeled* samples, and may query the function only at points that appear in the given sample. Blum *et al.* [3] consider both the sample-complexity and the query-complexity of these testers, where the latter is typically smaller (since the tester does not query the function on all the sample points). They view sample-based testers as a special case (which they call *passive testing*) in which the testers query the function on all points in the sample (and so in this case their notion of query-complexity equals the notion of sample-complexity).

1.4 The computational complexity aspect

Throughout most of this work, we ignore the computational complexity aspect (i.e., the running-time of the various testers). This choice seems crucial to some of our results; notable examples include Proposition 2.4, Theorem 3.1 (see also Theorems 1.1 and 3.5), Theorem 4.5 (see Item 2 in Theorem 1.2), and Theorem 7.2. In contrast, the following result (i.e., Theorem 1.4) is of interest only because it addresses the computational complexity aspect, which was ignored in Item 2 of Theorem 1.2).

Theorem 1.4 (on computational efficient sample-based testers in the dense graph model):

1. *Bipartiteness has a* (one-sided error) *sample-based tester of* time *complexity* $O(1/\epsilon) \cdot \sqrt{n}$.
2. *For every* $k \geq 3$*, k-Colorability has a* (one-sided error) *sample-based tester of* time *complexity* $f_k(\epsilon) \cdot n^{1-(1/2k)}$*, where* $f_k(\epsilon) \stackrel{\rm def}{=} \exp(\exp(\widetilde{O}(k/\epsilon)))$.

Indeed, we do not know whether for $k \geq 3$, k-Colorability has a sample-based tester of time complexity $\mathrm{poly}(1/\epsilon) \cdot \sqrt{n}$. In order to establish the second item in Theorem 1.4, we build on a technique introduced by Alon and Krivelevich [2]. They showed that if a graph is ϵ-far from being k-colorable, then with high constant probability, a subgraph induced by $\Theta(k \log k/\epsilon^2)$ uniformly selected vertices will not be k-colorable. To this end they introduced (as a mental experiment) a process by which new vertices that are added to the sample *restrict* the legal colorings of previously added vertices.[6] Their analysis uses the fact that with each newly added vertex we get all edges to previously selected vertices (as part of the induced subgraphs). In contrast, for the sample size we use, we cannot expect to obtain so much information, and this is one aspect in which we depart from their analysis. The second aspect is that we need to turn the mental experiment into an efficient algorithm (which finds a small subgraph that is not k-colorable) while the algorithm lacks some of the knowledge that the mental experiment has.

1.5 Organization

In Section 2.1 we recall the standard definitions of (query-based) testers and proximity oblivious testers. In Section 2.2 we recall and briefly discuss the definition of sample-based testers.

In Section 3 we show that certain POTs imply sample-based testers of sublinear sample complexity. In particular, we present a generalization of Theorem 1.1, which derives sample-based testers of sublinear complexity from any POT that refrains from "heavy queries" (i.e., queries that assign much weight to a specific location). Sample-based testers in the dense graphs model are studied in Section 4, and the proofs of Theorems 2 and 1.4 appear there. Section 5 presents an optimal (one-sided error) sample-based tester of linearity. In Section 6 we study symmetric properties.

In Section 7 we study the relation between one-sided error sample-based testing (under the uniform distribution) and distribution-free (sample-based) testing: The results of this study are summarized in Theorem 1.3.

In Appendix A.1, we consider a relaxation of the definition of POTs. The only relation of this section to the rest of the paper is that such relaxed POTs can be used in Section 3.

2 Preliminaries

Property testing is a relaxation of decision problems and it focuses on algorithms that can only read parts of the input. Thus, the input is represented as a function (to which the tester has oracle access) and the tester is required to accept functions that have some predetermined property (i.e., reside in some predetermined set) and reject any function that is "far" from the set of functions having the property. Distances between functions are defined as the fraction of the domain on which the functions disagree, and the threshold determining what is considered far is presented as a proximity parameter, which is explicitly given to the tester.

An asymptotic analysis is enabled by considering an infinite sequence of domains, functions, and properties. The domains and properties (in the infinite sequence) are described by a finite sequence

[6] Such a process was introduced previously in [14], but it did not lend itself to our purposes.

of parameters, which include the size of the domain, denoted n. For simplicity, we shall present our results for sequences that are determined by this size parameter. That is, for any n, we consider functions from D_n to R_n, where $|D_n| = n$. (Often, one just assumes that $D_n = [n] \stackrel{\rm def}{=} \{1, 2, \ldots, n\}$.) Thus, in addition to the input oracle, representing a function $f : D_n \to R_n$, the tester is explicitly given two parameters: a size parameter, denoted n, and a proximity parameter, denoted ϵ.

2.1 The standard definitions of query-based testers

In this section we recall the standard definition of property testing as well as the definition of proximity oblivious testers. Both definitions refer to oracle machines that are given oracle access to a function $f : D_n \to R_n$ (as well as free access to some relevant parameters such as n). We denote by $M^f(p)$ the output of oracle machine M on input parameter p when given oracle access to f.

Definition 2.1 (property tester, following [31, 14]): *Let* $\Pi = \bigcup_{n \in \mathbb{N}} \Pi_n$*, where* Π_n *contains functions defined over the domain* D_n *and ranging over* R_n*. A* tester for a property Π *is a probabilistic oracle machine* T *that satisfies the following two conditions:*

1. *The tester accepts each* $f \in \Pi$ *with probability at least* $2/3$*; that is, for every* $n \in \mathbb{N}$ *and* $f \in \Pi_n$ (and every $\epsilon > 0$), *it holds that* $\Pr[T^f(n, \epsilon) = 1] \geq 2/3$.
2. *Given proximity parameter* $\epsilon > 0$ *and oracle access to any* f *that is* ϵ*-far from* Π*, the tester rejects with probability at least* $2/3$*; that is, for every* $n \in \mathbb{N}$ *and* $\epsilon > 0$*, if* $f : D_n \to R_n$ *is* ϵ*-far from* Π_n*, then* $\Pr[T^f(n, \epsilon) = 0] \geq 2/3$*, where* f *is* ϵ-far from Π_n *if, for every* $g \in \Pi_n$*, it holds that* $\delta(f, g) \stackrel{\rm def}{=} |\{e \in D_n : f(e) \neq g(e)\}|/n > \epsilon$.

   Indeed, the distance of f from Π*, denoted* $\delta_\Pi(f)$*, equals* $\min_{g \in \Pi_n}\{\delta(f, g)\}$.

If the tester accepts every function in Π *with probability 1, then we say that it has* one-sided error*; that is,* T *has one-sided error if for every* $f \in \Pi_n$ *and every* $\epsilon > 0$*, it holds that* $\Pr[T^f(n, \epsilon) = 1] = 1$*. A tester is called* non-adaptive *if it determines all its queries based solely on its internal coin tosses* (and the parameters n and ϵ)*; otherwise it is called* adaptive.

Definition 2.1 does not specify the query complexity of the tester, and indeed an oracle machine that queries the entire domain of the function qualifies as a tester (with zero error probability...). Needless to say, we are interested in testers that have significantly lower query complexity.

Some testers (e.g., the celebrated linearity tester of [5]) operate by repeating some basic tests for a number of times that depends on the proximity parameter, whereas the basic test is oblivious of the proximity parameter. Such basic tests are captured by the following definition.

Definition 2.2 (Proximity Oblivious Tester (POT), following [18, 19]): *Let* Π *be as in Definition 2.1 and let* $\varrho : (0, 1] \to (0, 1]$ *be monotone. A* POT with detection probability ϱ for Π *is a probabilistic oracle machine* T *that makes a constant number of queries and satisfies the following two conditions with respect to some constant* $c \in (0, 1]$*:*

1. *For every* $n \in \mathbb{N}$ *and* $f \in \Pi_n$*, it holds that* $\Pr[T^f(n) = 1] \geq c$.
2. *For every* $n \in \mathbb{N}$ *and* $f : D_n \to R_n$ *not in* Π_n*, it holds that* $\Pr[T^f(n) = 1] \leq c - \varrho(\delta_\Pi(f))$*,*

The constant c *is called the* threshold probability*. A POT is said to have* one-sided error *if* $c = 1$.

We stress that, in contrast to a standard tester, a POT only gets one explicit parameter (i.e., the size parameter, n). Standard testers are obtained by invoking the POT for an adequate number of times. Specifically, for one-sided error POTs, we invoke the POT for $O(1/\varrho(\epsilon))$ times and accept if and only if all invocations returned 1. For general POTs with threshold probability c, we invoke the POT for $O(1/\varrho(\epsilon)^2)$ times and accept if and only if at least $c - (\varrho(\epsilon)/2)$ of the invocations returned 1.

2.2 Sample-based testers

A sample-based tester can be defined in two equivalent ways. The first definition views such a tester as one whose queries are distributed uniformly in the domain, independently of one another. That is, each query of such a tester is distributed uniformly in D_n, independently of all prior queries (and of the answers provided to these queries). The second definition views a sample-based tester as obtaining a sequence of "labeled samples" (i.e., the samples are independently and uniformly distributed in D_n and they are coupled with the corresponding values of the function), where the length of this sequence is predetermined based on n and ϵ:

Definition 2.3 (sample-based tester, following [14]): *Let* Π *be as in Definition 2.1 and* $s : \mathbb{N} \times (0,1] \to \mathbb{N}$. *A* (sample-based) tester of sample complexity s for a property Π *is a probabilistic algorithm* T *that satisfies the following two conditions for every* $n \in \mathbb{N}$*:*

1. *For every* $f \in \Pi_n$ (and every $\epsilon > 0$)*, it holds that*

$$\mathrm{Pr}_{r_1,\ldots,r_s \in D_n}[T(n, \epsilon, (r_1, f(r_1)), \ldots, (r_s, f(r_s)))=1] \geq \frac{2}{3},$$

   where $s = s(n, \epsilon)$ *and* $(r_1, \ldots, r_s)$ *is uniformly distributed in* D_n^s*.*

2. *For every* $\epsilon > 0$*, and every* $f : D_n \to R_n$ *that is* ϵ*-far from* Π*,*

$$\mathrm{Pr}_{r_1,\ldots,r_s \in D_n}[T(n, \epsilon, (r_1, f(r_1)), \ldots, (r_s, f(r_s)))=0] \geq \frac{2}{3},$$

   where again $s = s(n, \epsilon)$ *and* $(r_1, \ldots, r_s)$ *is uniformly distributed in* D_n^s*.*

The sequence $(r_1, f(r_1)), \ldots, (r_s, f(r_s)))$ *is called a* sample labeled by f*. As in Definition 2.1, if the tester accepts every function in* Π *with probability 1, then we say that it has* one-sided error*.*

The equivalence of this definition to the first one (or rather one of the directions of this equivalence) is based on our disregard of the computational complexity of the tester (i.e., the complexity of generating queries). This is clarified in the proof of the following proposition.

Proposition 2.4 (equivalence of the two formulations): *A property* Π *has a tester of query complexity* q *that uses queries that are uniformly and independently distributed in the domain if and only if it has a sample-based tester of sample complexity* q*.*

Proof: Given a sample-based tester T', we easily obtain a corresponding querying tester T by letting the latter emulate T' in a straightforward manner. That is, on input n and ϵ, the tester T selects uniformly and independently an adequate number of queries, and feeds T' with the corresponding sequence of query and answer pairs, where T obtains the answers by querying its oracle.

The other direction is a bit more tricky. The issue is that the final decision of the tester may depend on the coins that it has used to produce the queries, and so merely replacing the queries with uniformly distributed samples will not do. We should augment these samples with a sequence of random coins that would have led the original tester to make these queries. Details follow.

We are given a (possibly adaptive) tester T with a guarantee on the distribution of its queries, and wish to construct a sample-based tester T'. The sample-based tester T' is given a sequence of pairs $(r_1, v_1), \ldots, (r_s, v_s)$, where $v_i = f(r_i)$ for each $i \in [s]$, and it operates by selecting at random coins ω such that using coins ω and having oracle access to f, the oracle machine T makes the queries $r_1, \ldots, r_s$. By the hypothesis, the set of such ω's constitutes an n^{-s} fraction of the set of all possible coin tosses. Furthermore, T' can reconstruct the former set without making any queries to f (by emulating the execution of T using coins ω and using v_i as the answer to the i^{th} query, regardless of the identity of this query). ■

On the operation of sample-based one-sided error testers. The operation of any sample-based one-sided error tester is totally determined by its sample: Being sample-based, this tester has no control over its access to the function, and having one-sided error it has no real control on its decision; that is, without loss of generality, the tester accepts if and only if the labeled sample that it has obtained is consistent with some function that has the property. This is the case because in case of consistency it must accept, whereas in case of inconsistency it better reject (since this may only improve its performance).

3 POTs and sample-based testers

The fact that *certain* POTs yield sample-based testers of sublinear sample complexity was proven implicitly in [12, Apdx. A.2]. The context of that result is of testing graph properties in the bounded-degree model (introduced in [17]). Here we present much more general results.

3.1 The main result

In this subsection we prove Theorem 1.1, or more precisely, a slightly stronger version of it. Specifically, we consider POTs that make queries that are *each* almost uniformly distributed in the domain, although their joint distribution may be very dependent. Stated formally, for $\alpha \in (0, 1]$, a tester (or a POT) is called α-fair if, *for each i and $j \in D_n$* (and for every $f : D_n \to R_n$), *the i^{th} query of the tester* (when it accesses the oracle f) *equals j with probability at most* $1/\alpha n$. We stress that here we consider the marginal distribution of the i^{th} query. Note that by [20, Sec. 4], we may assume without loss of generality, that any constant-query POT in the dense graphs model is 1-fair. Lastly, we mention that the notion of a fair tester is reminiscent of the notion of a smooth decoder [22].

Theorem 3.1 (from POTs to sample-based testers – basic version): *Suppose that Π has a q-query POT with detection probability ϱ that is $\Omega(1)$-fair. Then, Π has a sample-based tester of sample complexity $s(n, \epsilon) = O(n^{1-(1/q)}/\varrho(\epsilon)^{2+(3/q)})$. Furthermore, if the POT has one-sided error, then so does the sample-based tester* (and the sample complexity can be reduced to $O(n^{1-(1/q)}/\varrho(\epsilon)^{1+(3/q)})$).

Note that the POT in the premise of Theorem 3.1 implies a standard (query-based) tester of query complexity $O(1/\varrho(\epsilon)^b)$, where $b = 2$ in the general case and $b = 1$ in the one-sided error case. The fairness condition is essential to Theorem 3.1; however, it is satisfied by almost all known POTs. See further discussion in Section 3.2.

Proof: As in [12, Apdx. A.2], the basic idea is that a random sample of the said size is very likely to contain a sequence of queries that are made by the POT on some setting of its random coins. Furthermore, this setting is almost uniformly distributed among all possible settings of the random coins. The proof is devoted to actually establishing that the above assertions hold, in particular given that the algorithm may be adaptive. We note that the situation here is more complex than in [12, Apdx. A.2], since the property is not necessarily closed under any non-trivial invariance.

Throughout the analysis we fix the POT, denoted T, and *fix the function, denoted f, tested by T*. By the hypothesis, T is α-fair, for some constant $\alpha > 0$. Denoting the randomness complexity of T by r, it follows that $2^r \geq \alpha n$. We consider all possible random strings $\omega \in \{0,1\}^r$. A key notion is that of the sequence of queries generated by ω, which is defined as *the sequence of queries that T makes when using randomness ω and having access to f.*

We start by presenting a sample-based algorithm that emulates the POT up to an $O(\delta)$-deviation, where δ is a parameter to be determined later.[7] For $t = \Theta(n^{1-(1/q)}/\delta^{3/q})$, the sample-based tester that we wish to construct is given a sequence of qt labeled examples $(s_1, f(s_1)), \ldots, (s_{qt}, f(s_{qt}))$, where $\overline{S} = (s_1, \ldots, s_{qt})$ is uniformly distributed in $[n]^{qt}$. For each $i \in [q]$, we let $\overline{S}_i$ denote the subsequence $(s_{(i-1)t+1}, \ldots, s_{it})$.

We say that $(\omega, \overline{S}) \in \{0,1\}^r \times [n]^{qt}$ is good if the sequence generated by ω is in $(\overline{S}_1, \ldots, \overline{S}_q)$ (i.e., for each $i \in [q]$, when given oracle access to f and using coins ω the i^{th} query of T is in $\overline{S}_i$). In such a case we say that $\overline{S}$ is good for ω, and that ω is good for $\overline{S}$. For starters, note that for every fixed $\omega \in \{0,1\}^r$, the probability that a uniformly distributed $\overline{S} \in [n]^{qt}$ is good for ω equals $(1-(1-(1/n))^t)^q \approx (t/n)^q$. We shall show that, with high probability, such a random $\overline{S}$ is good for a $(1 \pm o(1)) \cdot (t/n)^q$ fraction of the $\omega \in \{0,1\}^r$. This will imply that if the sample-based tester selects at random an ω that is good for the sample that it receives, and emulates T using ω as its random coins (while answering T's queries by using the corresponding labels), then this tester emulates the POT quite well. We start with the first claim.

Claim 3.1.1 *Let $\delta > 0$ and $\mu = (1-(1-(1/n))^t)^q$. If $\overline{S}$ is uniformly distributed in $[n]^{qt}$, then with probability at least $1 - \frac{q}{\alpha\delta^2\mu n}$ the sequence $\overline{S}$ is good for a $(1 \pm \delta) \cdot \mu$ fraction of the $\omega \in \{0,1\}^r$.*

Proof: For every $\omega \in \{0,1\}^r$, we denote by $\zeta_\omega = \zeta_\omega(\overline{S})$ the indicator random variable that is 1 if and only if $(\omega, \overline{S})$ is good. Recall that $\mathrm{E}[\zeta_\omega] = \Pr_{\overline{S} \in [n]^{qt}}[\zeta_\omega(\overline{S}) = 1] = \mu$. Letting $N \stackrel{\text{def}}{=} 2^r$ and $\overline{\zeta}_\omega \stackrel{\text{def}}{=} \zeta_\omega - \mu$, the claim asserts that

$$\Pr_{\overline{S} \in [n]^{qt}}\left[\left|\sum_{\omega \in \{0,1\}^r} \overline{\zeta}_\omega\right| > \delta \cdot \mu N\right] < \frac{q}{\alpha\delta^2\mu n} \,. \tag{1}$$

We prove Eq. (1) by using Chebyshev's Inequality:

$$\begin{aligned} \Pr_{\overline{S} \in [n]^{qt}}\left[\left|\sum_{\omega \in \{0,1\}^r} \overline{\zeta}_\omega\right| > \delta \cdot \mu N\right] &< \frac{\mathrm{E}\left[\left(\sum_{\omega \in \{0,1\}^r} \overline{\zeta}_\omega\right)^2\right]}{(\delta\mu N)^2} \\ &= \frac{1}{\delta^2\mu^2N^2} \cdot \sum_{\omega_1,\omega_2 \in \{0,1\}^r} \mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}] \,. \end{aligned} \tag{2}$$

[7] Indeed, this tester is reminiscent of the notion of a relaxed POT, as presented in Definition A.1, but we shall proceed without any definition regarding this matter.

We partition the sum in Eq. (2) into two sums separating pairs that generate intersecting sequences of queries from pairs that generate non-intersecting sequences of queries, where the sequences $(u_1, \ldots, u_q)$ and $(v_1, \ldots, v_q)$ intersect if there exists an $i \in [q]$ such that $u_i = v_i$.

We first note that the fraction of intersecting pairs is at most $q/\alpha n$, since for any fixed sequence $\overline{u} = (u_1, \ldots, u_q) \in [n]^q$ the sequence of queries generated by a uniformly distributed $\omega \in \{0,1\}^r$ intersects $\overline{u}$ with probability at most $q \cdot (1/\alpha n)$. On the other hand, for any pair (ω_1, ω_2), it holds that $\mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}] < \mathrm{E}[\zeta_{\omega_1}\zeta_{\omega_2}] \leq \mathrm{E}[\zeta_{\omega_1}] = \mu$. This bound is far from being tight. In particular, as shown next, $\mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}]$ is negative for any *non-intersecting* pair (ω_1, ω_2). Indeed, considering such a non-intersecting pair, note that $\mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}] = \Pr[\zeta_{\omega_1} = \zeta_{\omega_2} = 1] - \mu^2$, whereas $\Pr[\zeta_{\omega_1} = \zeta_{\omega_2} = 1] = \mu \cdot \Pr[\zeta_{\omega_2} = 1 | \zeta_{\omega_1} = 1]$ and $\Pr[\zeta_{\omega_2} = 1 | \zeta_{\omega_1} = 1] = (1 - (1 - (1/n))^{t-1})^q < (1 - (1 - (1/n))^t)^q = \mu$. Combining all the above facts, we upper bound Eq. (2) by

$$\frac{1}{\delta^2 \mu^2 N^2} \cdot \left(\frac{q}{\alpha n} \cdot N^2\right) \cdot \mu = \frac{q}{\delta^2 \mu \alpha n}$$

and the claim follows. ■

We set δ so as to equate the two errors in Claim 3.1.1; that is, we set $\delta = \frac{q}{\delta^2 \mu \alpha n}$, which implies $\mu n = 1/\alpha\delta^3 = \Theta(1/\delta^3)$. Indeed, this requires that $\mu n > 1$. Since $\mu = (1 - (1 - (1/n))^t)^q < (t/n)^q$, this implies that $t > n^{(q-1)/q}$. Using Claim 3.1.1, the analysis of the sample-based tester outlined above reduces to the following claim.

Claim 3.1.2 *Let $G = ((X, Y), E)$ be a bipartite graph such that each vertex in X has degree $|E|/|X|$ and at least a $1-\delta$ fraction of the vertices in Y have degree $(1\pm\delta)\cdot|E|/|Y|$, where $\delta \in [0, 0.5)$. Then, uniformly selecting a vertex y in Y, and uniformly selecting a neighbor of y, yields a distribution that is 2δ-close to the uniform distribution on X.*

(We shall apply Claim 3.1.2 with $X = \{0,1\}^r$ and $Y = [n]^{qt}$, where there is an edge between $x \in X$ and $y \in Y$ if x is good for y.)

Proof: Let Y' be the set of vertices in Y that have degree $(1 \pm \delta) \cdot |E|/|Y|$, and let E' be the set of edges with an endpoint in Y'. Then, $|E'| \geq (1 - \delta) \cdot |Y| \cdot (1 - \delta) \cdot |E|/|Y| > (1 - 2\delta) \cdot |E|$, and each edge in E' is selected with probability $\frac{1}{|Y|} \cdot \frac{1}{(1\pm\delta)\cdot|E|/|Y|} = \frac{1}{(1\pm\delta)\cdot|E|}$, since an edge incident to vertex $y \in Y$ that has degree d_y is selected with probability $\frac{1}{|Y|} \cdot \frac{1}{d_y}$. Hence, the distribution induced on E is δ'-close to uniform, where $\delta' < (|(1 \pm \delta)^{-1} - 1| + 2\delta)/2 < 2\delta$.[8] The claim follows, since the distribution on X is induced by a function of the distribution on E that maps the uniform distribution on E to the uniform distribution on X.[9] ■

The sample-based tester (formalized): On input parameters n, ϵ and a sequence of labeled samples $((s_1, v_1), \ldots, (s_{qt}, v_{qt}))$, where $t = \Theta(n^{1-(1/q)}/\varrho(\epsilon)^{3/q})$, the tester proceeds as follows, with the intention of emulating a single execution of T with statistical deviation of at most $\varrho(\epsilon)/3$.

1. The tester selects uniformly $\omega \in \{0,1\}^r$ such that ω is good for $\overline{S}$, where $\overline{S} = (s_1, \ldots, s_{qt})$.

   If no such ω exists, then the tester halts and outputs 1 with probability c (and outputs 0 otherwise), where c is the threshold probability of T. (This bad event occurs with probability at most $\varrho(\epsilon)/10$.)

   (The next step is executed only if ω was selected in Step 1.)

[8] Specifically, the said variation distance is at most $\frac{1}{2} \cdot \left(\sum_{e\in E'} \left|\frac{1}{(1\pm\delta)|E|} - \frac{1}{|E|}\right| + \max\left(\frac{|Y\setminus Y'|}{|Y|}, \frac{|E\setminus E'|}{|E|}\right)\right)$, since (for every two random variables ζ and ζ', and every set S) the variation distance between ζ and ζ' is upper-bounded by $\frac{1}{2} \cdot \left(\sum_{e\in S} |\Pr[\zeta = e] - \Pr[\zeta' = e]| + \max(\Pr[\zeta \notin S], \Pr[\zeta' \notin S])\right)$.

[9] We use the fact that, for every two random variables ζ and ζ', and every function f, the variation distance between $f(\zeta)$ and $f(\zeta')$ is upper-bounded by the variation distance between ζ and ζ'.

2. Let $(i_1,\ldots,i_q) \in [qt]$ be such that $(s_{i_1},\ldots,s_{i_q})$ is the sequence of queries generated by ω.

   The tester emulates T using randomness ω, while using v_{i_j} as the answer to the j^{th} query, and outputs whatever T does.

Note that for our choice of $t = \Theta(n^{1-(1/q)}/\varrho(\epsilon)^{3/q})$, it holds that $\mu n \approx (t/n)^q \cdot n = \Theta(1/\varrho(\epsilon)^3)$, and so (by Claims 3.1.1 and 3.1.2) this emulation of T deviates from a perfect one by at most $\varrho(\epsilon)/3$. The final sample-based tester is obtained by running $O(1/\varrho(\epsilon)^b)$ copies of the above emulation, where $b = 2$ for the general case and $b = 1$ for the one-sided error case. Details follow.

On input n, ϵ and $((s_1,v_1),\ldots,(s_{qt'},v_{qt'})$, where $t' = \Theta(t/\varrho(\epsilon)^b) = \Theta(n^{1-(1/q)}/\varrho(\epsilon)^{b+(3/q)})$, the tester runs $O(1/\varrho(\epsilon)^b)$ copies of the above emulation, when using the subsequence of labeled samples $((s_{(i-1)t+1}, v_{(i-1)t+1}),\ldots,(s_{it},v_{it})$, in the i^{th} emulation. In the general case the tester outputs 1 if and only if at least $c - \varrho(\epsilon)/2$ of the emulations returned 1, whereas in the one-sided error case the tester outputs 1 if and only if all the emulations returned 1. ■

3.2 On the fairness condition

The fairness condition is essential to the foregoing proof of Theorem 3.1. Indeed, if there exists $i \in [n]$ such that the given POT always makes the query i, then we have no chance to emulate it while using a random sample of size $o(n)$. One might conjecture that *any property that has a q-query POT, also has a q-query POT that is $\Omega(1)$-fair*; but as shown next this conjecture is false.

Proposition 3.2 (POTs do not necessarily imply fair POTs): *There exists a property Π that has a* (one-sided error) *three-query POT with linear detection probability, but has no constant-query POT that is $\Omega(1)$-fair.*

Proposition 3.2 does not mean that Π has no sample-based tester of sublinear complexity; in fact, the property used in the following proof does have such a tester. We shall show later (see Proposition 3.3) that there exists a property Π that has a constant-query POT but does not have a sample-based tester of sublinear complexity.[10]

Proof: The proof is by a reduction to results regarding MA-POTs, a notion introduced by Gur and Rothblum [21]. Loosely speaking, an MA-POT is a POT in which the tester obtains an explicit (short) proof (or witness) in addition to oracle access to the input function f. It is required that for every $f \in \Pi$ there exists a witness w_f (of the designated length) such that on input f and w_f the tester always accepts, whereas if f is not in Π then the tester rejects f with probability that is related to the distance of f from Π (as in Definition 2.2 for the case $c = 1$) no matter what w is provided as an alleged witness. We stress that the tester has free access to the witness, and the query complexity only accounts for the access to f.

Gur and Rothblum [21] showed a q-query MA-POT that utilizes proofs of logarithmic length for a class of Boolean functions having no (standard) property tester of query complexity $n^{1-\Omega(1/\sqrt{q})}$.[11] For our purposes, we shall use a simpler two-query MA-POT that utilizes proofs of logarithmic length for a class of Boolean functions having no (standard) property tester of query complexity $\Omega(\sqrt{n})$. Such a result is mentioned in [8][12], but we shall present an alternative one. In any case, let us detail the reduction first.

[10] Indeed, Proposition 3.3 implies Proposition 3.2, alas it uses a more contrived property (which consists of functions having a huge range).

[11] We refer to [21, Thm. 3.1] and note that its proof implicitly provides a constant-query POT of detection probability $\varrho(\epsilon) = \exp(-1/\epsilon)$. (Actually, the detection probability is $\Omega(\epsilon)$ if $\epsilon > 1/\log n$ (and at least $1/n$ otherwise).)

[12] We refer to their observation that the property $\Pi = \{uu^{\text{R}}vv^{\text{R}} : u,v \in \{0,1\}^*\}$ has a constant-query MA-POT of logarithmic proof complexity, whereas any standard tester for Π requires $\Omega(\sqrt{n})$ queries [1].

Claim 3.2.1 *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$, where Π_n contains functions from $[n]$ to $\{0,1\}^{\ell(n)}$. Suppose that Π has a q-query MA-POT with proof complexity $\ell(n)$, but has no standard tester of query complexity that only depends on the proximity parameter. Then, there exists a property Π' that has a $(q+1)$-query POT but no constant-query POT that makes $\Omega(1)$-fair queries. Furthermore, $\Pi' = \cup_{n\in\mathbb{N}}\Pi'_{n+1}$ such that Π'_{n+1} contains functions from $\{0,1,\ldots,n\}$ to $\{0,1\}^{\ell(n)}$.*

Proof: We define Π'_{n+1} such that $f' \in \Pi'_{n+1}$ if there exists $f \in \Pi_n$ such that $f'(i) = f(i)$ for every $i \in [n]$ and $f'(0) = w$ for any w that makes the guaranteed MA-POT accept f with probability 1. Note that at least one such w exists.

We first observe that Π' has no constant-query POT that makes $\Omega(1)$-fair queries. The reason is that an $\Omega(1)$-fair POT may only query the function f' at 0 with probability $O(1/n)$, whereas it must reject functions that are at constant distance from Π' with constant probability. Hence, it could be modified to a POT that never queries f' at 0 and still rejects functions that are at constant distance from Π' with constant probability. But this would have yielded a standard tester with performance that violates the hypothesis.

We now present a $(q+1)$-query POT for Π'. This POT always queries f' at 0, and invokes the MA-POT while providing it with oracle access to the corresponding f (i.e., f' restricted to $[n]$) and a witness that equals $f'(0)$. Note that this POT accepts each $f' \in \Pi'$ with probability 1, whereas $f' \notin \Pi'$ is rejected with probability that is related to the distance of f from Π (where in case $f \in \Pi$, the function f' is rejected with probability at least $2^{-\rho(n)}$, where ρ is the randomness complexity of the MA-POT, which is fine since such f' is at distance $1/(n+1)$ from Π'_{n+1}).[13] ■

Claim 3.2.2 *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$, where Π_n contains functions from $[n]$ to $[n]$ such that $f \in \Pi_n$ if and only if there exists $i \in [\lfloor n/2 \rfloor]$ such that for every $j \in [\lfloor n/2 \rfloor]$ it holds that $f(\lfloor n/2 \rfloor + j) = f(\mathrm{sh}_i(j))$, where $\mathrm{sh}_i(j) \stackrel{\rm def}{=} ((j+i) \bmod \lfloor n/2 \rfloor) + 1$. Then, Π has a two-query MA-POT with proof complexity $\log_2 n$, but any standard tester for Π has query complexity $\Omega(\sqrt{n})$.*[14]

Proof: Let $m \stackrel{\rm def}{=} \lfloor n/2 \rfloor$. The two-query MA-POT gets $i \in [m]$ as a witness, selects j uniformly in $[m]$ and compares $f(m+j)$ to $f(\mathrm{sh}_i(j))$, by making the corresponding queries. This MA-POT has linear detection probability. Turning to the lower bound, we show that a machine of query complexity $o(\sqrt{n})$ cannot distinguish a function uniformly selected in Π_n from a totally random function (augmented by a random choice of $i \in [m]$). Towards this end, we consider a matching of $[m+1, 2\cdot m]$ to $[m]$ such that $m+j$ is matched to $\mathrm{sh}_i(j)$. The point is that in the former case (i.e., of $f \in \Pi$), as long as the queries made contain no matched pair, the answers are uniformly distributed in $[n]$ (just as in the case of a totally random function). ■

Combining the two claims, the proposition follows. Note that the property that we obtained (i.e., the one satisfying the proposition) is a property of functions with a range that equals their domain. ■

Proposition 3.3 (POTs do not necessarily imply testers with sublinear sample complexity): *There exists a property Π that has a one-sided error two-query POT with linear detection probability, but has no sample-based tester of sublinear sample complexity.*

[13] Specifically, if the detection probability function of the MA-POT for Π is ϱ, then the detection probability function of the POT for Π' is $\varrho'(\epsilon) = \min\{\varrho(\epsilon), 2^{-\rho(1/\epsilon)}\}$. We comment that, w.l.o.g., we may have $\rho(n) = O(\log n)$ (cf., e.g., [21, Lem. 4.6]).

[14] This lower bound is tight, since Π has a standard tester of query complexity $O(\sqrt{n})$. In fact, Π has a sample-based tester of this complexity, which tries to find i by comparing the samples of the first and second part of $[n]$.

Proposition 3.3 is proved by using a class of functions with a huge range (i.e., $|R_n| = \exp(n)$); we wonder whether the result holds also for a class of functions with significantly smaller range (e.g., $|R_n| = n$ or maybe even $|R_n| = 2$).

Proof: Starting from any hard to test property Π' (cf., e.g., [14, Sec. 4.1]), we consider the property Π such that $f \in \Pi_{n+1}$ if and only if there exists $f' \in \Pi'_n$ such that $f(i) = f'(i)$ for every $i \in [n]$ and $f(0) = \langle f' \rangle$, where $\langle f' \rangle$ denotes the full description of $f' : [n] \to R'_n$. Indeed, we use $D_{n+1} = \{0, 1, \ldots, n\} = \{0\} \cup D'_n$ and $R_{n+1} = R'_n \cup (R'_n)^n$. (We could have presented the proof by using the connection to MA-POTs, as done in the proof of Proposition 3.2, but prefer to present a direct argument.)[15]

We first show that Π has a two-query POT with one-sided error and detection probability $\varrho(\epsilon) \geq \epsilon/2$. Given access to a function $f : D_{n+1} \to R_{n+1}$, we query f at 0 and at a uniformly distributed $i \in [n]$, obtaining the values v and u, which may be assumed to reside in $(R'_n)^n$ and R'_n respectively (or else we reject). We accept if and only if v, viewed as a function from $[n]$ to R'_n, is in Π'_n and u equals the i^{th} element in the n-long sequence v.[16] This machine accept any $f \in \Pi_{n+1}$ with probability 1. On the other hand, suppose that f is ϵ-far from Π_{n+1}, and let f' denote the restriction of f to $[n]$. We first discard of the case that $\epsilon = 1/(n+1)$, noting that in this case the machine rejects with probability at least ϵ. Now, using $\epsilon \geq 2/(n+1)$, we infer that f' is ϵ'-far from Π'_n, where $\epsilon' = ((n+1)\epsilon - 1)/n > \epsilon/2$. We may assume, w.l.o.g, that $f(0) \in \Pi'_n$, which implies that there are at least $\epsilon n/2$ indices $i \in [n]$ such that $f(i) = f'(i)$ differs from the i^{th} element in the n-long sequence $f(0)$. In this case, the machine rejects with probability at least $\epsilon/2$.

Finally, we claim that Π has no sample-based tester of sublinear sample complexity. The reason is that a sublinear sized sample is unlikely to yield the value of $f(0)$, whereas without this value testing is hard (i.e., requires a linear number of queries, let alone samples). ■

A sufficient condition for fairness. While Propositions 3.2 and 3.3 assert that not every POT can be transformed into a fair one, we next state a sufficient condition for such a transformation. We warn, however, that this condition is not necessary. In fact, any POT in the bounded-degree graph model can be transformed into a fair one (cf. [18, Clm. 5.5.2]), although the condition stated next does not hold.[17]

Proposition 3.4 (invariance under 1-transitivity implies 1-fairness): *Suppose that the property Π is invariant under a 1-transitive permutation group that acts on its domain; that is, for every $n \in \mathbb{N}$ and $i, j \in D_n$, there exists a permutation $\pi : D_n \to D_n$ such that $\pi(i) = j$ and for every $f : D_n \to R_n$ it holds that $f \in \Pi_n$ if and only if $f \circ \pi \in \Pi_n$, where $(f \circ \pi)(x) = f(\pi(x))$. If Π has a nonadaptive q-query POT, then it has one that is 1-fair, and if the former has one-sided error then so does the latter.*

For constant-size range (i.e., $|R_n| = O(1)$), non-adaptivity can be obtained at the cost of decreasing the detection probability by a factor that is exponential in q. (This can be done by guessing the answers at random beforehand, determining the corresponding queries and making them non-adaptively, outputting the verdict of the original POT if all guesses turn out to be correct, and outputting 1 with probability c otherwise (i.e., if any of these guesses turns out to be wrong).)

[15] Such an alternative presentation would proceed by showing that Π has a one-query MA-POT of proof complexity $n \log |R'_n|$.

[16] This machine corresponds to an MA-POT for Π', which is given $\langle f' \rangle$ as a witness, and makes a single query obtaining the value $f'(i)$ (and acting accordingly).

[17] Graph properties in this model are not invariant under any 1-transitive permutation group that acts on their domain, and (in general) the corresponding POTs are adaptive [28].

Proof: Consider the group G_n generated by the permutations guaranteed in the hypothesis. Given a q-query POT T for Π, consider the machine T' that, when given oracle access to a function $f : D_n \to R_n$, selects $\pi \in G_n$ uniformly at random, and invokes T when providing it with oracle access to $f \circ \pi$; that is, query j is answered by querying f at $\pi(j)$. Then, T' is a POT for Π, since $\delta_\Pi(f \circ \pi) = \delta_\Pi(f)$ by the invariance of Π under the permutation in G. On the other hand, by the 1-transitivity of G, a random $\pi \in G$ maps each $i \in D_n$ uniformly over D_n (i.e., $\Pr_{\pi \in G}[\pi(i) = j] = 1/n$ for every $j \in D_n$). Hence, each of the queries of T' is uniformly distributed in D_n, where here we rely on the non-adaptivity of T. ■

We note that non-adaptivity is essential to the proof of Proposition 3.4. Consider, for example, the property $\Pi_n = \{f_i : i \in [n]\}$ such that $f_i : \mathbb{Z}_n \to \mathbb{Z}_n$ satisfies $f_i(j) = j + i \bmod n$. This property is 1-transitive, but is only invariant under cyclic permutations (i.e., permutations of the form $\pi_s(i) = i + s$). Now consider a three-query adaptive POT T that, uniformly selects $r \in \mathbb{Z}_n$, queries its oracle f on $0, r$, and $v = r - f(r)$, and accepts if and only if $f(0) = f(r) - r$. (Indeed, T ignores the third answer, and its analysis reduces to observing that it accepts f on coins r if and only if $f(r) = f(0) + r$.) Then T', as defined in the foregoing proof, select $s \in \mathbb{Z}_n$ uniformly at random, and queries $f = f_i$ on $\pi_s(0)$, $\pi_s(r)$ and $\pi_s(v)$, whereas $\pi_s(v) = \pi_s(r - f_i(\pi_s(r))) = (r - ((r+s)+i)) + s = i$, which strongly violates the fairness condition (since the third query is totally determined by f_i). Of course this does not mean that Π_n has no POT that is non-adaptive; in fact, this Π_n has a sample-based POT that uses two samples (i.e., given $(r_1, f(r_1))$ and $(r_2, f(r_2))$, the POT accepts if and only if $f(r_1) - f(r_2) = r_1 - r_2$).[18]

3.3 A generalization (of Theorem 3.1)

For any $k \in [q]$, we consider POTs that make q queries such that any k of these queries are (almost) uniformly distributed in D_n^k. Indeed, the case of $k = 1$ is handled in Theorem 3.1 and the case of $k = q$ is straightforward, but here we are interested in the intermediate cases. For example, the basic test that underlies the Linearity Tester of [5] selects three elements in the domain such that any pair are uniformly and independently distributed. In general, for $\alpha \in (0, 1]$, we say that a POT is (k, α)-fair if, *for every* $1 \le i_1 < i_2 < \cdots < i_k \le q$ *and* $e_1, \ldots, e_k \in D_n$ (and for every $f : D_n \to R_n$), *with probability at most* $1/\alpha n^k$ *it holds that for every* $j \in [k]$ *the* i_j^{th} *query of the tester equals* e_j (when it accesses the oracle f).

Theorem 3.5 (Theorem 3.1, generalized): *Suppose that* Π *has a* q*-query POT with detection probability* ϱ *that is* $(k, \Omega(1))$*-fair. Then,* Π *has a sample-based tester of sample complexity* $s(n, \epsilon) = \max(O(n^{1-(k/q)}/\varrho(\epsilon)^{b+(3/q)}), O(\varrho(\epsilon)^{-(3+b)}))$*, where* $b = 2$ *in the general case and* $b = 1$ *if POT has one-sided error. Furthermore, if the POT has one-sided error, then so does the sample-based tester.*

Applying Theorem 3.5 to the 3-query POT for Linearity (cf. [5]), we obtain a (one-sided error) sample-based tester of sample complexity $s(n, \epsilon) = \text{poly}(1/\epsilon) \cdot n^{1/3}$. Note that this sample complexity is optimal as an emulation of this POT, since a sample of $o(n^{1/3})$ group elements is unlikely to contain a triplet that sums-up to zero, but more efficient sample-based tester can be obtained directly (see Section 5).

Proof: We proceed as in the proof of Theorem 3.1, while focusing on strengthening Claim 3.1.1 under the stronger fairness hypothesis made here. Specifically, we partition the sum in Eq. (2)

[18] This POT has detection probability $\varrho(\delta) = \delta$. This is shown by noting that if this POT accepts with probability p, then there is a choice of r_1 for which it accepts with probability at least p, and it follows that f is $(1 - p)$-close to Π_n (just as in the analysis of the foregoing T).

into $k+1$ sums separating pairs according to the number of intersections that occur in these pairs. More specifically, we say that the sequences $(u_1,\dots,u_q)$ and $(v_1,\dots,v_q)$ are j-intersecting if $|\{i\in[q] : u_i=v_i\}|=j$. (Indeed, the proof of Claim 3.1.1 only distinguishes 0-intersecting pairs from intersecting pairs (i.e., pairs that j-intersect for some $j\in[q]$).) Letting I_j denote the set of j-intersecting pairs, we have:

$$\begin{aligned}
&\sum_{\omega_1,\omega_2\in\{0,1\}^r} \mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}] \\
&\quad= \sum_{(\omega_1,\omega_2)\in I_0} \mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}] + \sum_{j\in[k]}\sum_{(\omega_1,\omega_2)\in I_j} \mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}] + \sum_{(\omega_1,\omega_2)\in\cup_{j>k}I_j} \mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}]\,. \qquad (3)
\end{aligned}$$

Recall that the contribution of the 0-intersecting pairs is negative (see the proof of Claim 3.1.1). Now, for every $j\in[k]$, the fraction of j-intersecting pairs is at most $\binom{q}{j}/\alpha n^j$, whereas each j-intersecting pair (ω_1,ω_2) contributes $\mu\cdot(1-(1-(1/n))^{t-1})^{q-j} < \mu^2\cdot(1-(1-(1/n))^t)^{-j}$, since $\mathrm{E}[\overline{\zeta}_{\omega_1}\overline{\zeta}_{\omega_2}] < \Pr[\zeta_{\omega_1}=\zeta_{\omega_2}=1]$ and $\Pr[\zeta_{\omega_2}=1|\zeta_{\omega_1}=1] = (1-(1-(1/n))^{t-1})^{q-j}$. Similarly, the fraction of pairs in $\cup_{j>k}^{q} I_j$ is at most $\binom{q}{k}/\alpha n^k$, but we can only upper bound the contribution of each such pair by μ. Since $t=s(n,\epsilon)/q=o(n)$, we have $(1-(1-(1/n))^{t-1})^{-j} < 2\cdot(t/n)^{-j}$, and so Eq. (3) is upper bounded by

$$\begin{aligned}
&\sum_{j\in[k]} \frac{\binom{q}{j}N^2}{\alpha n^j}\cdot 2\mu^2\cdot(t/n)^{-j} + \frac{\binom{q}{k}N^2}{\alpha n^k}\cdot\mu \\
&\quad< \;2^{q+1}\cdot N^2\cdot\left(\frac{\mu^2}{\alpha t}+\frac{\mu}{\alpha n^k}\right)\,. \qquad (4)
\end{aligned}$$

where $N=2^r$ (as in the proof of Claim 3.1.1). Combining Eq. (2) and Eq. (4), it follows that

$$\begin{aligned}
\Pr_{\overline{S}\in[n]^{qt}}\left[\left|\sum_{\omega\in\{0,1\}^r}\overline{\zeta}_\omega\right| > \delta\cdot\mu N\right] \;&<\; \frac{2^{q+1}}{\delta^2\mu^2}\cdot\left(\frac{\mu^2}{\alpha t}+\frac{\mu}{\alpha n^k}\right) \\
&=\; \frac{2^{q+1}}{\delta^2\alpha t}+\frac{2^{q+1}}{\delta^2\mu\alpha n^k}\,. \qquad (5)
\end{aligned}$$

Wishing to upper bound each of the terms in Eq. (5) by $\delta/2$, we get $t\ge 2^{q+2}/\alpha\delta^3$ and $\mu n^k \ge 2^{q+2}/\alpha\delta^3$. By setting $t=\max(O(n^{1-(k/q)}/\delta^{3/q}), O(1/\delta^3))$, both inequalities are satisfied. Continuing as in the proof of Theorem 3.1, the current theorem follows. ■

Remark 3.6 (applicability to standard testers): *The ideas underlying the proofs of Theorem 3.1 and 3.5 can be applied also to standard testers. In this case the query complexity q is not constant, but rather depends on the proximity parameter ϵ and possibly also on the size parameter n. On the other hand, the detection probability* (represented by the function ϱ) *is a constant* (i.e., 1/3)*. The most appealing version refers to the case that the query complexity only depends on ϵ, in which case we get:*

> *Suppose that Π has a $\Omega(1)$-fair tester of query complexity $q:[0,1]\to\mathbb{N}$ that only depends on the proximity parameter. Then, Π has a sample-based tester of sample complexity $s(n,\epsilon)=O(n^{1-(1/q(\epsilon))})$. Furthermore, if the original tester has one-sided error, then so does the sample-based tester.*

Hence, sample-based testers of sublinear sample complexity exist for many natural properties.

4 Sample-based testers in the dense graphs model

In this section, we consider testing graph properties in the adjacency matrix model (a.k.a the dense graphs model). In this model a graph $G = (V, E)$ is represented by the Boolean function $g : \binom{V}{2} \to \{0,1\}$ such that $g(u,v) = 1$ if and only if u and v are adjacent in G (i.e., $\{u,v\} \in E$). Thus, the domain size n equals $\binom{|V|}{2}$, and the distance between graphs is measured in terms of their aforementioned representation (i.e., as the fraction of (the number of) unordered vertex pairs on which they differ over n). Here we only consider *graph properties*, which are sets of graphs that are closed under isomorphism; that is, Π is a graph property if for every graph $G = (V, E)$ and every permutation π of V it holds that $G \in \Pi$ if and only if $\pi(G) \in \Pi$, where $\pi(G) \stackrel{\text{def}}{=} (V, \{\{\pi(u), \pi(v)\} : \{u,v\} \in E\})$.

We shall sometimes view E as a set of ordered pairs (rather than a set of unordered ones). Ditto regarding the sequence of samples (i.e., vertex pairs).

4.1 General results

The starting point of our general study of sample-based testers in the dense graphs model is a transformation of standard testers (in this model) into "canonical" ones [20, Thm. 2]. In fact, we shall use part of this transformation, which yields "quasi-canonical" testers as defined next.

Definition 4.1 (quasi-canonical testers): *A* quasi-canonical tester of vertex complexity $\nu : \mathbb{N} \times [0,1] \to \mathbb{N}$ for a graph property Π *is a tester of* Π (in the dense graphs model) *that operates as follows:*

1. *On input parameters $n = \binom{|V|}{2}$ and ϵ, it select uniformly a set of $\nu(n, \epsilon)$ vertices, denoted S, and queries all the vertex pairs* (i.e., all pairs in $\binom{S}{2}$)*.*
2. *Let H be the unlabeled induced graph that corresponds to the answers obtained in Step 1. The tester makes its decision based solely on n, ϵ and H, and possibly on fresh coin tosses* (but not on the coin tosses used to select the sample S).

Definition 4.1 differs from the definition of canonical testers in [20] only in that Step 2 is allowed to be randomized.

Theorem 4.2 (Combining Lemma 4.1 and Claim 4.3 in [20]): *Let* Π *be a graph property and suppose that* Π *has query complexity* $q : \mathbb{N} \times [0,1] \to \mathbb{N}$ (in the dense graphs model)*. Then* Π *has a quasi-canonical tester of vertex complexity* $2q$*.*

(The canonical tester of [20] is obtained by an additional step, which increases the complexity by a constant factor. We avoid this step, because we do not need it.)

Theorem 4.3 (from quasi-canonical testers to sample-based testers): *Let* Π *be a graph property and suppose that* Π *has a quasi-canonical tester of vertex complexity* $\nu : \mathbb{N} \times [0,1] \to \mathbb{N}$*. Then,* Π *has a sample-based tester of sample complexity* $s(n, \epsilon) = O(n^{1-1/(\nu(n,\epsilon)-1)})$*.*

In particular, if $\nu(n, \epsilon) = \nu'(\epsilon)$ does not depend on n, then for every constant ϵ we obtain sublinear sample complexity.

Proof: A random sample of size $s = s(n, \epsilon)$ corresponds to selecting each sample point (i.e., a vertex pair) with probability $p = s/n$, independently of all other choices.[19] Recall that $n = \binom{|V|}{2} \approx |V|^2/2$. Call a set of vertices S good for the sample if the sample contains all vertex pairs that correspond to this set S (i.e., for every two distinct vertices $u, v \in S$, it holds that the pair $\{u, v\}$ is in the sample). Then, the expected number of vertex sets of size $\nu = \nu(n, \epsilon)$ that are good for a random sample (selected as above), denoted μ, is given by

$$\begin{aligned} \binom{|V|}{\nu} \cdot p^{\binom{\nu}{2}} &> (\sqrt{2n}/\nu)^{\nu} \cdot p^{\nu \cdot (\nu-1)/2} \\ &= \left((2n/\nu^2) \cdot (s/n)^{\nu-1}\right)^{\nu/2} \ , \end{aligned}$$

which is greater than 1 if $s > 2 \cdot n^{1-1/(\nu-1)}$. As shown next, with probability at least 0.99, a random sample of $O(n^{1-1/(\nu-1)})$ vertex pairs contains a good set of ν vertices, and each such good set is uniformly distributed among all vertex sets of size ν.

For every set of ν vertices, S, let ζ_S denote the indicator random variable representing whether S is good for a random sample, and let $\overline{\zeta}_S = \zeta_S - \mathrm{E}[\zeta_S]$. Letting $N \stackrel{\rm def}{=} \binom{|V|}{\nu}$ and $\mu = \mathrm{E}[\zeta_S] = p^{\nu \cdot (\nu-1)/2}$, the claim follows by applying Chebyshev's Inequality: Starting with

$$\Pr\left[\sum_{S \in \binom{V}{\nu}} \zeta_S > 0\right] < \frac{\mathrm{E}\left[\left(\sum_{S \in \binom{V}{\nu}} \overline{\zeta}_S\right)^2\right]}{(\mu N)^2}$$

we use the fact that $\mathrm{E}[\overline{\zeta}_S \overline{\zeta}_T] = 0$ if $S \cap T = \emptyset$ (and in fact also if $|S \cap T| = 1$). For $|S \cap T| \geq 1$, we use $\mathrm{E}[\overline{\zeta}_S \overline{\zeta}_T] < \mathrm{E}[\zeta_S \zeta_T] = p^{\binom{|S|}{2} + \binom{|T|}{2} - \binom{|S \cap T|}{2}}$. Hence,

$$\begin{aligned} \sum_{S,T \in \binom{V}{\nu}} \mathrm{E}[\overline{\zeta}_S \overline{\zeta}_T] &= \sum_{i \in [\nu]} \sum_{S,T \in \binom{V}{\nu} : |S \cap T| = i} \mathrm{E}[\overline{\zeta}_S \overline{\zeta}_T] \\ &< \sum_{i \in [\nu]} \binom{|V|}{2\nu - i} \cdot \binom{2\nu - i}{\nu} \cdot \binom{\nu}{i} \cdot p^{2\binom{\nu}{2} - \binom{i}{2}} \\ &< \sum_{i \in [\nu]} O(|V|/\nu)^{2\nu - i} \cdot p^{(2\nu - i) \cdot (\nu - 1)/2} \\ &= \exp(\nu) \cdot \sum_{i \in [\nu]} \left((|V|/\nu) \cdot p^{(\nu-1)/2}\right)^{2\nu - i} \ , \end{aligned}$$

which is significantly smaller than $(N\mu)^2 > ((|V|/\nu) \cdot p^{(\nu-1)/2})^{2\nu}$, since $|V| \cdot p^{(\nu-1)/2} > \exp(\nu)$.

Having shown that, with probability at least 0.99, a random sample of $O(n^{1-1/(\nu-1)})$ vertex pairs contains a good set of ν vertices, we note that the subgraph induced by such a set is visible in the sample. Noting that each such good set is uniformly distributed among all vertex sets of size ν, it begs to emulate the quasi-canonical tester using this set. Indeed, our sample-base tester selects at random a good set, uniformly among all ν-vertex sets that are good for the given sample, and emulate the quasi-canonical tester using the induced subgraph (which is visible in the sample). ■

[19] Actually, the correspondence is not exact, but it suffices for our purposes. See Appendix A.2. The issue at hand here is the difference between picking m elements uniformly in $[n]$ and picking each element in $[n]$ with proability m/n.

Quasi-canonical proximity oblivious testers. Using the same ideas, we can obtain analogous results for proximity oblivious testers (in the dense graphs model). We first need an analogue definition of quasi-canonical POTs. Such a definition is obtained by combining Definitions 2.2 and 4.1. Specifically, a quasi-canonical POT of vertex complexity ν is a POT that operates as in Definition 4.1, which means that ν is a constant. Analogously to Theorem 4.2, it follows that every q-query POT yields a quasi-canonical POT of vertex complexity $2q$ that maintains the performance guarantees of the original POT, where this fact is only implicit in [20, Sec. 4]. Using the ideas underlying the proof of Theorem 4.3, we get the first item in Theorem 1.2, restated next.

Theorem 4.4 (from quasi-canonical POTs to sample-based testers): *Suppose that Π has a quasi-canonical POT of vertex complexity ν and detection probability ϱ. Then, Π has a sample-based tester of sample complexity $s(n,\epsilon) = O(n^{1-1/(\nu-1)}/\varrho(\epsilon)^2)$. Furthermore, if the POT has one-sided error, then so does the sample-based tester* (and the sample complexity can be reduced to $O(n^{1-1/(1-\nu)}/\varrho(\epsilon))$).

(The $\varrho(\epsilon)^{-b}$ factor, for $b = 1, 2$, arises from the need to repeat the resulting "sample-based POT" for an adequate number of times so as to derive a tester.) Note that Theorem 4.4 improves upon Theorem 3.1 whenever we are given a quasi-canonical POT, since the query complexity of such a POT is quadratic in its vertex complexity. In general, we get an improvement whenever we are given a q-query POT such that its q queries always refer to at most $\nu \leq q$ vertices, since in this case the transformation in [20, Sec. 4] yields a quasi-canonical POT of vertex complexity ν.

4.2 Graph partition problems

Here we refer to the general framework of graph partition problems presented in [14, Sec. 9]. Each problem in this framework is specified by $k + k + \binom{k}{2}$ pairs of parameters in $[0,1]$, which specify lower and upper bounds on the densities of the parts in a k-partition of the graph as well as on the number of edges within each part and between the parts. For example, k-colorability is specified by trivial lower and upper bounds for all densities, except for upper bounds (of zero) on the number of edges within each part, which mandate that each part is an independent set. Another example is ρ-clique, which can be captured by requiring that the first part has density exactly ρ and the number of edges within it is $\approx \rho^2 \cdot n$. We start with the following generic result, of which the second item in Theorem 1.2 is a special case..

Theorem 4.5 (sample complexity linear in the number of vertices): *Every graph partition property has a sample-based tester of sample complexity $s(n,\epsilon) = O(\sqrt{n}/\epsilon^2)$. Furthermore, for k-colorability the bound is $O(\sqrt{n}/\epsilon)$.*

Unfortunately, the tester that we use for proving Theorem 4.5 runs in exponential-time (i.e., in time that is exponential in its sample complexity). We thus view Theorem 4.5 as a starting point for a search for sample-based testers of sublinear time complexity.

Proof: The tester considers all possible k-partitions of the vertex set of the graph that match the vertex density bounds. (Indeed, there are at most $k^{|V|}$ such partitions.) For each such partition, the tester tries to find evidence for violation of the bounds regarding edge density. By using $O(tk^4/\epsilon^2)$ labeled samples, it can estimate the edge density between two pairs of vertex sets up to an additive error of $\epsilon/2k^2$ with error probability at most $\exp(-t)$. Setting $t = O(|V| \log k)$ and applying a union bound, we may assume that all estimates obtained by the tester are within this deviation. Hence, the tester rejects if and only if evidence of violation was found for each of the foregoing

k-partitions of V. The improved bound for k-Colorability follows by observing that in this case the violation events refer to existence of edges (rather than deviation from a given edge density). ■

We next restate and prove the first item in Theorem 1.4 as well as show that result is tight.

Theorem 4.6 (on the complexity of sample-based testers for Bipartiteness):

1. *Bipartiteness has a* (one-sided error) *sample-based tester of time complexity $O(\epsilon^{-1}\sqrt{n})$. In particular, this tester uses $O(\epsilon^{-1}\sqrt{n})$ samples.*

2. *Any sample-based tester for Bipartiteness has sample complexity $s(n,\epsilon) = \widetilde{\Omega}(\epsilon^{-1}) \cdot \sqrt{n}$, provided $\epsilon > 1/\sqrt{n}$.*

Proof: The tester accepts if and only if the subgraph of the input graph that is revealed by the samples labeled 1 (which indicate the existence of an edge) is bipartite. That is, the tester uses the sample in order to construct a subgraph of the input graph, and accepts if and only if this subgraph is bipartite.

This algorithm always accepts graphs that are bipartite. In analyzing what happens when the input graph, G, is ϵ-far from being bipartite, we follow the idea that underlines the proof of Theorem 4.5, but use it as a mental experiment (rather than an actual algorithm, where the actual algorithm is as stated above). Specifically, considering all possible 2-partitions of the graph $G = (V, E)$ (which is ϵ-far from being bipartite), we note that, with very high probability, a sample of size $O(|V|/\epsilon)$ contains at least one violating edge for each of these 2-partitions (where a violating edge is one that connects two vertices that are on the same side of the 2-partition). Hence, the subgraph of revealed edges (considered by the algorithm) is not bipartite, and Item 1 follows.

Turning to Item 2, we consider the following two distributions over n'-vertex graphs, where $n' = \lceil\sqrt{2n}\rceil$ and each distribution is obtained by a random labeling of the corresponding unlabeled graph: The two graphs are

1. A 6-cycle of equal sized independent sets, denoted $V_0, \ldots, V_5$, such that each pair $(V_i, V_{i+1 \bmod 6})$ is connected by a complete bipartite graph.[20]

2. Two disjoint 3-cycles of independent sets such that the three sets in each 3-cycle are connected by bipartite graphs (as in the first graph).

(A random relabeling is used so as to deem the vertex labels uninformative.) Note that all graphs in the first distribution are bipartite, while all graphs in the second distribution are 1/18-far from any bipartite graph. These two distributions are indistinguishable by samples that reveal no cycles, whereas the probability of revealing a cycle of length i when each edge is selected with probability p is at most $n' \cdot 2^{i-1} \cdot (n'/6)^{i-1} \cdot p^i$. Hence, $\sum_{i \geq 3} n' \cdot 2^{i-1} \cdot (n'/6)^{i-1} \cdot p^i = \omega(1)$ must hold, and $n'p = \Omega(1)$ follows. Noting that p corresponds to $s(n, 0.05)/n$, we infer that $s(n, 0.05)/n = p = \Omega(1/n')$,[21] which implies $s(n, 0.05) = \Omega(\sqrt{n})$. By applying the above construction to a $\sqrt{\epsilon}$ fraction of the vertices, we get $s(n,\epsilon) = \Omega(\epsilon^{-1} \cdot s(\binom{\sqrt{\epsilon}n'}{2}, 0.05)) = \Omega(\sqrt{n/\epsilon})$, which establishes a weaker lower bound than the one in Item 2.

To establish the claimed lower bound, we generalize the construction as follows (for any $\epsilon > 1/\sqrt{n}$).[22] We consider the following two distributions on n'-vertex graphs, where $n' = \sqrt{2n}$, $\ell \in \{\lfloor\log_2(1/\epsilon)\rfloor - 1, \lfloor\log_2(1/\epsilon)\rfloor\}$ is odd, and $m = \epsilon^{-1}/2\ell^2$.

[20] In other words, this is a $n'/6$-factor blow-up of a 6-vertex cycle.

[21] Again, the correspondence is not accurate, but it suffices for our purposes: See Appendix A.2.

[22] In the foregoing paragraph, we used $\ell = 3$ and $m = 1$, which corresponds to $\epsilon = 1/18$.

1. A collection of $2\ell m$ independent sets, arranged in m disjoint cycle of length 2ℓ.
2. A collection of $2\ell m$ independent sets, arranged in $2m$ disjoint cycle of length ℓ.

Note that each of the graphs in the second distribution is ϵ-far from any bipartite graph (since $\epsilon = 1/2\ell^2 m$). On the other hand, these two distributions are indistinguishable by samples that reveal no cycle of length at least ℓ, whereas the probability of revealing a cycle of length i when each edge is revealed with probability p is at most $n' \cdot 2^{i-1} \cdot (n'/2\ell m)^{i-1} \cdot p^i$. Hence, $\sum_{i\geq\ell} n' \cdot 2^{i-1} \cdot (n'/2\ell m)^{i-1} \cdot p^i = \Omega(1)$ must hold, and $n'p/\ell m = \Omega(1)$ follows. Noting that p corresponds to $s(n,\epsilon)/n$, we infer that $s(n,\epsilon)/n = p = \Omega(\ell m/n')$, which implies $s(n,\epsilon) = \Omega(\epsilon^{-1}\sqrt{n}/\log(1/\epsilon))$. ■

Digest. As clarified in the proof of Item 1 of Theorem 4.6, the proof of Theorem 4.5 actually asserts that, for any graph partition property Π, with very high probability, the subgraph of $G = (V, E)$ revealed by $s = O(|V|/\epsilon^2)$ samples of its vertex pairs indicates whether G is in Π or is ϵ-far from Π. Specifically, for every graph partition problem Π_n, consider the corresponding problem Π'_s in which edge densities are taken as a fraction of s (rather than as a fraction of n). Then, with high probability, the distance of G from Π_n is approximated by the distance of the revealed graph from Π'_s. Furthermore, in the case of k-coloring, we may let $\Pi' = \Pi$ (and s can be reduced to $O(|V|/\epsilon)$ as in the case of Bipartiteness). Thus, the computational complexity of the sample-based tester for Π_n is at most the computational complexity of Π'_s.

One natural question is whether we can obtain more efficient (w.r.t computational complexity) sample-based testers for graph partition properties, while maintaining the sample complexity of $s(n,\epsilon) = s'(\epsilon) \cdot \sqrt{n}$, where $s' : (0, 1] \to \mathbb{N}$ is an arbitrary function. Recall that using a sample of size $\widetilde{O}(n)$, which allows to fully reconstruct the graph, we can test Π_n in $(\widetilde{O}(n) + \exp(\text{poly}(1/\epsilon)))$-time (by reconstructing a random induced graph of size $\text{poly}(1/\epsilon)$, and applying brute force on it; cf. [14, Sec. 9]). More generally, we obtain the following trade-off.

Theorem 4.7 (sample vs time trade-off): *For every $t = t(n,\epsilon) \in [\text{poly}(1/\epsilon), \sqrt{n}]$, every graph partition property has a sample-based tester of sample complexity $s(n,\epsilon) = O(n/(t\epsilon^2))$ and time complexity $\exp(t) + s(n,\epsilon)$.*

Theorem 4.5 corresponds to $t = \sqrt{n}$, whereas the prior comment corresponds to $t = \text{poly}(1/\epsilon)$.

Proof: When obtaining $s(n,\epsilon)$ random (vertex pair) samples labeled by the graph $G = (V, E)$, where $n = \binom{|V|}{2}$, the tester selects a random set of t vertices, denoted U, and (w.v.h.p) derives a residual sample of $\Omega(s(n,\epsilon) \cdot (t/|V|)^2)$ pairs in $U \times U$. Hence, the size of the residual sample is $\Omega(t/\epsilon^2)$, and it is labeled by G_U (the subgraph of G induced by U). Using this residual sample, the (sample-based) tester emulates the (sample-based) tester of Theorem 4.5 (for testing G_U with respect to a relaxed version of Π). Hence, our tester has time complexity $\exp(t)$, and its analysis is based on the fact that a random induced subgraph (of that size) maintains the distance of G to Π. (The latter fact originates in [14, Sec. 5.2], see also [14, Cor. 7.2] and [14, Cor. 8.9].) ■

An improved result for k-Colorability. While not resolving the general question raised above, the following result obtains a relatively efficient (w.r.t computational complexity) sample-based testers of sublinear sample complexity for the special case of k-Colorability (2nd item of Theorem 1.4). Specifically:

Theorem 4.8 (on the complexity of sample-based testers for k-Colorability): *For every $k \geq 3$, there exists a* (one-sided error) *sample-based tester of time complexity $f_k(\epsilon) \cdot n^{1-(1/2k)}$ for k-Colorability, where $f_k(\epsilon) \stackrel{\rm def}{=} \exp(\exp(\widetilde{O}(k/\epsilon)))$. Furthermore, this tester uses $s_k(n,\epsilon) = (k/\epsilon^2) \cdot n^{1-(1/2k)}$ samples.*

Proof: We first prove a weaker result and later modify it to obtain the claimed result. The weaker result asserts a sample-based tester of sample complexity $s_k(n,\epsilon)$ that has time complexity $f_k(\epsilon) \cdot n^{\log f_k(\epsilon)}$, and is based upon the following tester:

> On input a random labeled sample of size $s_k(n,\epsilon)$, the tester constructs the subgraph R of the input graph that is revealed by the samples labeled 1 (which indicate the existence of an edge), and accepts if and only if for every set U of $k^{3k/\epsilon}$ vertices the subgraph of R induced by U is k-colorable.
>
> That is, letting $G = (V,E)$ denote the input graph and S denote the set of samples (i.e, $S \in (V \times V)^{s_k(n,\epsilon)}$), the tester first constructs the subgraph $R = (V, E \cap S)$. Then, for every $U \subset V$ of size $k^{3k/\epsilon}$, the tester checks whether the induced subgraph $R_U = (U, (E \cap S) \cap (U \times U))$ is k-colorable. The tester accepts if and only if all checks are positive.

Thus, the foregoing tester always accepts k-colorable graphs, and has running time $\binom{|V|}{k^{3k/\epsilon}} \cdot \exp(k^{3k/\epsilon})$. The rest of the analysis is aimed at showing that *if G is ϵ-far from k-colorable, then, with high probability over the choice of the sample, there exists a set U of $k^{3k/\epsilon}$ vertices such that the subgraph of R induced by U is not k-colorable.*

Our analysis adapts the analysis of the (query-based) k-colorability tester of [2], which in turn builds upon the analysis of [14]. We use [2] (rather than [14]) as our starting point not because of its improved query complexity but rather because its "adaptive" nature lends itself well to our current needs. Specifically, the analysis of [2] offers a flexible analysis of an adequate k-ary tree of "color traces" (defined next). In all our definitions, we refer to a fixed graph $G = (V,E)$ (and to a fixed value of the proximity parameter ϵ).

Definition 4.8.1 (color traces): *A* color trace *is a sequence of $t \geq 0$ pairs over $V \times [k]$; that is, $((v_1,c_1),\ldots,(v_t,c_t))$ is a color trace if for every $i \in [t]$ it holds that $v_i \in V$ and $c_i \in [k]$.*

Note that the empty sequence is a color trace. The k-ary tree eluded to above consists of nodes holding vertices of G (i.e., the v_i's)[23] and edges labeled by colors in $[k]$ (i.e., c_i's) such that color traces are read by taking a path from the root of this tree and reading both the labels of the nodes and of the edges. In other words, we shall maintain a set of color traces that is closed under taking prefixes; that is, if $((v_1,c_1),\ldots,(v_t,c_t))$ is in the set, then $((v_1,c_1),\ldots,(v_{t-1},c_{t-1}))$ is also in the set. Furthermore, the k-ary requirement corresponds to saying that if $((v_1,c_1),\ldots,(v_t,c_t))$ is in the set then for every $c \in [k]$ the color trace $((v_1,c_1),\ldots,(v_{t-1},c_{t-1}),(v_t,c))$ is also in the set. Color traces are also viewed as partial assignments (of colors to the vertices), where the color trace $((v_1,c_1),\ldots,(v_t,c_t))$ corresponds to assigning vertex v_i the color c_i, for every $i \in [t]$. We shall consider the constraints imposed by such color traces on the (rest of the) vertices in G.

Definition 4.8.2 (forbidden colors): *The set of* forbidden colors of vertex v w.r.t the color trace $\overline{\tau} = ((v_1,c_1),\ldots,(v_t,c_t))$, *denoted $F_{\overline{\tau}}(v)$, is the set of colors assigned by $\overline{\tau}$ to vertices that reside on $\overline{\tau}$ and neighbor v; that is, $F_{\overline{\tau}}(v) \stackrel{\rm def}{=} \{c_i : i \in [t] \wedge \{v,v_i\} \in E\}$.*

[23] With the exception of leaving the leaves of the tree empty.

That is, the forbidden set of v is the set of colors that are not allowed for v in any k-coloring that extends the coloring represented in the trace. Indeed, the color trace imposes natural restrictions on itself, and violating these restrictions yield an illegal color trace (as defined next).

Definition 4.8.3 (illegal traces): *A color trace $\overline{\tau} = ((v_1, c_1), \dots, (v_t, c_t))$ is* illegal *if there exists $i \in [t]$ such that c_i is a forbidden color of vertex v_i w.r.t the color trace $\overline{\tau}' = ((v_1, c_1), \dots, (v_{i-1}, c_{i-1}))$; that is, if $c_i \in F_{\overline{\tau}'}(v_i)$. Otherwise, the trace is called* legal*.*

Indeed, illegal traces must have length at least two. Turning back to the forbidden sets of all vertices in G, we consider the effect on these sets that is due to augmenting a specific color trace with a specific vertex-color pair.

Definition 4.8.4 (restricting pairs and vertices): *We say that $(v, c) \in V \times [k]$ is* restricting w.r.t $\overline{\tau}$ *if $|R_{\overline{\tau}}(v, c)| \geq \epsilon|V|/3$, where*

$$R_{\overline{\tau}}(v, c) \stackrel{\rm def}{=} \{w \in V : F_{\overline{\tau},(v,c)}(w) \supset F_{\overline{\tau}}(w)\} \; . \qquad (6)$$

We say that $v \in V$ is restricting w.r.t $\overline{\tau}$ *if for every $c \in [k]$ either $c \in F_{\overline{\tau}}(v)$ or (v, c) is restricting w.r.t $\overline{\tau}$.*

Indeed, for every $\overline{\tau}, (v, c)$ and w, it holds that $F_{\overline{\tau},(v,c)}(w)$ equals either $F_{\overline{\tau}}(w)$ or $F_{\overline{\tau}}(w) \cup \{c\}$, and $F_{\overline{\tau},(v,c)}(w) \supset F_{\overline{\tau}}(w)$ holds if and only if $c \notin F_{\overline{\tau}}(w)$ and $\{v, w\} \in E$. Also, every v such that $F_{\overline{\tau}}(v) = [k]$ is restricting w.r.t $\overline{\tau}$. It turns out that if G is ϵ-far from being k-colorable, then every legal color trace has many restricting vertices. (The claim holds also for illegal traces, provided that they are not too long, but the following claim suffices for our needs.)

Claim 4.8.5 *If there exists a legal color trace $\overline{\tau} = ((v_1, c_1), \dots, (v_t, c_t))$ such that less than $\epsilon|V|/3$ vertices are restricting w.r.t $\overline{\tau}$, then G is ϵ-close to being k-colorable.*

Proof: Consider the following k-partition of V, denoted $\chi : V \to [k]$. For each $i \in [t]$, we set $\chi(v_i) = c_i$. For any *non-restricting* vertex v, we set $\chi(v) = c$ such that (v, c) is *non-restricting* w.r.t $\overline{\tau}$. (Such a color c exists, since otherwise v must be restricting. For all other vertices (i.e., the restricting ones), we set the color χ arbitrarily.) We upper bound the number of violating edges (i.e., edges with both endpoints assigned the same color) by charging edges that are incident at a non-restricting vertex v to the pair $(v, \chi(v))$, and charging edges that connect two restricting vertices to the corresponding vertex pair. Denoting the set of non-restricting (w.r.t $\overline{\tau}$) vertices by V', the total charge is upper-bounded by

$$\sum_{v \in V'} |R_{\overline{\tau}}(v, \chi(v))| + \binom{|V \setminus V'|}{2} \; < \; |V| \cdot \frac{\epsilon|V|}{3} + \binom{\epsilon|V|/3}{2}$$

and the claim follows. ■

Definition 4.8.6 (open traces): *A legal color trace $\overline{\tau}$ is called* open *if there exist at least $\epsilon|V|/3$ vertices that are restricting w.r.t $\overline{\tau}$.*

Claim 4.8.5 implies that if G is ϵ-far from being k-colorable, then each color trace must be either illegal or open. The following "restriction procedure" is aimed at producing a set of traces with a frontier (i.e., maximal traces) that are all illegal traces, where a trace is in the frontier of the set (i.e., is maximal) if it is not a proper prefix of any other trace in the set. The restriction procedure may only augment an existing trace $\overline{\tau} = ((v_1, c_1), \ldots, (v_t, c_t))$ by a vertex-color pair (v, c) if $F_{\overline{\tau}}(v)$ $(= \{c_i : i \in [t] \wedge \{v, v_i\} \in E\})$ equals $\{c_i : i \in [t] \wedge \{v, v_i\} \in E \cap S\}$, where S is the sequence of samples provided to the tester.[24] This means that the sample S contains "evidence" for each color contained in the forbidden set $F_{\overline{\tau}}(v)$ of the vertex v, where the said evidence for a color $c' \in F_{\overline{\tau}}(v)$ is a pair $\{v, v_i\} \in E \cap S$ such that $c_i = c'$ (i.e., the pair $\{v, v_i\}$ is in the sample and is an edge of the graph and so v cannot be legally colored by c_i). We stress that the following restriction procedure is merely a mental experiment conducted in the analysis.

Restriction Procedure (as a mental experiment). The procedure is given a sequence of labeled samples, where each sample point is a pair of vertices and the label indicates whether or not the edge is in the graph; that is, the sample point (u, v) is labeled by $g(u, v)$ such that $g(u, v) = 1$ if $\{u, v\} \in E$ and $g(u, v) = 0$ otherwise. (If (u, v) is not in the sample set S, then g is undefined on (u, v).) The procedure is initiated with an empty trace, which is typically an open trace (since otherwise $|E| < \epsilon \cdot \binom{|V|}{2}$). As long as the set of traces contains a *maximal* trace $\overline{\tau} = ((v_1, c_1), \ldots, (v_t, c_t))$ that is both *legal* and *open*[25], the procedure takes the following steps.

1. If there exists no vertex v that satisfies the following two conditions, then the procedure halts with an error message. The conditions (regarding v and $\overline{\tau}$) are:

   (a) v is restricting w.r.t $\overline{\tau}$; and

   (b) for every $c \in F_{\overline{\tau}}(v)$ there exists $i \in [t]$ such that $c_i = c$ and $g(v, v_i) = 1$.
   (Recall that $g(v, v_i) = 1$ means that (v, v_i) is in the sample set S and $\{v, v_i\} \in E$.)

   Otherwise (i.e., some v satisfies the above condition), the procedure picks such a vertex v.

2. For every $c \in [k]$, the procedure augments the set of traces with $((v_1, c_1), \ldots, (v_t, c_t), (v, c))$.
   (Indeed, this trace is illegal if $c \in F_{\overline{\tau}}(v)$, and this illegality is visible in the sample.)

If the current set of traces contains no maximal trace that is both legal and open, then the procedure halts and outputs this set of traces.

Claim 4.8.7 *The set of color traces that the procedure constructs contains traces of length at most $3k/\epsilon$.*

Proof: The procedure only augments traces by using restricting vertices, but any restricting vertex w.r.t a given trace increases the sum of the sizes of the forbidden sets by at least $\epsilon|V|/3$. Specifically,

[24] This is the point where we depart from [2]: The algorithm in [2] uses a single sample of $\widetilde{O}(1/\epsilon^2)$ vertices and queries all vertex pairs in this sample. Only these vertices will be placed in color traces, and the algorithm has evidence for each color contained in the forbidden set of a vertex v used to augment $\overline{\tau}$. In contrast, our algorithm obtains a random sample S of vertex pairs, and the restriction procedure presented below is confined to evidence that is revealed by S. In particular, the restriction procedure of [2] (re)uses the same vertices on all traces, whereas our restriction procedure uses different vertices when augmenting different traces.

[25] Recall that this condition (as well as the other conditions below) is not tested by the actual algorithm (i.e., the one outlined in the beginning of this proof). This condition is only considered by the mental experiment that we conduct in our analysis. Furthermore, we may assume that all legal traces are open, since Claim 4.8.5 asserts that this is the case if G is ϵ-far from being k-colorable.

suppose that the trace $\overline{\tau} = ((v_1, c_1), \ldots, (v_t, c_t))$ is contained in the output, and let $\overline{\tau}^{(i)}$ denote its i-long prefix (i.e., the trace $((v_1, c_1), \ldots, (v_i, c_i))$). Then, for every $i \in [t]$, it holds that

$$\begin{aligned} \sum_{v \in V} |F_{\overline{\tau}^{(i)}}(v)| &= |R_{\overline{\tau}^{(i-1)}}(v_i, c_i)| + \sum_{v \in V} |F_{\overline{\tau}^{(i-1)}}(v)| \\ &\geq \frac{\epsilon |V|}{3} + \sum_{v \in V} |F_{\overline{\tau}^{(i-1)}}(v)| \end{aligned}$$

and it follows that $t \leq k|V|/(\epsilon|V|/3)$. ■

Claim 4.8.8 *Suppose that G is ϵ-far from being k-colorable. If the restriction procedure halts without error, then the sample S contains a subgraph that is not k-colorable. Furthermore, this subgraph contains only vertices that appear in the set of color traces output by the procedure.*

Proof: By Claim 4.8.5 each color trace must be either illegal or open (since G is ϵ-far from being k-colorable). Hence, the (non-error) output of the procedure is a set of color traces such that all maximal traces in this set are illegal. Furthermore, for each vertex that occurs on any trace, the sample contains evidence for the forbidden colors w.r.t the relevant prefix of the trace. Thus, an illegal trace asserts that the corresponding coloring of its vertices is not a legal k-coloring in the subgraph R revealed by the sample. Now, since all maximal traces are illegal and the set of traces corresponds to a k-ary tree (i.e., for each non-maximal trace $\overline{\tau}$, there exists a v such that, for each $c \in [k]$, the set of traces contains the trace $(\overline{\tau}, (v, c)))$, it follows that there is no legal k-coloring of R (or rather of the subgraph of R induced by the vertices that appear on this set of traces). ■

Claim 4.8.9 *The probability that the restriction procedure halts with error is at most $1/3$, where the probability is taken uniformly over the choice of the sample S.*

Proof: It will be more convenient to consider a sample S that is chosen by selecting each vertex pair with probability $p = s_k(n, \epsilon)/2n$, independent of all other choices. (With very high probability, such a sample of independently chosen pairs contains at most $2pn = s_k(n, \epsilon)$ pairs, whereas the probability that the restriction procedure halts with error decreases when the sample size increases.)[26] The restriction procedure makes at most $k^{3k/\epsilon}$ iterations, where in each iteration it tries to extend a legal open trace $\overline{\tau} = ((v_1, c_1), \ldots, (v_t, c_t))$. At this point, the set of restricting vertices, denoted V', has size at least $\epsilon|V|/3$. For each vertex $v \in V'$, let $I_v \subseteq \{i \in [t] : \{v, v_i\} \in E\}$ be a set of $|F_{\overline{\tau}}(v)|$ indices such that for every $c \in F_{\overline{\tau}}(v)$ there exists $i \in I_v$ such that $c_i = c$ (and $\{v, v_i\} \in E$). Now, the probability that the sample contains $\{(v, v_i) : i \in I_v\}$ equals $p^{|I_v|}$, and these events are independent for the various v's in V' (since each event refers to the existence of samples that are adjacent at a different vertex $v \in V'$).[27] Thus, the current iteration fails with probability at most

$$\prod_{v \in V'} (1 - p^{|F_{\overline{\tau}}(v)|}) < (1 - p^k)^{\epsilon|V|/3}$$

which is $\exp(-p^k \cdot \epsilon|V|)$. Using $p = \Omega(\epsilon^{-1} \cdot |V|^{-1/k})$, we have $\exp(-p^k \cdot \epsilon|V|) \ll k^{-3k/\epsilon}$ and the claim follows. ■

Combining Claims 4.8.9 and 4.8.8, it follows that *if G is ϵ-far from k-colorable, then, with probability at least $2/3$ over the choice of the sample S, the vertices that appear in the output of the restriction*

[26] See further justification in Appendix A.2.

[27] Also note that these events are unconditioned by any other events that were considered when constructing $\overline{\tau}$, since these events refer to samples that connect pairs of v_i's.

procedure induce a subgraph of R that is not k-colorable, where R is the subgraph of G that is visible in the sample. Using Claim 4.8.7, the number of vertices in this output (and hence in the induced subgraph) is at most $f'_k(\epsilon) = k^{3k/\epsilon}$. This establishes the claim that k-colorability has a sample-based tester of sample complexity $s_k(n, \epsilon)$ and time complexity $n^{f'_k(\epsilon)} \cdot \exp(f'_k(\epsilon))$.

In order to improve the time complexity of the tester, we turn the restriction procedure from a mental experiment to an actual procedure to be implemented by the tester. An immediate difficulty that arises is that the tester may not be able to determine the forbidden sets of various vertices and the (approximate) number of restricting vertices (w.r.t various color traces). Our solution is just to guess these unknowns at random (and to compensate for the lower probability of success by a suitable number of repetitions (to be discussed below)).

Restriction Procedure (as an actual implementation). The procedure is initiated with an empty trace, and assumes that each trace is legal unless some evidence is found against this assumption (see below). (For every legal trace, the procedure also assumes that the trace is open, and so we shall not refer at all to the question of whether a trace is open.) (Note that the procedure does not know the forbidding set of a vertex, nor its neighbors on the trace, and so it cannot tell whether or not a trace is actually legal, let alone whether or not a trace is open.) In particular, as long as the set of traces contains a *maximal* trace $\overline{\tau} = ((v_1, c_1), \ldots, (v_t, c_t))$ that is *assumed* to be legal, the procedure takes the following steps.

1. If $t \geq 3k/\epsilon$, then the procedure halts with an error message.

   Otherwise (i.e., $t < 3k/\epsilon$), if $t \leq k$, then the procedure sets $I = [t]$, otherwise the procedure randomly selects I uniformly among all k-subsets of $[t]$. The procedure selects at random a vertex v among all vertices v such that for every $i \in I$ the pair (v, v_i) is in the sample.

   (If there exists no vertex v that satisfies the above condition, then the procedure halts with an error message. This is quite unlikely (since $(s_k(n, \epsilon)/n)^k \cdot |V| \gg 1$), and otherwise we continue to the next step.)

   (Motivation: Our hope is that the vertex v selected above is restricting w.r.t $\overline{\tau}$, and that $\{c_i : i \in I \wedge \{v, v_i\} \in E\} = F_{\overline{\tau}}(v)$.)

2. Let v be the vertex selected in the previous step, and let $C_v \subseteq F_{\overline{\tau}}(v)$ be the set of colors that cannot be assigned to v due to an edge $\{v, v_i\}$ that is visible in the sample; that is, $C_v = \{c_i : i \in I \wedge \{v, v_i\} \in E\}$, where I is the set selected in the previous step. For every $c \in [k]$, the procedure augments the set of traces with the trace $((v_1, c_1), \ldots, (v_t, c_t), (v, c))$. For each such new trace, the procedure assumes that it is legal unless $c \in C_v$; that is, $((v_1, c_1), \ldots, (v_t, c_t), (v, c))$ is assumed to be legal if and only if $c \notin C_v$.

If the current set of traces contains no maximal trace that is assumed to be legal, then the procedure halts outputting this set of traces.

Note that we augmented the original Step 1 such that the procedure never outputs a set of traces that contains a trace that is longer than $3k/\epsilon$. Thus, the restriction procedure can be implemented in time $O(k^{3k/\epsilon}) \cdot s_k(n, \epsilon)$.

Turning to the analysis of the output of the procedure, note that whenever a set of traces is output, the maximal traces in the output are all illegal (and their illegality is visible in the sample). Thus, we obtain a $k^{3k/\epsilon}$-vertex subgraph of G that is not k-colorable such all edges of this subgraph appear in the sample. This will happen if (and only if) all random guesses made by the restriction procedure are successful (i.e., satisfy our hopes as stated in the foregoing motivation). Indeed, the probability of this random event is quite low, but our final tester will consist of invoking the restriction procedure for an adequate number of times. We refer to such a single invocation as a basic tester. We stress that the basic tester accepts whenever the procedure halts with an error message, and rejects only if the procedure outputs a set of traces (which by the foregoing discussion yields a subgraph of R that is induced by the vertices that reside on these traces such that this subgraph is not k-colorable).

Assuming that G is ϵ-far from being k-colorable, we lower bound the probability that the basic tester rejects. This boils down to lower-bounding the probability that the actual restriction procedure correctly emulates the mental experiment. The key to the analysis is provided by the following claim.

Claim 4.8.10 *Suppose that $\overline{\tau} = ((v_1, c_1), \ldots, (v_t, c_t))$ is a legal and open trace produced by the restriction procedure, and that the procedure executes Step 1 on $\overline{\tau}$. Then, with probability at least $(\epsilon/3k)^{2k+1}$, the procedure selects a vertex that is restricting w.r.t $\overline{\tau}$, where the probability is taken uniformly over the choice of the sample S and the choices made in Step 1.*

Proof: Again, it will be more convenient to consider a sample S that is chosen by selecting each vertex pair with probability $p = s_k(n, \epsilon)/2n$, independent of all other choices.[28] For every $I \subseteq [t]$, let S_I be the set of vertices v such that for every $i \in I$ the pair (v, v_i) is in the sample S. Note that each S_I is a random set of vertices that is obtained by selecting each vertex with probability exactly $p^{|I|}$. Turning to the vertices that are restricting w.r.t $\overline{\tau}$, we denote by V_J the set of restricting vertices v such that $\{c_j : j \in J\} = F_{\overline{\tau}}(v)$. Then, there exists a set $J \subseteq [t]$ of size at most k such that $\frac{|V_J|}{|V|} > \frac{\epsilon/3}{(3k/\epsilon)^k} > 2 \cdot (\epsilon/3k)^{k+1}$, since the union of all these V_J's contains all restricting vertices. Letting $I \subseteq [t]$ be an arbitrary $\min(k, t)$-superset of J, and suppose that I is selected at Step 1 (which happens with probability at least $(\epsilon/3k)^k$). Then, with very high probability over the choice of S, the set $S_I \cap V_J$ constitutes at least a $(\epsilon/3k)^{k+1}$ fraction of S_I, where we use $(\epsilon/3k)^{k+1} \cdot p^k |V| \gg 1$ (cf. Claim 4.8.9). Hence, for J fixed as above, a set $I \supseteq J$ is selected (in Step 1) with probability at least $(\epsilon/3k)^{k+1}$, the random $v \in R_I$ (also selected in Step 1) is in V_J with probability at least $(\epsilon/3k)^k$, and the claim follows. ■

Indeed, as suggested in the foregoing motivation, it follows that the actual restriction procedure emulates the imaginary one with probability at least $((\epsilon/3k)^{2k+1})^{k^{3k/\epsilon}} = 1/f_k(\epsilon)$, which establishes the time bound claimed in the theorem (since it suffices to repeat the basic test for $f_k(\epsilon)$ times). But this argument means that the sample complexity is also multiplied by a factor of $f_k(\epsilon)$.

To obtain a better upper bound on the sample complexity, we modify the restriction procedure such that the choice of I is made by branching in parallel (to $(3k/\epsilon)^k$ branches) rather than by a random selection (among $(3k/\epsilon)^k$ options). Note that the proof of Claim 4.8.10 actually shows that the probability that a restricting vertex is selected (on the adequate branch) is at least $(\epsilon/3k)^{k+1}$.

Next, rather than selecting at random one vertex (in Step 1 of the restriction procedure), we select at random $(3k/\epsilon)^{2k}$ vertices (among all vertices v such that for every $i \in I$ the pair (v, v_i) is in the sample), and branch on these $(3k/\epsilon)^{2k}$ chosen vertices too, where all is done using the same sample. We can use the same sample on all branches, because we actually care only about what happens on one good branch. Hence, we get back to the sample complexity of the simpler version (i.e., the one that uses an imaginary procedure). For sake of clarity, we spell out the revised restriction procedure.

Restriction Procedure (revised parallel implementation). For $K \stackrel{\rm def}{=} (3k/\epsilon)^k$ and $M \stackrel{\rm def}{=} k^{3k/\epsilon}$, we run in parallel $(K \cdot K^2)^M$ parallel copies of the restriction procedure described above, all using the same

[28] Again, the correspondence is not accurate, but it suffices for our purposes: See Appendix A.2.

sample. The copies correspond to choices $((I_1, i_1), \ldots, (I_M, i_M))$, where each I_j is a k-subset of $[3k/\epsilon]$ and $i_j \in [K^2]$. A generic copy, which corresponds to such a sequence of pairs, uses the sets $I_1, \ldots, I_M$ instead of the random choices of k-subsets made in Step 1 such that when augmenting the j^{th} trace it uses the k-subset I_j. These copies also share their randomness, denoted $(r_1, \ldots, r_M)$, which is used to select the vertices v in Step 1 such that r_j is used when augmenting the j^{th} trace. The generic copy that corresponds to $((I_1, i_1), \ldots, (I_M, i_M))$ uses the i_j^{th} block of r_j when augmenting the j^{th} trace (i.e., $r_j = (r_{j,1}, \ldots, r_{j,K^2})$ and the i_j^{th} block of r_j means r_{j,i_j}). Thus, the modified Step 1 of this generic copy reads:

> Suppose that this is the j^{th} time that Step 1 is executed. Then (assuming $t < 3k/\epsilon$ and $t > k$)[29], sets $I = I_j$ and uses the randomness r_{j,i_j} in order to selects v (among all vertices v such that for every $i \in I$ the pair (v, v_i) is in the sample).

Each of the copies produces an output as the original restriction procedure, and the parallel restriction procedure outputs a set of traces (rather than an error message) if any of the copies has output such a set. ■

Beyond k-Colorability. The case of k-Colorability represents almost all that can be tested via one-sided error testers of query complexity that only depends on ϵ. Recall that by [20, Thm. 3], the only such (non-trivial)[30] properties are (1) the property of being a clique, and (2) properties that imply k-Colorability. The first property (which consists of a single graph for every size) can be tested by $O(1/\epsilon)$ random samples, and the result of Theorem 4.8 extends to the second class (as detailed next). Properties of the second class correspond to k-vertex graphs such that the property associated with the graph $H = ([k], A)$ consists of all graphs $G = (V, E)$ that can be k-partitioned such that edges appear only between vertices that reside in the i^{th} and j^{th} part when $(i, j) \in A$. Such graphs G are called H-embeddable.

Definition 4.9 (H-embeddable graphs): *Let $H = ([k], A)$ be a simple graph* (with no self-loops). *The graph $G = (V, E)$ is H*-embeddable *if there exists a k-partition of V into $(V_1, \ldots, V_k)$ such that $E \cap (V_i \times V_j) \neq \emptyset$ only if $(i, j) \in A$.*

Note that k-Colorability corresponds to the set of graphs that are embedded in the k-vertex clique.

Theorem 4.10 (on the complexity of sample-based testers for H-Embeddability): *Let $H = ([k], A)$ be a simple graph* (with no self-loops). *Then there exists a* (one-sided error) *sample-based tester of time complexity $f_k(\epsilon) \cdot n^{1-(1/2k)}$ for H-Embeddability, where $f_k(\epsilon) \stackrel{\text{def}}{=} \exp(\exp(\widetilde{O}(k/\epsilon)))$. Furthermore, this tester uses $s_k(n, \epsilon) = (k/\epsilon^2) \cdot n^{1-(1/2k)}$ samples.*

Proof Sketch: Building upon the proof of Theorem 4.8, we let $F'_{\overline{\tau}}(v) \stackrel{\text{def}}{=} \cup_{c \in F_{\overline{\tau}}(v)} \{c' \in [k] : h(c, c') = 0\}$, where $h(c, c') = 0$ if and only if $\{c, c'\} \notin A$. That is, while $F_{\overline{\tau}}(v)$ denotes the set of colors of the vertices on the color trace $\overline{\tau}$, here the set $F'_{\overline{\tau}}(v) \supseteq F_{\overline{\tau}}(v)$ is the set of colors that are forbidden for v (in a legal extension of the coloring implicit in $\overline{\tau}$). All definitions and claims (with one exception discuss below) will refer to the modified sets of forbidden colors $F'_{\overline{\tau}}(v)$. The only exception is the "evidence condition" in Item 1b of the Restriction Procedure, where we still require that *for every $c \in F_{\overline{\tau}}(v)$* (rather than $c \in F'_{\overline{\tau}}(v)$) *there exists $i \in [t]$ such that $c_i = c$ and $g(v, v_i) = 1$* (where $g(v, v_i) = 1$ means that (v, v_i) is in the sample and $\{v, v_i\} \in E$). ■

[29] The other cases are handled exactly as before; that is, the copy halts with error if $t \geq 3k/\epsilon$ and sets $I = [t]$ if $t \leq k$.

[30] A property is called non-trivial (for testing) if, for every $\epsilon > 0$ and all sufficiently large n, there exists both an n-sized object in the property and an n-sized object that is ϵ-far from the property.

5 Sample-based testers for linearity

Recall that "testing linearity" actually refers to testing homomorphism between two groups. For two groups $(G,+)$ and $(H,\odot)$, the function $h:G\rightarrow H$ is a group homomorphism if for every $x,y\in G$ it holds that $h(x+y)=h(x)\odot h(y)$. Recall that group homomorphism can be tested by $O(1/\epsilon)$ queries regardless of the size of the groups [5], and that by Theorem 3.5 there exists a (one-sided error) sample-based tester of sample complexity $s(n,\epsilon)=\text{poly}(1/\epsilon)\cdot n^{1/3}$. As noted in Section 3.3, a more efficient sample-based tester can be obtained directly.

Theorem 5.1 (the sample-complexity of testing group homomorphism):

1. *For any two groups $(G,+)$ and $H(,\odot)$, group homomorphism from G to H has a one-sided error sample-based tester of sample complexity $O(\epsilon^{-1}+\log|G|)$.*
2. *There exist groups $(G,+)$ and $H(,\odot)$, such that sample-based testing group homomorphism from G to H requires $\Omega(\epsilon^{-1}+\log|G|)$ samples, provided $\epsilon\geq 1/|G|$.*

Proof: The proof of the upper bound (i.e., Item 1) is based on the fact that, with very high probability, the partial sums of $O(\log|G|)$ random elements of G generates all elements of G. In fact, a stronger statement holds:

Claim 5.1.1 *Suppose that $r_1,\ldots,r_t$ are uniformly and independently distributed in G. Then, with probability at least $1-(\delta^{-2}|G|/(2^t-1))$, for every $v\in G$, it holds that $|\{I\subseteq[t]:I\neq\emptyset\bigwedge\sum_{i\in I}r_i=v\}|=(1\pm\delta)\cdot(2^t-1)/|G|$, where $\sum_{i\in I}r_i$ is a shorthand for $r_{i_1}+\cdots+r_{i_{|I|}}$ such that $i_1<i_2<\cdots<i_{|I|}$ is an ordering of the elements of I.*

(We are careful about the meaning of $\sum_{i\in I}$, since G may not be Abelian.)

Proof: For every non-empty $I\subseteq[t]$, we consider the random variable $R_I=R_I(r_1,\ldots,r_t)$ defines as $\sum_{i\in I}r_i$. Each of these random variables is uniformly distributed in G. Furthermore, these random variables are pairwise independent, since for every $I\setminus J\neq\emptyset$ and $u,v\in G$ we have $\Pr[\sum_{i\in I}r_i=u|\sum_{j\in J}r_j=v]=1/|G|$ (by fixing the values of all r_i except for some $i\in I\setminus J$). Now, fix an arbitrary element $v\in G$, and let $\zeta_I=1$ if $R_I=v$ and $\zeta_I=0$ otherwise. Then, $\text{E}[\zeta_I]=1/|G|$ and the ζ_I's are pairwise independent. Letting $\overline{\zeta}_I=\zeta_I-\text{E}[\zeta_I]$, and applying Chebyshev's Inequality we get

$$\begin{aligned}\Pr\left[\left|\sum_{\emptyset\neq I\subseteq[t]}\overline{\zeta}_I\right|>\delta\cdot(2^t-1)/|G|\right] &< \frac{\text{E}\left[\left(\sum_I\overline{\zeta}_I\right)^2\right]}{(\delta\cdot(2^t-1)/|G|)^2}\\ &< \frac{\text{E}\left[\sum_I{\zeta_I}^2\right]}{(\delta\cdot(2^t-1)/|G|)^2}\\ &= \frac{(2^t-1)/|G|}{\delta^2\cdot(2^t-1)^2/|G|^2}\end{aligned}$$

and the claim follows. ■

By Claim 5.1.1, for $t=O(\log|G|)$ and for any homomorphism $h:G\rightarrow H$, the values of h at t random samples, determine the value of h on all of G. This suggests the following sample-based tester. On input an f-labeled sample $((r_1,f(r_1)),\ldots,(r_t,f(r_t)),(r_{t+1},f(r_{t+1})),\ldots,(r_{t+t'},f(r_{t+t'}))$, where $t'=O(1/\epsilon)$, the tester proceeds as follows:

1. Reconstruct a function $\widetilde{f}$ based on the labeled sample and check that $\widetilde{f}$ is a homomorphism. Specifically, for every $x \in G$ find a (non-empty) set $I \subseteq [t]$ such that $\sum_{i \in I} r_i = x$, and set $\widetilde{f}(x) = \odot_{i \in I} f(r_i)$. If $\widetilde{f}$ is not defined on all points in G, then accept. If $\widetilde{f}$ is not a group homomorphism, then reject.

   We continue to the next step only if $\widetilde{f}$ is defined over all of G and is a group homomorphism. (Note that we do not check whether the value defined based on I matches the value that could have been defined based on a different I' such that $\sum_{i \in I} r_i = x$; such a check is natural but not needed.)

2. Test that $f = \widetilde{f}$. That is, accept if and only if for every $i \in [t']$ it holds that $f(r_{t+i}) = \widetilde{f}(r_{t+i})$.

If f is a group homomorphism and $\widetilde{f}$ was defined on $x \in G$, then necessarily $\widetilde{f}(x) = \odot_{i \in I} f(r_i) = f(x)$ for all I's such that $\sum_{i \in I} r_i = x$. Hence, the test always accepts group isomorphisms. Assume, on the other hand, that f is ϵ-far from being a group isomorphism. Then, with probability at least 5/6, for every $x \in G$, there exists an $I \subseteq [t]$ such that $\sum_{i \in I} r_i = x$. In this case, unless the tester rejects already in Step 1, it holds that $\widetilde{f}$ is defined over all of G and is a group homomorphism. Hence, f and $\widetilde{f}$ differ on at least an ϵ fraction of G, and it follows that Step 2 rejects with probability at least 5/6. Having established the upper bound, we now turn to the lower bound.

Claim 5.1.2 *Any sample-based tester for group homomorphism from $\mathbb{Z}_2^m$ to $\mathbb{Z}_2$ requires $\Omega(\epsilon^{-1} + m)$ samples.*

In this case, the group homomorphisms correspond to linear functions.

Proof: The key observation is that, with very high probability, a sample of $m/2$ random points in $\mathbb{Z}_2^m$ yields $m/2$ linearly independent vectors. In this case, the labels provided to these points by a random linear function are identically distributed to the labels provided by a totally random function (which, with very high probability, is 1/5-far from any linear function). Hence, a lower bound of $m/2$ follows. To obtain an $\Omega(1/\epsilon)$ lower bound consider a linear function that is modified on an 2ϵ fraction of its domain, and note that $\Omega(1/\epsilon)$ samples are required for hitting this modified region. ■

This completes the proof of the theorem. ■

Digest. The sample-based tester presented in the proof of Theorem 5.1 is based on an exact learning algorithm: Indeed, when given $O(\log |G|)$ random samples labeled by a group homomorphism f, with very high probability, Step 1 reconstructs f. Furthermore, when Step 1 fails, it indicates this event (by halting with a partially defined $\widetilde{f}$). Thus, an alternative presentation of the (one-sided error) tester may proceed along the reduction of testing to (proper) learning presented in [14, Prop. 3.1], while relying on the features of the aforementioned learning algorithm (i.e., its zero-error features). Indeed, it follows that *testing linearity is not easier than* (exact) *learning of linear function.*

6 Invariance under the symmetric group

In this section we consider the case of properties that are invariant under the symmetric group acting on the functions' domain. That is, we consider properties $\Pi = \cup_{n \in \mathbb{N}} \Pi_n$ such that for every $n \in \mathbb{N}$ and every permutation $\pi : D_n \to D_n$ it holds that $f \in \Pi_n$ if and only if $f \circ \pi \in \Pi_n$, where $(f \circ \pi)(x) = f(\pi(x))$. We call such properties symmetric.

6.1 Queries versus samples

We first note that for testing symmetric properties queries add little power beyond samples. This is quite intuitive when considering sampling without repetitions, but our sampling model is with repetitions and the gap between these two models is significant for $\Omega(\sqrt{n})$ samples. The proof of the following result starts by validating the former intuition and closing the latter gap.

Theorem 6.1 (samples are as good as queries for testing symmetric properties): *Suppose that* Π *is a symmetric property that has a standard tester of query complexity* q. *Then,* Π *has a sample-based tester of query complexity* s' *such that* $s'(n,\epsilon) = O(q(n,\epsilon))$ *if* $q(n,\epsilon) < n/2$ *and* $s'(n,\epsilon) = O(n \log n)$ *otherwise. Furthermore, if the standard tester has one-sided error, then so does the sample-based tester.*

Proof: The basic idea is to emulate the execution of the standard tester on a random isomorphic copy of the function that we are required to test. That is, given the standard tester T, and wishing to test the function $f : D_n \to R_n$, we select uniformly a permutation $\pi : D_n \to D_n$ and emulate an execution of T while providing it with oracle access to the function $f \circ \pi$. That is, if T makes the query i, then we make the query $\pi(i)$. (We assume, without loss of generality, that T never makes the same query twice.) Thus, we perfectly emulate an execution of $T^{f\circ\pi}$, while making queries that are almost uniformly and independently distributed[31], and our decision regarding f is as good as T's decision regarding $f\circ\pi$. Noting that the distance of f to Π_n equals the distance of $f\circ\pi$ to Π_n, we are almost done. The only problem is that the sequence of q queries is uniformly distributed among all q-long sequences of *distinct* elements in D_n, whereas we wish it to be uniformly distributed in D_n^q.

We start by addressing this problem in the case of $q \geq n/2$. In this case, we merely use $O(n \log n)$ independently and uniformly distributed queries in order to obtain the value of f on the entire domain. We may fail with some small constant probability, and in such a case we accept without taking any other action. Otherwise, we just decide based on our full knowledge of f.

Turning to the case of $q < n/2$, we first try to obtain the value of f on a uniformly distributed sequence of q distinct elements in D_n by making $3q$ independently and uniformly distributed queries in D_n. With very high probability, we shall succeed, since for $q < n/2$ it holds that a uniformly distributed sequence $\overline{s} = (s_1, \ldots, s_{3q}) \in D_n^{3q}$ contains at least q distinct elements (i.e., $|\{s_i : i \in [3q]\}| \geq q$). In case we fail, we accept without taking any other action. Otherwise, we emulate the execution of T using the first q distinct elements that we obtained. ■

6.2 The role of the range size

We first note that symmetric properties can be presented as properties of the frequencies of occurrences of the various values in the range. Specifically, let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$ be a symmetric property such that Π_n contains only functions from D_n to R_n, and suppose that $m = |R_n|$. Actually, assume, without loss of generality, that $D_n = [n]$ and $R_n = [m]$. Then, there exists a set $S_n \subset [n]^m$ such that $f \in \Pi_n$ if and only if $(\mathtt{cnt}_1(f), \ldots, \mathtt{cnt}_m(f)) \in S_n$, where $\mathtt{cnt}_i(f) \stackrel{\rm def}{=} |\{j \in [n] : f(j) = i\}|$. It follows that testing Π_n amounts to testing whether the corresponding frequencies sequence is in the corresponding S_n, which means that we can expect the complexity to depend only on m and ϵ (and not on n, at least for large enough n's).

The foregoing observation suggests to consider the complexity of testing symmetric properties also as a function of the size of the range of the function (in addition to its dependence of the size

[31] Indeed, here we use the equivalent formulation of sample-based testers as provided in Proposition 2.4

of the domain). This requires presenting properties as sequences indexed by two (independent) size parameters, n and m, representing the sizes of the domain and range, respectively. That is, we consider classes of the form $\Pi = \cup_{n,m\in\mathbb{N}}\Pi_{n,m}$ such that $\Pi_{n,m}$ contains functions with domain D_n and range R_m, where $|D_n| = n$ and $|R_m| = m$. The tester will be given both n and m (as well as the proximity parameter ϵ), and its sample complexity will be considered as a function of n, m and ϵ. Note that complexity measures that depend on parameters other than the size of the domain are not uncommon in studies of properties of functions (see, e.g., the bounded-degree graph model of [17] and testing monotonicity of functions over arbitrary (product) domains and arbitrary ranges [13, 7]).

Theorem 6.2 (the complexity of testing symmetric functions is determined by the size of the range): *Let $\Pi = \cup_{n,m\in\mathbb{N}}\Pi_{n,m}$ be a property of symmetric functions as above. Then the sample complexity of testing Π, denoted $s : \mathbb{N}^2 \times (0, 1] \to \mathbb{N}$, satisfies $s(n, m, \epsilon) = O(m/\epsilon^2)$.*

Proof: Generalizing the foregoing argument, we note that there exists a set $S_{n,m} \subset [n]^m$ such that $f \in \Pi_{n,m}$ if and only if $(\mathtt{cnt}_1(f), \ldots, \mathtt{cnt}_m(f)) \in S_{n,m}$, where $\mathtt{cnt}_i(f) \stackrel{\rm def}{=} |\{j \in D_n : f(j)=i\}|$. Hence, testing f with respect to $\Pi_{n,m}$ (and proximity ϵ) amounts to approximating the sequence $(\mathtt{cnt}_1(f), \ldots, \mathtt{cnt}_m(f))$ (up to relative proximity of ϵ). Such an approximation can be obtained using a sample of size $O(m/\epsilon^2)$; cf. [6, Lem. 3]. Below, for the sake of illustration, we prove a somewhat weaker bound.

The problem at hand is actually one of approximately learning a probability distribution over a set of m elements when given samples drawn independently from this distribution (i.e., testing f amounts to approximating the distribution in which i appears with probability $\mathtt{cnt}_i(f)/n$). We represent a probability distribution over $[m]$ as a vector $(p_1, \ldots, p_m)$ such that p_i is the probability of the $i^{\rm th}$ element. We wish to find a vector $(\widetilde{p}_1, \ldots, \widetilde{p}_m)$, such that with probability at least 2/3, it holds that $\sum_{i\in[m]} |\widetilde{p}_i - p_i| \le 2\epsilon$.

The obvious solution is to approximate each p_i up to an additive deviation of ϵ/m, but this will require a sample size of $\Omega((m/\epsilon)^2)$. Instead, letting $I = \{i \in [m] : p_i \ge 1/m\}$, it suffices to have $\widetilde{p}_i = (1 \pm \epsilon)p_i$ for every $i \in I$ and $\widetilde{p}_i = p_i \pm \epsilon/m$ for the other i's (i.e., $i \in [m] \setminus I$). Using a sample of size $s = O(\epsilon^{-2} m \log m)$, we obtain $\Pr[\widetilde{p}_i \ne (1 \pm \epsilon)p_i] < 1/3m$ for each $i \in I$, and $\Pr[\widetilde{p}_i \ne p_i \pm \epsilon/m] < 1/3m$ for each $i \in [m] \setminus I$. (Both bounds are obtained by using a multiplicative Chernoff bound for Bernoulli trials with success probability p_i; for $i \in I$ we consider a multiplicative deviation of ϵ and use $p_i \ge 1/m$ (which implies $\exp(-p_i\epsilon^2 s/3) < 1/3m$), whereas for $i \in [m] \setminus I$ we consider a multiplicative deviation of $\delta = \epsilon/mp_i$ and use $p_i \cdot \delta^2 = \epsilon^2/m^2 p_i > \epsilon^2/m = \Omega(s^{-1} \log m)$ (while assuming $p_i \ge \epsilon/m)^{32}$.) ■

6.3 Relation to testing distributions

The proof of Theorem 6.2 (implicitly) reduces the task of testing a symmetric property of functions to the task of testing distributions (to be reviewed next), which is in turn reduced to the task of (approximately) learning distributions. Our objective here is to go in the opposite direction; that is, to cast the model of testing distributions as a special case of testing symmetric properties of functions.

We stress that the model of testing (properties of) distributions has little in common with the model of testing (properties of) functions (especially, in its standard incarnation that refers to a fixed (i.e., uniform) distribution on the function's domain). Nevertheless, we show that the study of

[32] In case $p_i < \epsilon/m$, we can just use the bound obtained for $p_i = \epsilon/m$.

testing distributions can be reduced to the study of testing functions. Needless to say, this does not mean that one may not gain clarity and insight when working in the model of testing distributions. It only means that the latter model does fit within the more general framework of testing properties of functions.

Recall that in the context of testing distributions (cf., [4]), one is given samples of an unknown distribution and is asked to determine whether the distribution has some property (of distributions) or is statistically far from any distribution that has this property (i.e., the variation distance is large).[33] In the context of testing distributions, the key parameter is an upper bound on the size of the support of the distribution. Thus, we shall consider properties of the form $\Pi = \cup_{m\in\mathbb{N}}\Pi_m$, where Π_m is a class of distributions over a predetermined set S_m of size m. In the testing problem, the tester is given the parameter m, a distance parameter ϵ, and samples from an unknown distribution D with a support that is a subset of S_m (i.e., D assigns positive probability mass only to elements of S_m).

Definition 6.3 (testing distributions, following [4]): *We say that a distribution D is ϵ-far from Π_m if for every D' in Π_m the variation distance between D and D' is at least ϵ (i.e., $\frac{1}{2}\cdot\sum_v |\mathrm{Pr}_{r\sim D}[r=v] - \mathrm{Pr}_{r'\sim D'}[r'=v]| \geq \epsilon$). Let $\Pi = \cup_{m\in\mathbb{N}}\Pi_m$ such that each distribution in Π_m has support that is a subset of S_m and $|S_m| = m$. A distribution tester of sample complexity $s : (0,1]^2 \to \mathbb{N}$ for Π is a probabilistic algorithm T that satisfies the following two conditions for every $m \in \mathbb{N}$ and $\epsilon > 0$:*

1. *For every distribution D in Π_m, it holds that*

$$\mathrm{Pr}_{r_1,\ldots,r_s\sim D}[T(m,\epsilon,(r_1,\ldots,r_s))=1] \geq \frac{2}{3},$$

   where $s = s(m,\epsilon)$ and $(r_1,\ldots,r_s)$ is distributed according to D^s.

2. *For every distribution D with support that is a subset of S_m, if D is ϵ-far from Π_m then*

$$\mathrm{Pr}_{r_1,\ldots,r_s\sim D}[T(m,\epsilon,(r_1,\ldots,r_s))=0] \geq \frac{2}{3},$$

   where again $s = s(m,\epsilon)$ and $(r_1,\ldots,r_s)$ is distributed according to D^s.

Recall that any property of distributions as in Definition 6.3 can be tested using a sample of size $O(m/\epsilon^2)$, see [6, Lem. 3] as well as the proof of Theorem 6.2 (which establishes a somewhat weaker bound). This upper bound is obtained via a learning algorithm, and the study of distribution testing is focused on improving upon this upper bound.

It is convenient to describe distributions as above by m-long sequences of non-negative number that sum-up to 1; that is, the sequence $(p_1,\ldots,p_m)$ represents a distribution in which the i^{th} element of S_m occurs with probability p_i. Thus, variation distance (or statistical distance) between distribution corresponds to half the norm-1 value of the difference between the two corresponding sequences. This is indeed closely related to the frequency sequences used in Section 6.2.

The basic intuition is that testing a property of distributions $\Pi = \cup_{m\in\mathbb{N}}\Pi_m$ is the same as (sample-based) testing a property of functions $\Pi' = \cup_{n,m\in\mathbb{N}}\Pi'_{n,m}$ that contains all functions having a frequency sequence that corresponds to a distribution in Π. Note that the property Π' is symmetric by its very definition. The foregoing intuition is sound provided that (1) it does not matter that the frequencies that correspond to functions with domain D_n are restricted to be multiples of $1/n$,

[33] The variation distance between D and D' is $\frac{1}{2}\cdot\sum_v |\mathrm{Pr}_{r\sim D}[r=v] - \mathrm{Pr}_{r'\sim D'}[r'=v]|$, where $\mathrm{Pr}_{r\sim D}[r=v]$ denotes the probability that an element distributed according to D equals v.

and (2) the Π'-tester gains little by being provided also with uniformly distributed arguments of the function (rather than only with the value of the function on such arguments). Both conditions hold provided that n is sufficiently large (i.e., larger than some fixed polynomial in m/ϵ).

Theorem 6.4 (distribution testing reduces to sample-based testing of symmetric properties): *Let $\Pi = \cup_{m\in\mathbb{N}}\Pi_m$ be a property of distributions such that all distributions in Π_m have a support that is a subset of S_m, where $|S_m| = m$. Denote the sample complexity of Π by $s : \mathbb{N} \times (0,1] \to \mathbb{N}$. Then, there exists a* symmetric *property of functions $\Pi' = \cup_{n,m\in\mathbb{N}}\Pi'_{n,m}$ such that the following two conditions hold:*

1. *All functions in $\Pi_{n,m}$ have domain $[n]$ and range S_m.*
2. *There exists a constant c such that for every $n \geq c\cdot m^2/\epsilon^4$, the sample complexity of testing Π', denoted $s' : \mathbb{N}^2 \times (0,1) \to \mathbb{N}$, satisfies $s(m,e) = O(s'(n,m,\epsilon/2))$ and $s(m,\epsilon) = \Omega(s'(n,m,2\epsilon))$.*

Indeed, the sample complexity of testing the function class will not depend on the standard (domain) size parameter n, but rather on the auxiliary parameter m (representing the size of the range). The proof of Theorem 6.4 provides reductions between settings of parameters for one problem (e.g., m and ϵ for Π) and settings of parameters for the other problem (e.g., n, m and ϵ for Π'). In particular, for any value of ϵ, the equivalence is between ϵ-testing Π_m and $\Theta(\epsilon)$-testing $\Pi'_{\mathrm{poly}(m/\epsilon),m}$.

Proof: The proof is based on the association of properties of distributions over S_m with symmetric properties of function that range over S_m. The symmetric condition allows to disregard the identity of the argument that evaluates to a specific value, provided that the sample complexity is smaller than the square root of the domain size (since otherwise the issue of sampling without repetitions gets in our away). For the association to work, we need to assume that the tested distributions are of the form $(q_1,\ldots,q_m)$ such that each q_v is a multiple of $1/n$. In this case the said distribution can be associated with a function $f : [n] \to S_m$ such that for every $v \in S_m$ it holds that $|\{i \in [n] : f(i)=v\}| = q_v \cdot n$. For n that is sufficiently large (w.r.t m and $1/\epsilon$), the statistical distance between any distribution on S_m and such a distribution $(q_1,\ldots,q_m)$ is sufficiently small.

Fixing the property (of distributions) Π as in the hypothesis, we shall consider the following (symmetric) property (of functions) $\Pi' = \cup_{n,m\in\mathbb{N}}\Pi'_{n,m}$: *A function $f : [n] \to S_m$ is in $\Pi'_{n,m}$ if and only if there exists a distribution $(p_1,\ldots,p_m)$ in Π_m such that the max-norm distance between $(p_1\cdot n,\ldots,p_m\cdot n)$ and $(\mathtt{cnt}_1(f),\ldots,\mathtt{cnt}_m(f))$ is at most 1, where $\mathtt{cnt}_i(f) \stackrel{\rm def}{=} |\{j \in [n] : f(j)=i\}|$.*

Claim 6.4.1 *For every $n \geq \max(20s(m,\epsilon/2), 2m/\epsilon)$, it holds that $s'(n,m,\epsilon) = O(s(m,\epsilon/2))$.*

Proof: Fixing ϵ, consider a distribution tester, denoted T, for Π_m that uses $t = s(m,\epsilon/2)$ samples. We construct an ϵ-tester, denoted T', for functions in $\Pi'_{n,m}$. Machine T' is given a sequence of uniformly distributed samples that are labeled by some function $f : [n] \to S_m$; that is, T' is given a sequence $((r_1,f(r_1)),\ldots,(r_t,f(r_t)))$, where $r_1,\ldots,r_t$ are uniformly and independently distributed in $[n]$. Suppose that T' just invokes T on $(f(r_1),\ldots,f(r_m))$, and outputs whatever T does. Then, effectively, T' invokes T on samples taken from the distribution that is described by $\overline{q} = (q_1,\ldots,q_m)$ such that $q_v = \Pr_{r\in[n]}[f(r)=v]$.

Assume first that $f \in \Pi'_{n,m}$. Then, there exists $\overline{p} = (p_1,\ldots,p_m)$ in Π_m such that $\max_{i\in[m]}\{|p_i - q_i|\} \leq 1/n$. Hence, the variation distance between a sample of t elements from $\overline{q}$ and t elements from $\overline{p}$ is at most $t/n \leq 1/20$. It follows that T accepts (when invoked with samples of $\overline{q}$) with probability at least $(2/3) - (t/n) > 0.6$, and hence T' accepts f with probability at least 0.6. (We shall use straightforward error reduction in order to regain the error bound of 1/3.)

On the other hand, if $f : [n] \to S_m$ is ϵ-far from $\Pi'_{n,m}$, then $\overline{q}$ is $\epsilon/2$-far from any distribution in Π_m, because otherwise $\overline{q}$ is $((\epsilon/2) + (m/n))$-close to a distribution $\overline{p} \in \{i/n : i \in \{0, \ldots, n\}\}^m$ that is (m/n)-close to Π_m (and contradiction follows since $m/n \leq \epsilon/2$, whereas $\overline{p}$ corresponds to the frequency sequence of a function $g : [n] \to S_m$ in $\Pi'_{n,m}$). In this case, T (and so also T') rejects with probability at least 2/3. ■

Claim 6.4.2 *For every $n = \Omega(m^2/\epsilon^4)$, it holds that $s(m, \epsilon) = O(s'(m, n, \epsilon/2))$.*

Proof: Fixing ϵ, consider a function tester, denoted T', for $\Pi'_{n,m}$ that uses $t = s'(n, m, \epsilon/2)$ samples. By Theorem 6.2, it holds that $s'(n, m, \epsilon) = O(m/\epsilon^2)$. We construct an ϵ-tester, denoted T, for distributions in Π_m. Machine T is given a sequence, denoted $(v_1, \ldots, v_t)$, of samples from distribution $\overline{p} = (p_1, \ldots, p_m)$ that ranges over S_m. Suppose that T invokes T' on $((r_1, v_1), \ldots, (r_t, v_t))$, where $r_1, \ldots, r_n$ are uniformly and independently distributed in $[n]$. Since $t < \sqrt{n}/3$, these r_i's are distinct with probability at least 0.9. Let $\overline{q} = (q_1, \ldots, q_m) \in \{i/n : i \in \{0, \ldots, n\}\}^m$ be such that $\max_{i\in[m]}\{|p_i - q_i|\} \leq 1/n$. Then, effectively, T invokes T' on random samples labeled by some function $f : [n] \to S_m$ such that $\Pr_{r\in[n]}[f(r) = v] = q_v$.

Note that if $\overline{p}$ is in Π_m, then $f \in \Pi'_{n,m}$. Furthermore, the variation distance between a sample of t elements from $\overline{q}$ and t elements from $\overline{p}$ is at most $t/n < 0.01$, since $n = \Omega(t^2)$. Thus, when T' is given a sequence generated according to $\overline{p}$ (augmented with uniformly selected r_i's), it accepts with probability at least $(2/3) - (t/n) - 0.1 > 0.6$, where the 0.1 term is due to the event of colliding r_i's. It follows that T accepts $\overline{p}$ with probability at least 0.6. (Again, we shall use straightforward error reduction in order to regain the error bound of 1/3.)

On the other hand, if $\overline{q}$ is ϵ-far from Π_m, then f is $\epsilon/2$-far from $\Pi'_{n,m}$, because otherwise $g \in \Pi'_{n,m}$ that is $\epsilon/2$-close to f yields a distribution that is m/n-close to Π_m (as well as $\epsilon/2$-close to $\overline{q}$, and contradiction follows since $(m/n) + (\epsilon/2) < \epsilon$). Again, switching between $\overline{q}$ and $\overline{p}$, we infer that T' rejects with probability at least $(2/3) - (t/n) - 0.1 > 0.6$ when given a sequence generated according to $\overline{p}$ (augmented with uniformly selected r_i's), and it follows that T accepts $\overline{p}$ with probability at least 0.6. ■

This completes the proof of the theorem. ■

7 Distribution-Free Testing

Recall that distribution-free testing is a generalization of the standard notion of testing functions where the distance measure is defined according to an arbitrary distribution of the function's domain. In this case, the tester is given samples drawn according to this (unknown) distribution, where these samples are labeled by the tested function. We stress that the task is to test properties of the function, not of the distribution.[34]

For a distribution μ, by writing $r_1, \ldots, r_t \sim \mu$ we mean that $r_1, \ldots, r_t$ are selected independently such that each r_i is distributed according to μ. Denoting by μ^t the t-wise Cartesian product of μ, we can also write $(r_1, \ldots, r_t) \sim \mu^t$.

Definition 7.1 (distribution-free sample-based tester, following [14]): *Let* Π *and* $s : \mathbb{N} \times (0, 1] \to \mathbb{N}$ *be as in Definition 2.3. A* (sample-based) distribution-free tester of sample complexity s for a property Π *is a probabilistic algorithm T that satisfies the following two conditions for every $n \in \mathbb{N}$ and every distribution μ_n over D_n:*

[34] Indeed, the latter subject (i.e., testing distributions) is discussed in Section 6.3.

1. *For every* $f \in \Pi_n$ (and every $\epsilon > 0$)*, it holds that*

$$\Pr_{r_1,...,r_s \sim \mu_n}[T(n,\epsilon,(r_1,f(r_1)),\ldots,(r_s,f(r_s)))=1] \geq \frac{2}{3},$$

   where $s = s(n,\epsilon)$ *and* $(r_1,\ldots,r_s)$ *is distributed according to* μ_n^s*.*

2. *For every* $\epsilon > 0$*, and every* $f : D_n \to R_n$ *that is* ϵ*-far from* Π_n *with respect to the distribution* μ_n (i.e., $\delta_{\Pi_n}^{\mu_n}(f) > \epsilon$)*, it holds that*

$$\Pr_{r_1,...,r_s \sim \mu_n}[T(n,\epsilon,(r_1,f(r_1)),\ldots,(r_s,f(r_s)))=0] \geq \frac{2}{3},$$

   where again $s = s(n,\epsilon)$ *and* $(r_1,\ldots,r_s)$ *is distributed according to* μ_n^s*. We say that* $f : D_n \to R_n$ is ϵ-far from Π_n with respect to the distribution μ_n *if* $\delta_{\Pi_n}^{\mu_n}(f) > \epsilon$*, where*

$$\delta_{\Pi_n}^{\mu_n}(f) \stackrel{\text{def}}{=} \min_{g \in \Pi_n}\{\Pr_{x \sim \mu_n}[f(x) \neq g(x)]\} \ . \qquad (7)$$

Again, if the tester accepts every function in Π *with probability 1, then we say that it has* one-sided error*.*

Definition 2.3 refers to the case that μ_n is uniform on D_n. In what follows, the (unqualified) term "sample-based tester" refers to testers that work only for this (uniform) distribution, as defined in Definition 2.3.

7.1 A general result

We start by proving the first item in Theorem 1.3. Namely, we show that the sample complexity of distribution-free testing cannot be significantly smaller than the sample complexity of one-sided error testing (with respect to the uniform distribution). Specifically, we show that the latter is at most nearly-quadratic in the former.

Theorem 7.2 (distribution-free testers imply one-sided error testers under the uniform distribution): *For every property* Π*, if* Π *has a sample-based distribution-free tester of sample complexity* $s : \mathbb{N} \times (0,1] \to \mathbb{N}$*, then* Π *has a one-sided error sample-based tester of sample complexity* $\widetilde{O}(s^2)$ (under the uniform distribution)*.*

The sample complexity of the one-sided error sample-based tester is actually $O(s^2)$ if $s < \sqrt{n}/25$ and $O(n \log n)$ otherwise. While an upper bound of $O(s^2)$ is quite likely, we shall show (in Theorem 7.4) that an upper bound of $O(s)$ cannot hold in general. We stress that Theorem 7.2 (as well as the last sentence) refer to an upper bound that holds for *any* property (rather than to specific cases, as studied in Section 7.2).

Proof: Let T' be a sample-based distribution-free tester of sample complexity $s : \mathbb{N} \times (0,1] \to \mathbb{N}$ for Π. Assume that T' has error probability at most 1/6 (rather than at most 1/3),[35] and that $s < \sqrt{n}/25$ (to be justified at the end). Consider the following sample-based algorithm T, of sample complexity $t = O(s^2)$. On input parameters n, ϵ and a sequence $((r_1, v_1), \ldots, (r_t, v_t))$, where the v_i's are determined by the tested function, algorithm T accepts if and only if there exists $g \in \Pi_n$

[35] Indeed, this is justified by straightforward error reduction, which merely increases s by a constant factor.

such that $g(r_i) = v_i$ for every $i \in [t]$. We claim that T is a one-sided error tester for Π under the uniform distribution.

By its construction, T accepts every function in Π with probability 1. Hence, if T is not a one-sided error tester for Π (under the uniform distribution), then it must be that it accepts some no-instance (i.e., a function far from Π) with too high probability (i.e., probability exceeding $1/3$). Thus, we assume towards the contradiction that there exist n, ϵ and $f : D_n \to R_n$ such that *f is ϵ-far from Π_n* (under the uniform distribution) *and T accepts f with probability at least* $1/3$ (when given $t = t(n, \epsilon)$ uniformly distributed samples that are labeled according to f). By the construction of T, it follows that for at least one third of the possible $\overline{r} = (r_1, \ldots, r_t) \in D_n^t$ there exists $g_{\overline{r}} \in \Pi_n$ such that $g_{\overline{r}}(r_i) = f(r_i)$ for every $i \in [t]$. For each such $\overline{r} = (r_1, \ldots, r_t)$, consider the distribution $\mu_{\overline{r}}$ that is uniform over the corresponding (multi)set $\{r_1, \ldots, r_t\}$. The key observation is that T' must accept a sequence of s samples drawn from $\mu_{\overline{r}}$ that are labeled according to f with probability at least $5/6$, since these labels are consistent with $g_{\overline{r}} \in \Pi$. (The fact that f is ϵ-far from Π according to the uniform distribution on D_n is irrelevant here, and in particular $\delta_{\Pi_n}^{\mu_{\overline{r}}}(f) = 0$.)[36] Thus, the following holds.

Claim 7.2.1 *Let T and f be as in the foregoing paragraph, and let $X = (X_1, \ldots, X_s)$ denote a random sequence produced by first selecting uniformly $\overline{r} \in D_n^t$, and then picking s independent samples from the distribution $\mu_{\overline{r}}$. Then, on input n, ϵ and $((X_1, f(X_1)), \ldots, (X_s, f(X_s))$, algorithm T' accepts with probability at least $\frac{1}{3} \cdot \frac{5}{6}$.*

On the other hand, on input n, ϵ and $((Y_1, f(Y_1)), \ldots, (Y_s, f(Y_s))$, where the Y_i's are independently and uniformly distributed in D_n, algorithm T' must reject with probability at least $5/6$, because by the hypothesis f is ϵ-far from Π_n (under the uniform distribution). If $t \leq n$, then this yields a contradiction, because (as shown next) the two distribution (i.e., X and Y) are 0.1-close.

Claim 7.2.2 *Suppose that $500s^2 < t \leq n$, and let $X = (X_1, \ldots, X_s)$ and $Y = (Y_1, \ldots, Y_s)$ be as above; that is, X is as defined in Claim 7.2.1 and Y is a sequence of independently and uniformly distributed elements of D_n. Then, the variation distance between X and Y is at most* 0.1.

Recall that $t = O(s^2)$ by construction, whereas $t \leq n$ was assumed upfront and will be justified later.

Proof: Note that conditioned on X containing no collision (i.e., $X_i = X_j$ for $i \neq j$), the sequence X is uniformly distributed over all s-long sequences of distinct elements of D_n. The same holds for Y, and thus it suffices to prove that, in both X and Y, the probability that the same element appears twice is at most $1/20$. The claim is obvious for Y, since the collision probability is at most $\binom{s}{2} \cdot \frac{1}{n} < 1/1000$.

Turning to X, we note that the argument is equally easy in the case that $40t^2 < n$ (since $\binom{t}{2} \cdot \frac{1}{n} + \binom{s}{2} \cdot \frac{1}{t} < 1/80 + 1/1000$), alas we only have $t \leq n$. Nevertheless, we first upper bound by $1/40$ the probability that a uniformly chosen sequence $\overline{r} = (r_1, \ldots, r_t) \in D_n^t$ contains more than $20t$ collisions (i.e., pairs (i, j) such that $r_i = r_j$). This follows by noting that the expected number of collisions is $\binom{t}{2} \cdot \frac{1}{n} < \frac{t}{2}$. Fixing an arbitrary sequence $\overline{r}$ with at most $20t$ collisions, we consider the probability that a sample of s elements taken from $\overline{r}$ has a collision. Such collisions may be due either to a collision in $\overline{r}$ or to picking the same position in $\overline{r}$ twice. Thus, this probability is at most $\binom{s}{2} \cdot (\frac{20t}{t^2} + \frac{1}{t}) < 1/40$, and the claim follows. ■

[36] We stress that our argument is not based on the fact that $\delta_{\Pi_n}^{\mu_{\overline{r}}}(f) = 0$ but rather on the fact that the samples labeled by f can be thought to be labeled by $g_{\overline{r}} \in \Pi$.

Finally, we get to our assumption that $t \leq n$. Recall that $t = O(s^2)$, hence if $t > n$, then $s = \Omega(\sqrt{n})$. But in this case, with high probability, a labeled (uniformly distributed) sample of size $O(s^2 \log s)$ allows to reconstruct the function, and the theorem follows. ■

Remark 7.3 (a general phenomenon): *The proof of Theorem 7.2 is actually oblivious to the specifics of the promise problem at stake* (i.e., the fact that we refer to property testing). *Indeed, it can be stated for any promise problem (P, Q), where the promise is that the input is in P and the question is whether it is in Q:*

> *If a promise problem (P, Q) has a sample-based distribution-free two-sided error decider of sample complexity s, then (P, Q) has a one-sided error sample-based decider of sample complexity $\widetilde{O}(s^2)$* (under the uniform distribution)*.*

The proof shows that if $f \in P \setminus Q$ is accepted with probability greater than $1/3$ by the canonical one-sided error decider, which accepts a function if and only if the sample is consistent with some $g \in P \cap Q$, then a distribution-free decider for (P, Q) cannot have sample complexity s, because when testing f it must err with probability greater than $1/6$ either on the uniform distribution on some set of size $O(s^2)$ or on the uniform distribution on f's entire domain.

7.2 Specific results

In this section we consider several cases that indicate that there is no fixed relation between the sample complexity of one-sided error testers under the uniform distribution and the sample complexity of distribution-free testers. The first three cases are based on natural properties. We recall the notation that was introduced in the introduction: $\mathtt{OSE}(\Pi)$ denotes the sample complexity of one-sided error sample-based testing Π (under the uniform distribution); that is, $\mathtt{OSE}(\Pi)$ is a function $s(n, \epsilon)$ that represents the number of samples made by the best such tester on input parameters n and ϵ. Likewise, $\mathtt{DF}(\Pi)$ denotes the sample complexity of distribution-free (sample-based) testing Π. Indeed, Theorem 7.2 asserts that for every Π it holds that $\mathtt{OSE}(\Pi) = \widetilde{O}(\mathtt{DF}(\Pi)^2)$.

We first show that a general upper bound as in Theorem 7.2 (i.e., an upper bound on $\mathtt{OSE}$ in terms of $\mathtt{DF}$) cannot be linear (let alone sub-linear). That is, there exist (natural) properties Π such that $\mathtt{OSE}(\Pi) = \Omega(\text{poly}(\epsilon) \cdot \mathtt{DF}(\Pi) \log \mathtt{DF}(\Pi))$, as stated in Item 2.

Theorem 7.4 (distribution-free sample-complexity may be lower than the one-sided error sample complexity under the uniform distribution): *There exists a property Π such that*

1. *Any one-sided error tester for Π* (under the uniform distribution) *must have query complexity $q(n, \epsilon) = \Omega(n)$ for every $\epsilon < 1/3$.*
2. *There exists a* (sample-based) *distribution-free tester for Π of sample complexity $s(n, \epsilon) =$* $\text{poly}(1/\epsilon) \cdot \frac{n}{\log n}$.

For example, Π_n may be the set of Boolean functions over $[n]$ that evaluate to 1 more often than to 0.

It follows that there are properties for which the one-sided error sample complexity (under the uniform distribution) is higher than the distribution-free sample complexity. Note that Π is a symmetric property, and so queries are not more powerful for it than samples (see Theorem 6.1).

Proof: We consider the class, denoted Π_n, of Boolean functions over $[n]$ such that $f : [n] \to \{0, 1\}$ is in Π_n if and only if $|\{i \in [n] : f(i) = 1\}| > n/2$.

Claim 7.4.1 *Any one-sided error tester for* Π_n (under the uniform distribution) *has query complexity* $q(n, \epsilon) \geq n/2$ *for any* $\epsilon < 1/2$.

Proof: A one-sided error tester may reject only if the answers it has obtained are not consistent with any function in Π_n, which means that it must see at least $n/2$ zeros. On the other hand, when having oracle access to a function with at least $(0.5 + \epsilon) \cdot n$ zeros, which is ϵ-far from Π_n, the tester must reject. The claim follows. ■

Claim 7.4.2 *There exists a* (sample-based) *distribution-free tester of sample complexity* $s(n, \epsilon) = \widetilde{O}(1/\epsilon^2) \cdot \frac{n}{\log n}$ *for* Π.

Proof: We shall rely on an algorithm of Valiant and Valiant [34], which approximates the support size of an unknown distribution over $[n]$ based on $O(n/\log n)$ samples of the distribution. Specifically, we use the following result.

> **Theorem 1 in [34]** (rephrased): *There exists an algorithm* $A_{\rm vv}$ *that on input a parameter* δ *and* $\widetilde{O}(1/\delta^2) \cdot n/\log n$ *samples of a random variable, denoted* X_n*, that ranges over* $[n]$*, outputs a representation of a distribution* Y_n *that with overwhelmingly high probability is* δ*-close to a distribution* X'_n *that equals* X_n *up to relabeling.*
>
> By saying that X equals X' up to relabeling we mean that there exists a permutation $\pi : [n] \to [n]$ such that for every $v \in [n]$ it holds that $\Pr_{x \sim X'}[x = v] = \Pr_{x \sim X}[x = \pi(v)]$.

(Actually, the representation of Y_n is succinct so to allow the algorithm to run in time that is linear in the number of samples that it obtains, but this is irrelevant to us here.) Now, suppose that, on input n and ϵ, we get $s = s(n, \epsilon)$ samples drawn according to an unknown distribution μ_n and labeled by an unknown Boolean function f. Our task is to decide whether f is in Π_n or is ϵ-far from Π_n with respect to the distribution μ_n. We propose the following algorithm.

1. The algorithm uses a small portion of the label sample (i.e., $O(1/\epsilon^2)$ samples) in order to approximate the probability $p_n \stackrel{\rm def}{=} \Pr_{r \sim \mu_n}[f(r) = 0]$ up to an additive error of $\epsilon/4$. If the estimate, denoted $\widetilde{p}$, is below $\epsilon/2$, then the algorithm halts with output 1 (i.e., it accepts).

   Assuming the algorithm did not halt in Step 1, we may assume that $p_n > \epsilon/4$.

2. The algorithm defines X_n as the distribution μ_n conditioned on f evaluating to 0; that is, $\Pr_{x \sim X_n}[x = v] = \Pr_{x \sim \mu_n}[x = v | f(x) = 0]$. It invokes algorithm $A_{\rm vv}$ with parameter $\delta = \epsilon/4\widetilde{p}$ providing it with $\widetilde{O}(1/\delta^2) \cdot n/\log n$ samples of X_n, and obtaining a description of a distribution Y_n.

   Samples of X_n are obtained from the s labeled samples provided to the main algorithm; specifically, the main algorithm forwards the first $s_0 \stackrel{\rm def}{=} \widetilde{O}(1/\delta^2) \cdot n/\log n$ samples that are labeled 0 to $A_{\rm vv}$. With very high probability, the number of samples labeled 0 is at least $p_2 \cdot s/2 \geq \widetilde{p}^2 \cdot s/4$, which equals s_0 (since we may set the parameters such that $s_0 = \epsilon^2 s/64\delta^2$). which equals $\widetilde{O}(1/\delta^2) \cdot n/\log n$.

3. The algorithm accepts if and only if Y_n is δ-close to a distribution that has support size smaller than $n/2$.

We now turn to the analysis of the foregoing algorithm. Consider first the case that $f \in \Pi_n$. If $p_0 \leq \epsilon/4$, then (by an additive Chernoff bound) with high constant probability $\widetilde{p} < \epsilon/2$, and the algorithm accepts. Otherwise (by a multiplicative Chernoff bound) with high constant probability, $\widetilde{p} \leq 2p_0$. Assuming the algorithm does not accept due to $\widetilde{p}$ being smaller than $\epsilon/2$ (so that it continues to its second step), with high probability the distribution Y_n (output by $A_{\rm vv}$) is δ-close to a relabeling of X_n, which in turn has support size smaller than $n/2$. Hence, in either cases, the algorithm accepts with high probability.

Now suppose that f is ϵ-far from Π_n with respect to the distribution μ_n. In this case $p_n > \epsilon$, and with high probability it holds that $\widetilde{p} > p_n/2 \geq \epsilon/2$, which means that the algorithm proceeds to Step 2. Furthermore, X_n is ϵ/p_n-far from having support size smaller than $n/2$, since otherwise we can modify f in a corresponding manner in contradiction to the hypothesis (i.e., if X_n is ϵ/p_n-close to X'_n that has support smaller than $n/2$, then setting $f'(x) = 1$ iff x is not in the support of X'_n yields a function in Π_n that is ϵ-close to f w.r.t the distribution μ_n). Recalling that $\delta = \epsilon/4\widetilde{p}$ (and $\widetilde{p} > p_n/2$, which implies that $\epsilon/p_n > \epsilon/2\widetilde{p} = 2\delta$), it follows that (w.v.h.p.) Y_n is $(2\delta - \delta)$-far from any distribution having support smaller than $n/2$, which implies that our algorithm rejects. ■

Theorem 7.4 follows. ■

On the non-tightness of Theorem 7.2. Having just shown that OSE may be larger than DF, we now turn to cases in which OSE is at most linear in DF, which in particular means that in these cases Theorem 7.2 is not tight. The first case is of a property Π such that $\mathtt{OSE}(\Pi) = \Theta(\mathtt{DF}(\Pi))$ holds (Item 3 in Theorem 1.3), and the other cases are ones in which OSE is significantly smaller than DF.

Theorem 7.5 (distribution-free sample-complexity may equal the one-sided error sample complexity under the uniform distribution): *There exists a property* Π *such that*

1. *Any sample-based tester for* Π (under the uniform distribution) *has sample complexity* $s(n,\epsilon) = \Omega(\sqrt{n/\epsilon})$, *provided that* $\epsilon \geq 1/n$. *Indeed, this lower bound holds also for two-sided error testers.*

2. *There exists a* (sample-based) *distribution-free tester for* Π *of sample complexity* $s(n,\epsilon) = O(\sqrt{n/\epsilon})$. *Furthermore, this tester has one-sided error.*

For example, Π_n *may be the set of monotone functions over* $[n]$.

It follows that there are properties for which the one-sided error sample complexity (under the uniform distribution) is at least linear in the distribution-free sample complexity. Indeed, for this assertion it would have suffices to have a lower bound for one-sided error testers (under the uniform distribution) and an upper bound for two-sided error (distribution-free) testers. However, the stronger assertion of Theorem 7.5 implies that all four complexity measures are linearly related: The one-sided and two-sided error sample complexities for both the uniform distribution and the distribution-free case are all linearly related.

Proof: For an arbitrary ordered set R_n, we consider the class, denoted Π_n, of monotone function from $[n] \to R_n$; that is, $f : [n] \to R_n$ is in Π_n if and only if $f(i) \leq f(j)$ for every $i < j$.

Claim 7.5.1 *For* $R_n = [n]$, *any sample-based tester for* Π_n (under the uniform distribution) *has sample complexity* $s(n,\epsilon) = \Omega(\sqrt{n/\epsilon})$, *provided that* $\epsilon \geq 1/n$.

Our guess is that Claim 7.5.1 was discovered before by several researchers, but we are unaware of a prior publication of it.

Proof: Assume for simplicity that n is even, and let $n' = n/2$. Furthermore, assume that $\epsilon > c/n$ for a sufficiently large constant c. We consider only functions f that satisfy $f(2i-1), f(2i) \in \{2i-1, 2i\}$ for every $i \in [n']$. For any $m \in \{0, \ldots, n'\}$, consider the class of functions F_m such that $f \in F_m$ if $\{f(2i-1), f(2i)\} = \{2i-1, 2i\}$ for every $i \leq m$ and $f(2i-1) = f(2i) \in \{2i-1, 2i\}$ for every $i > m$. Note that $F_0 \subset \Pi_n$, while (for $m > c$) with very high constant probability a function selected uniformly in F_m is $m/3n$-far from Π_n. We shall show that $\Omega(n/\sqrt{m})$ samples are requited in order to distinguish a function uniformly selected in F_0 from a function uniformly selected in F_m.

Let $X = ((X_1, g(X_1)), \ldots, (X_t, g(X_t))$ denote a sequence of pairs such that the X_i's are uniformly and independently distributed in $[n]$ and g is uniformly distributed in F_0. Likewise, $Y = ((Y_1, f(Y_1)), \ldots, (Y_t, f(Y_t))$ denotes a sequence where the Y_i's are uniformly and independently distributed in $[n]$ and f is uniformly distributed in F_m. Let $\chi((z_1, v_1), \ldots, (z_t, v_t)) = 1$ if there exists $i \in [m]$ such that $|\{j : z_j \in \{2i-1, 2i\}\}| > 1$ and $\chi((z_1, v_1), \ldots, (z_t, v_t)) = 0$ otherwise. Then, condition on $\chi = 0$, the random variables X and Y are identically distributed; that is, for every γ, it holds that $\Pr[X = \gamma | \chi(X) = 0] = \Pr[Y = \gamma | \chi(Y) = 0])$. On the other hand $\Pr[\chi(X) = 1] = \Pr[\chi(Y) = 1]$ is upper bounded by $\binom{t}{2} \cdot \frac{2m}{n} \cdot \frac{1}{n} < \frac{t^2 m}{n^2}$. Thus, for $t < n/\sqrt{10m}$, the variation distance between X and Y is smaller than $1/10$. Using $m = 3\epsilon n$, the claim follows. ■

Claim 7.5.2 *For any ordered set R_n, there exists a* (sample-based) *distribution-free tester of sample complexity $s(n, \epsilon) = O(\sqrt{n/\epsilon})$ and one-sided error for Π.*

Proof: The algorithm simply accepts the function f if and only if the sample obtained is consistent with a monotone function (i.e., there exists a monotone function g such that $f(r) = g(r)$ for every r that appears in the sample). Hence, this algorithm accepts every $f \in \Pi_n$ with probability 1. We now consider an arbitrary function f that is ϵ-far from Π_n with respect to some distribution μ (i.e., $\delta^{\mu}_{\Pi_n}(f) > \epsilon$).

Following [7], we consider the ("violation") graph $G_f = ([n], E_f)$ such that $(i, j) \in E_f$ if and only if $i < j$ and $f(i) > f(j)$. (The edges of G_f are called violating edges.) Note that every vertex cover of this graph has weight greater than ϵ, where the weight of any set of vertices is the sum of the probabilities assigned to its elements under μ. (Otherwise, modifying the function on this vertex cover, in an adequate manner, yields a monotone function that is ϵ-close to f.)[37] Fixing a minimum weight vertex cover C of G_f, we note that every $S \subseteq C$ must have weight that is not larger than the weights of the set of its neighbors (i.e., of the set $\bigcup_{v \in S} \Gamma_f(v)$, where $\Gamma_f(v)$ denotes the set of neighbors of v in G_f).[38] We shall show that, with high probability, a sample of $s = O(\sqrt{n/\epsilon})$ vertices (i.e., points in $[n]$), drawn independently from μ, contains a violating edge.

First, we consider a sample, denoted S_1, of $s/2$ vertices drawn independently from μ. The expected weight of $S_1 \cap C$ is at least $(s/2) \cdot \epsilon/n$, and, with very high probability, its weight is at least $\epsilon s/4n$. Fixing such a set S_1, we consider a second sample, denoted S_2, of $s/2$ vertices drawn independently from μ. Since the weight of the set of neighbors of S_1 is at least $\epsilon s/4n$, each of these $s/2$ samples is a neighbor of some vertex of S_1 with probability at least $\epsilon s/4n$. Hence, the

[37] If i is in the vertex cover and $j < i$ is the largest integer that is not in the cover, then we let $f(i) = f(j)$. This simple modification rule benefit from the fact that the domain is totally ordered. Nevertheless, a similar modification is possible for any partially ordered domain; see [9, Lem. 1].

[38] Otherwise, we can replace S by $\bigcup_{v \in S} \Gamma_f(v)$, obtaining a vertex cover of smaller weight. (Note that edges with an endpoint in S are covered by the other endpoint, which must be in $\bigcup_{v \in S} \Gamma_f(v)$.)

probability that $S_1 \times S_2$ contains no violating edge is at most $(1-(\epsilon s/4n))^{s/2}$, which is very small since $\epsilon s^2/8n \gg 1$. The claim follows. ■

Theorem 7.5 follows. ■

We next establish Item 4 in Theorem 1.3.

Theorem 7.6 (distribution-free sample-complexity may be exponential in the one-sided error sample complexity under the uniform distribution): *There exists a property* Π *such that*

1. *Any sample-based distribution-free tester for* Π *has sample complexity* $s(2^\ell,\epsilon) = \Omega(\sqrt{\ell}/\epsilon)$*. Indeed, this lower bound holds also for two-sided error testers.*

2. *There exists a sample-based one-sided error tester for* Π (under the uniform distribution) *having sample complexity* $s(2^\ell,\epsilon) = \widetilde{O}(\epsilon^{-1}\log \ell)$*.*

For example, Π_{2^ℓ} *may be the set of monomials over* $\{0,1\}^\ell$*.*

It follows that there are properties for which the one-sided error sample complexity (under the uniform distribution) is logarithmic in the distribution-free sample complexity (up to a factor of $1/\epsilon$). We comment that the upper bound in Item 2 is tight: Any sample-based tester for Π (under the uniform distribution) has sample complexity $s(2^\ell,\epsilon) = \Omega(\epsilon^{-1}\log \ell)$, provided that $\epsilon > 2^{-(\ell-\ell^{0.01})}$. (This lower bound holds also for two-sided error testers; see Appendix A.3 for details.)

Proof: We consider the property $\Pi = \bigcup_{\ell\in\mathbb{N}} \Pi_{2^\ell}$ such that Π_{2^ℓ} consists of Boolean functions $f : \{0,1\}^\ell \to \{0,1\}$ that can be represented by monomials (i.e., $f(x_1,\ldots,x_\ell)$ can be written as a conjunction of a subset of the literals $\{x_1,\overline{x}_1,\ldots,x_\ell,\overline{x}_\ell\}$). Glasner and Servedio [10] proved that the distribution-free *query* complexity of testing Π_{2^ℓ} is $\ell^{0.2-o(1)}$; here we shall present a much simpler proof (of a higher lower bound) for the case that only samples are allowed. Turning to the upper bound, we note that Parnas *et al.* [27] proved that the query complexity of this property under the uniform distribution is $O(1/\epsilon)$; here we present a higher upper bound, but for sample-based tester. We stress that our sample-based tester has one-sided error.

Claim 7.6.1 *Any sample-based distribution-free tester for* Π *has sample complexity* $s(2^\ell,\epsilon) = \Omega(\sqrt{\ell}/\epsilon)$*.*

We note that a higher lower bound of $\Omega(\ell^{2/3})$ follows from a more complex proof of Feige (priv. comm., 2011).

Proof: We first establish the lower bound for $\epsilon = \Omega(1)$, and later extend it to a general ϵ. To this end we define two sets, $\mathcal{Y}$ and $\mathcal{N}$, of pairs (f,D), where f is a Boolean function over $\{0,1\}^\ell$ and D is a distribution over $\{0,1\}^\ell$. For each pair (f,D) in $\mathcal{Y}$, the function f is a monomial, and for each pair (f,D) in $\mathcal{N}$ the function f is $\Omega(1)$-far from being a monomial with respect to the underlying distribution D. In fact, the set $\mathcal{N}$ consists of a single pair $(f^{\rm no},D^{\rm no})$. By Definition 7.1, any (sample based) distribution-free testing algorithm must reject $f^{\rm no}$ with high constant probability when given a sample generated by $D^{\rm no}$ and labeled by $f^{\rm no}$, and must accept with high constant probability every f such that $(f,D)\in\mathcal{Y}$ when given a sample generated by D and labeled by f. We show that for $s \le \sqrt{\ell}/8$, the variation distance between a labeled sample of size s that is generated by $D^{\rm no}$ and labeled by $f^{\rm no}$ and a labeled sample of size s that is generated and labeled according to a pair (f,D) that is selected uniformly in $\mathcal{Y}$, is a small constant. This implies a lower bound of

$\Omega(\sqrt{\ell})$ (for constant ϵ) on the sample complexity of any (sample based) distribution-free tester for monomials. Details on the construction of $\mathcal{Y}$ and $\mathcal{N}$ follow.

Assume for simplicity that ℓ is even and consider a partition of the variables $x_1, \ldots, x_\ell$ into pairs (x_{2i-1}, x_{2i}) for $i \in [\ell/2]$. We start by defining $D^{\rm no}$ and $f^{\rm no}$. For each $i \in [\ell/2]$ and $(b, b') \in \{00, 01, 10\}$, let $\alpha^i(b, b')$ be the string that is all 1s except for bits in positions $2i - 1$ and $2i$, which are b and b', respectively; that is, $\alpha^i(b, b') = (11)^{i-1}bb'(11)^{\ell/2-i}$. The distribution $D^{\rm no}$ is uniform over the set of strings $S^{\rm no} \stackrel{\rm def}{=} \{\alpha^i(b, b') : i \in [\ell/2], (b, b') \in \{00, 01, 10\}\}$, and for each $i \in [\ell/2]$, we have $f^{\rm no}(\alpha^i(00)) = 0$, and $f^{\rm no}(\alpha^i(01)) = f^{\rm no}(\alpha^i(10)) = 1$. It is irrelevant to the argument how $f^{\rm no}$ is defined outside of $S^{\rm no}$, so we set it arbitrarily to 0.

We observe that $f^{\rm no}$ is 1/3-far from being a monomial with respect to $D^{\rm no}$. This is true since for every (disjoint) triple of strings $\alpha^i(00), \alpha^i(01), \alpha^i(10)$ the value of the function must be modified on at least one of the three strings so that the function become consistent with some monomial.

We turn to defining the set $\mathcal{Y}$. For each $\beta \in \{00, 01, 10\}^{\ell/2}$, the set $\mathcal{Y}$ contains the pair (f^β, D^β), where the distribution D^β is uniform over the set of $\ell/2$ strings $S^\beta \stackrel{\rm def}{=} \{\alpha^i(\beta_{2i-1}, \beta_{2i}) : i \in [\ell/2]\}$ (so that $S^\beta \subset S^{\rm no}$), and f^β is the (monotone) monomial that is the conjunction of the following variables. For each $i \in [\ell/2]$, if $\beta_{2i-1} = 1$, then x_{2i-1} is a variable of f^β and x_{2i} is not; if $\beta_{2i} = 1$, then x_{2i} is a variable of f^β and x_{2i-1} is not; and, if $\beta_{2i-1} = \beta_{2i} = 0$, then both x_{2i-1} and x_{2i} are variables of f^β. By this definition, $f^\beta(\alpha^i(\beta_{2i-1}, \beta_{2i})) = 1$ when $\beta_{2i-1}\beta_{2i} \in \{01, 10\}$ and $f^\beta(\alpha^i(\beta_{2i-1}, \beta_{2i})) = 0$ when $\beta_{2i-1} = \beta_{2i} = 0$.

Note that for each $\beta \in \{00, 01, 10\}^{\ell/2}$, the functions $f^{\rm no}$ and f^β agree on all strings in S^β, and so the distance between them with respect to D^β is 0. This is in contrast to the fact that $f^{\rm no}$ and f^β disagree on a third of the strings in $S^{\rm no}$, and so the distance between them with respect to $D^{\rm no}$ is 1/3. Also note that for a fixed choice of β, the distribution over samples generated by D^β and labeled by f^β differs significantly from the distribution over samples generated by $D^{\rm no}$ and labeled by $f^{\rm no}$, even for constant size samples. This is true simply because with high constant probability we get a sample point from $S^{\rm no} \setminus S^\beta$. However, if we also select $\beta \in \{00, 01, 10\}^{\ell/2}$ uniformly at random, then the corresponding distribution over labeled samples will be close to that generated by $D^{\rm no}$ and labeled by $f^{\rm no}$ as long the sample size is sufficiently smaller than $\sqrt{\ell}$. Details follow.

Consider the following two distributions over labeled samples $((r^1, v^1), \ldots, (r^s, v^s))$ of size s where $r^j \in \{0, 1\}^\ell$ and $v^j \in \{0, 1\}$ for each $1 \le j \le s$. The first distribution, $L^{\rm no}$, generates a labeled sample simply by selecting each r^j independently according to $D^{\rm no}$ and setting $v^j = f^{\rm no}(r^j)$. The second distribution, $L^{\rm yes}$, generates a labeled sample by first selecting $\beta \in \{00, 01, 10\}^{\ell/2}$ uniformly at random and then generating each r^j independently according to D^β and setting $v^j = f^\beta(r^j)$.

An alternative (equivalent) formulation of the generation process of samples distributed according to $L^{\rm no}$ is the following: For each $1 \le j \le s$, select $i \in [\ell/2]$ uniformly at random, select $(b, b') \in \{00, 01, 10\}$ uniformly at random, and let $r^j = \alpha^i(b, b')$ (and $v^j = f^{\rm no}(r^j)$). Similarly, we can generate a sample distributed according to $L^{\rm yes}$ in the following manner. Initially, set $\beta = ?^\ell$ (indicated that no bit in β is determined). For each $1 \le j \le s$, select $i \in [\ell/2]$ uniformly at random. If $\beta_{2i-1} = ?$, then select $(b, b') \in \{00, 01, 10\}$ uniformly at random and set $\beta_{2i-1} = b$, $\beta_{2i} = b'$. Now let $r^j = \alpha^i(\beta_{2i-1}, \beta_{2i})$ and let $v^j = 1$ if $(\beta_{2i-1}, \beta_{2i}) \in \{01, 10\}$ and $v^j = 0$ otherwise. Given this alternative description, we observe that conditioned on the event that no $i \in [\ell/2]$ is selected more than once, the two distributions on samples are exactly the same. By the (lower bound of) the "Birthday Paradox", if $s \le \sqrt{\ell}/c$ and c is a sufficiently large constant, then, with high constant probability over both generation processes, no i is selected more than once. It follows that the statistical distance between $L^{\rm no}$ and $L^{\rm yes}$ is a small constant.

It remains to extend the argument to any $\epsilon \le 1/3$. This is done by adding the string 1^ℓ to $S^{\rm no}$ as well as to each S^β, while assigning 1^ℓ weight $1 - 3\epsilon$, and setting $f^{\rm no}(1^\ell) = f^\beta(1^\ell) = 1$. Namely,

$D^{\rm no}(1^\ell) = D^\beta(1^\ell) = 1-3\epsilon$, and $D^{\rm no}$ as well as each D^β is uniform over the remaining strings in its support. By this modification, the distance with respect to $D^{\rm no}$ between $f^{\rm no}$ and any monomial is at least ϵ. By combining the argument given for the case $\epsilon = 1/3$ with the fact that the probability that each sample point differs from 1^ℓ (both when the sample is generated according to $D^{\rm no}$ and when it is generated according to D^β for some β) is $O(\epsilon)$, the lower bound of $\Omega(\sqrt{\ell}/\epsilon)$ follows. ■

Claim 7.6.2 *There exists a sample-based one-sided error tester for* Π (under the uniform distribution) *having sample complexity* $s(2^\ell, \epsilon) = \widetilde{O}(\epsilon^{-1}\log\ell)$.

Proof: As noted at the end of Section 2.2, the operation of a sample-based one-sided tester is totally determined by its sample: Being sample-based, this tester has no control over its access to the function, and having one-sided error it has no real control on its decision; that is, without loss of generality, the tester accepts if and only if the labeled sample that it has obtained is consistent with some function in Π. Thus, the claim amounts to proving that *if* $f : \{0,1\}^\ell \to \{0,1\}$ *is* ϵ*-far from* Π*, then, with probability at least* $2/3$*, no function in* Π *is consistent with the labeling given by* f *to a random sample of* $\widetilde{O}(\epsilon^{-1}\log\ell)$ *points.* Fixing ℓ and ϵ for the rest of the discussion, and letting $t \stackrel{\rm def}{=} \lceil \log_2(2/\epsilon) \rceil$, the proof is based on two observations:

1. If f is ϵ-far from Π and $g \in \Pi$ is a monomial that contains exactly t literals, then $\Pr_{x\in\{0,1\}^\ell}[f(x)\neq 0 = g(x)] > \epsilon/2$, where here and throughout the entire proof all probabilities are taken uniformly over $\{0,1\}^\ell$.

   This holds since f is ϵ-far from g, whereas $\Pr[f(x)\neq 1 = g(x)] \leq \Pr[g(x)=1] \leq 2^{-t} \leq \epsilon/2$.

2. If $g \in \Pi$ contains more than t literals, then there exists $h \in \Pi$ that contains exactly t literals such that $h^{-1}(0)$ is a subset of $g^{-1}(0)$; that is, if $h(x) = 0$ (for some x), then $g(x) = 0$.

   This holds by considering an arbitrary monomial $g' \in \Pi'$ that contains exactly t of the literals that appear in g.

Let Π^{ls} (resp., Π^{eq}) denote the set of monomials containing less than (resp., exactly) t literals. Suppose that f is ϵ-far from Π, and let $s = O(\epsilon^{-1} t \log\ell) = \widetilde{O}(\epsilon^{-1}\log\ell)$. Then, using $\Pr_r[f(r) = g(r)] \leq 1-\epsilon$ for every $g \in \Pi^{\mathsf{ls}}$,

$$\begin{aligned}
&\Pr_{r_1,\dots,r_s\in\{0,1\}^\ell}\left[\forall g \in \Pi^{\mathsf{ls}}\ \exists i \in [s] \text{ s.t. } f(r_i) \neq g(r_i)\right] \\
&\quad = \quad 1 - \Pr_{r_1,\dots,r_s\in\{0,1\}^\ell}\left[\exists g \in \Pi^{\mathsf{ls}} \text{ s.t. } \forall \in [s]\, f(r_i) = g(r_i)\right] \\
&\quad \geq \quad 1 - \sum_{j\in[t-1]} \binom{2\ell}{j} \cdot (1-\epsilon)^s \ > \ \frac{5}{6}\,.
\end{aligned} \tag{8}$$

Similarly, using Observation 1, we have

$$\begin{aligned}
&\Pr_{r_1,\dots,r_s\in\{0,1\}^\ell}\left[\forall g \in \Pi^{\mathsf{eq}}\ \exists i \in [s] \text{ s.t. } f(r_i) \neq 0 = g(r_i)\right] \\
&\quad \geq \quad 1 - \binom{2\ell}{t} \cdot (1-\epsilon/2)^s \ > \ \frac{5}{6}\,.
\end{aligned} \tag{9}$$

Fixing any sequence $(r_1,\dots,r_s)$ that satisfies the conditions in both Eq. (8) and Eq. (9), and using Observation 2, we conclude that for every $g \in \Pi$ there exists $i \in [s]$ such that $f(r_i) \neq g(r_i)$, where for $g \in \Pi \setminus \Pi^{\mathsf{ls}}$ we actually have $f(r_i) \neq 0 = g(r_i)$. The claim follows. ■

Theorem 7.6 follows from Claims 7.6.1, and 7.6.2. ■

A detour: A generalization of the argument underlying the proof of Claim 7.6.2. Let us consider an arbitrary property Π of functions over $[n]$, where in Claim 7.6.2 $[n] \equiv \{0,1\}^\ell$. For every two functions f and g over $[n]$, let $\Delta(f,g) \stackrel{\rm def}{=} \{x \in [n] : f(x) \neq g(x)\}$. As in the proof of Claim 7.6.2, for a random sample of s points that are labeled by a function f that is ϵ-far from Π, we wish to lower bound the probability that f disagrees with each $g \in \Pi$ on some point in that sample. The issue is improving over the obvious lower bound of $1 - |\Pi| \cdot (1-\epsilon)^s$. The argument establishing Claim 7.6.2 made implicit use of a suitable notion of a cover of such f's (with respect to Π). For $B : \mathbb{N} \times [0,1] \to \mathbb{N}$ and $\rho : [0,1] \to [0,1]$, we say that a collection $\mathcal{S}_\Pi(f)$ of subsets (of $[n]$) is a (B,ρ)-cover of f (w.r.t Π) if the following two conditions hold:

1. For every $g \in \Pi$ there exists $S \in \mathcal{S}_\Pi(f)$ such that $S \subseteq \Delta(f,g)$.
2. If f is ϵ-far from Π, then $|\mathcal{S}_\Pi(f)| \leq B(n,\epsilon)$ and for every $S \in \mathcal{S}_\Pi(f)$ it holds that $|S| \geq \rho(\epsilon) \cdot n$.

(In the proof of Claim 7.6.2, we implicitly referred to the collection $\mathcal{S}_\Pi(f)$ that contained all $\Delta(f,g)$ for $g \in \Pi^{\sf 1s}$ and all $\{x : f(x) \neq 0 = g(x)\} \subseteq \Delta(f,g)$ for $g \in \Pi^{\sf eq}$.) We stress that the collection $\mathcal{S}_\Pi(f)$ is *not* constructed by the tester, it is merely used in its analysis, which asserts that a sample of $s(n,\epsilon) = O(\rho(\epsilon)^{-1} \log B(n,\epsilon))$ random points suffices provided that $\mathcal{S}_\Pi(f)$ is a (B,ρ)-cover of Π w.r.t f. To see that this is the case, consider any function f (over the n-element domain) that is ϵ-far from Π. Then, by Condition 2, with probability at least $1 - (1-\rho(\epsilon))^{s(n,\epsilon)} \geq 2/3$, a sample of $s(n,\epsilon)$ random points hits each subset in the collection $\mathcal{S}_\Pi(f)$. In this case, by Condition 1, for every $g \in \Pi$ this sample hits $\Delta(f,g)$.

Back to the main thread. Turning back to the gap between distribution-free sample-complexity and one-sided error sample complexity (under the uniform distribution), we note than an ever larger gap exists. Specifically, distribution-free sample-based testing may be extremely hard (i.e., require a linear number of queries), while for the same property one-sided error testing (under the uniform distribution) may be extremely easy (or even trivial). The next theorem established Item 5 in Theorem 1.3.

Theorem 7.7 (distribution-free sample-based testing can be extremely harder than one-sided error sample-based testing under the uniform distribution): *There exists a property* Π *such that*

1. *Any distribution-free tester for* Π *has query complexity* $q(n,\Omega(1)) = \Omega(n)$. *Indeed, this lower bound holds also for two-sided error testers that make queries in addition to obtaining samples from the distribution.*
2. *There exists a sample-based one-sided error tester for* Π (under the uniform distribution) *having sample complexity* $s(n,\epsilon) = \tilde{O}(1/\epsilon)$.

In fact, we may select Π *such that any function is* $1/n$*-close to* Π_n.

Note that in the latter case, testing Π_n has sample complexity $\widetilde{O}(1/\epsilon)$, where the latter expression smoothens the actual complexity that is $s(n,\epsilon) = 0$ if $\epsilon > 1/n$ and $s(n,\epsilon) = O(n \log n)$ otherwise.

Proof: Our construction of Π_n is a disjoint union of an extremely easy-to-test property, denoted $\Pi^{\rm E}_{n-1}$, and an extremely hard-to-test property, denoted $\Pi^{\rm H}_{n-1}$. Specifically, $\Pi^{\rm E}_{n-1}$ may be trivial (i.e., contains all functions from $[n-1]$ to $\{0,1\}$), whereas $\Pi^{\rm H}_{n-1}$ may be a property that is extremely hard to test (i.e., requires a linear number of queries even for two-sided error testing under the uniform distribution, cf., e.g., [14, Sec. 4.1]).[39] The disjoint union of these two properties is captured by Π_n such that $f:\{0,1,\ldots,n-1\}$ is in Π_n if either $f(0)=0$ and $f'\in\Pi^{\rm E}_{n-1}$ or $f(0)=1$ and $f'\in\Pi^{\rm H}_{n-1}$, where f' is the restriction of f to $[n-1]$ (i.e., $f'(i)=f(i)$ for $i\in[n-1]$).

Indeed, testing Π_n is almost trivial *under the uniform distribution*, since in this case any $f:[n]\to\{0,1\}$ is $1/n$-close to Π_n (by merely resetting $f(0)=0$). Thus, for $\epsilon>1/n$, the tester may just accept (without looking at anything else), whereas for $\epsilon\leq 1/n$ it may just reconstruct the function based on $O(n\log n)$ labeled samples (and rule accordingly). Hence:

Claim 7.7.1 *The one-sided error sample complexity of* Π (under the uniform distribution) *is* $s(n,\epsilon)=\tilde{O}(1/\epsilon)$.

The forgoing phenomenon (of each function being $1/n$-close to Π_n) *does not necessarily hold with respect to an arbitrary distribution*: Consider, for example, a distribution that assign probability mass of $1/2$ to $i=0$. In this case, if $f(0)=1$ and $\epsilon<1/2$, we cannot afford to reset $f(0)=0$ and must test the residual f' with respect to $\Pi^{\rm H}_{n-1}$. In this case (i.e., $\epsilon<1/2$), testing Π_n with respect to the distribution D_n that assigns probability $1/2$ to $i=0$ and is uniform on the rest of $[n]$, amounts to testing $\Pi^{\rm H}_{n-1}$. Formally:

Claim 7.7.2 *Let $s(n,\epsilon)$ denote the sample complexity of one-sided error testing Π_n under the distribution D_n, and $s'(n-1,\epsilon)$ denote the sample complexity of one-sided error testing $\Pi^{\rm H}_{n-1}$ under the uniform distribution. Then, $s'(n-1,\epsilon)\leq s(n,\epsilon/2)$. Ditto with respect to the query complexity of testers that are allowed queries.*[40]

Indeed, the sample-based tester for $\Pi^{\rm H}_{n-1}$ invokes the sample-based tester for Π_n, while providing it with an adequately distributed sample. Specifically, for each $j\in[t]$, with probability $1/2$ it places $(0,1)$ as the $j^{\rm th}$ labeled sample of the invoked tester, and otherwise it just uses the $j^{\rm th}$ labeled sample that it got. Hence, given a uniformly distributed sample labeled by f', the new tester creates a sample labeled by the corresponding f (i.e., $f(0)=1$ and $f(i)=f'(i)$ for every $i\in[n-1]$), where this sample is distributed according to D_n. A similar transformation holds for testers that make queries. Using the hardness of $\Pi^{\rm H}_{n-1}$, the theorem follows. ■

Acknowledgements

We wish to thank a few anonymous readers for their useful comments.

References

[1] N. Alon, M. Krivelevich, I. Newman, and M. Szegedy. Regular languages are testable with a constant number of queries. *SIAM Journal on Computing*, pages 1842–1862, 2001.

[2] N. Alon and M. Krivelevich. Testing k-Colorability. *SIAM Journal on Disc. Math.*, Vol. 15 (2), pages 211-227, 2002.

[3] M. Balcan, E. Blais, A. Blum, and L. Yang. Active Property Testing. In *53rd FOCS*, pages 21–30, 2012.

[39] More generally, we might have use any easy to test $\Pi^{\rm E}_{n-1}$, provided $\Pi^{\rm H}_{n-1}\subset\Pi^{\rm E}_{n-1}$. In this case, the main claims would have hold, although it would not necessarily be the case that any function is $1/n$-close to Π_n.

[40] Note that since we deal with a fixed distribution in each of the two problems, queries subsume samples.

[4] T. Batu, L. Fortnow, R. Rubinfeld, W.D. Smith and P. White. Testing that Distributions are Close. In *Proceedings of 41st FOCS*, pages 259–269, 2000.

[5] M. Blum, M. Luby and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *JCSS*, Vol. 47, No. 3, pages 549–595, 1993. Extended abstract in *22nd STOC*, 1990.

[6] L. Devroye. The Equivalence of Weak, Strong and Complete Convergence in L1 for Kernel Density Estimates. *Annals of Statistics*, Vol. 11 (3), pages 896–904, 1983.

[7] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky. Improved Testing Algorithms for Monotonicity. Proceedings of *Random99*, Springer, LNCS 1671, pages 97–108.

[8] E. Fischer, Y. Goldhirsh, and O. Lachish. Some properties are not even partially testable. ECCC TR13-082, 2013.

[9] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. In *34th STOC*, pages 474–483, 2002.

[10] D. Glasner and R.A. Servedio. Distribution-free testing lower bounds for basic Boolean functions. *Theory of Computing*, Vo. 5 (1), pages 191–216, 2009.

[11] O. Goldreich, editor. *Property Testing – Current Research and Surveys.* Lecture Notes in Computer Science, Vol. 6390, Springer, 2010.

[12] O. Goldreich. Introduction to Testing Graph Properties. In [11].

[13] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samorodnitsky. Testing Monotonicity. *Combinatorica*, Vol. 20 (3), pages 301–337, 2000.

[14] O. Goldreich, S. Goldwasser, and D. Ron. Property Testing and its Connection to Learning and Approximation. *Journal of the ACM*, pages 653–750, July 1998. Extended abstract in *37th FOCS*, 1996.

[15] O. Goldreich, S. Goldwasser, and A. Nussboim. On the Implementation of Huge Random Objects. *SICOMP*, Vol. 39, No. 7, pages 2761–2822, 2010.

[16] O. Goldreich and T. Kaufman. Proximity Oblivious Testing and the Role of Invariances. In proceedings of *15th RANDOM*, LNCS 6845, Springer, pages 579–592, 2011.

[17] O. Goldreich and D. Ron. Property Testing in Bounded Degree Graphs. *Algorithmica*, pages 302–343, 2002. Extended abstract in *29th STOC*, 1997.

[18] O. Goldreich and D. Ron. On Proximity Oblivious Testing. *SIAM Journal on Computing*, Vol. 40, No. 2, pages 534–566, 2011. Extended abstract in *41st STOC*, 2009.

[19] O. Goldreich and I. Shinkar. Two-Sided Error Proximity Oblivious Testing. In proceedings of *16th RANDOM*, LNCS 7408, Springer, pages 565–578, 2012.

[20] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, Vol. 23 (1), pages 23–57, August 2003. Extended abstract in *42nd FOCS*, 2001.

[21] T. Gur and R. Rothblum. Non-Interactive Proofs of Proximity. ECCC TR13-078, May 2013.

[22] J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proc. 32nd ACM Symposium on the Theory of Computing*, pages 80–86, 2000.

[23] T. Kaufman and M. Sudan. Algebraic Property Testing: The Role of Invariances. In *40th STOC*, pages 403–412, 2008.

[24] M. Kearns and D. Ron. Testing problems with sub-learning sample complexity. *JCSS*, Vol. 61 (3), pages 428–456, 2000.

[25] M.J. Kearns and U.V. Vazirani. *An introduction to Computational Learning Theory.* MIT Press, 1994.

[26] D.E. Knuth. *The Art of Computer Programming*, Vol. 2 (*Seminumerical Algorithms*). Addison-Wesley Publishing Company, Inc., 1969 (first edition) and 1981 (second edition).

[27] M. Parnas, D. Ron, and A. Samorodnitsky. Testing Basic Boolean Formulae. *SIAM Journal on Disc. Math. and Alg.*, Vol. 16 (1), pages 20–46, 2002.

[28] S. Raskhodnikova and A. Smith. A note on adaptivity in testing properties of bounded-degree graphs. *ECCC*, TR06-089, 2006.

[29] D. Ron. Property Testing: A Learning Theory Perspective. *Foundations and Trends in Machine Learning*, Vol. 1 (3), pages 307–402, 2008.

[30] D. Ron. Algorithmic and Analysis Techniques in Property Testing. *Foundations and Trends in TCS*, Vol. 5 (2), pages 73–205, 2009.

[31] R. Rubinfeld and M. Sudan. Robust Characterization of Polynomials with Applications to Program Testing. *SIAM Journal on Computing*, 25(2), pages 252–271, 1996.

[32] M. Sudan. Invariance in Property Testing. In [11], pages 211-227.

[33] L.G. Valiant. A Theory of the Learnable. *Communications of the ACM*, Vol. 27/11, pages 1134–1142, 1984.

[34] G. Valiant and P. Valiant. Estimating the unseen: an $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. In *43rd ACM Symposium on the Theory of Computing*, pages 685–694, 2011. See *ECCC* TR10-180 for the algorithm, and TR10-179 for the lower bound.

Appendix: In Passing

A.1 A relaxed notion of POT

The following definition is a hybrid of the standard definition of a tester and a POT: Specifically, the tester gets the proximity parameter ϵ and its behavior and performance may depend on ϵ, but its query complexity is fixed.

Definition A.1 (relaxed POT): *Let* Π *and* ϱ *be as in Definition 2.2. A* relaxed POT with detection probability ϱ for Π *is a probabilistic oracle machine* T *that makes a constant number of queries and satisfies the following two conditions:*

1. *For every* $n \in \mathbb{N}$, $\epsilon > 0$, *and* $f \in \Pi_n$, *it holds that* $\Pr[T^f(n,\epsilon)=1] \geq c$.
2. *For every* $n \in \mathbb{N}$, $\epsilon > 0$, *and* $f : D_n \to R_n$ *that is* ϵ*-far from* Π_n, *it holds that* $\Pr[T^f(n,\epsilon)=1] \leq c - \varrho(\epsilon)$.

Again, the constant c *is called the* threshold probability, *and the tester is said to have* one-sided error *if* $c = 1$.

Note that detection probability of a relaxed POT is not required to increase with the distance of the tested object to the property. (Instead, the detection probability is determined by the proximity parameter ϵ, which is given to the tester, and the detection guarantee applies only to functions that are ϵ-far from Π.) Nevertheless, if $c = 1$, then such a tester can be used to derive a POT (in which the detection probability does increase with the distance of the tested object to the property).

Proposition A.2 *If* Π *has a one-sided error* relaxed *POT with detection probability* ϱ, *then* Π *has a one-sided error POT with detection probability* ϱ' *such that* $\varrho'(\delta) = O(\log(1/\delta))^2 \cdot \varrho(\delta/2)$. *Furthermore, the resulting POT preserves the fairness of the relaxed POT, where fairness is as defined in the beginning of Section 3.*

Proof: Let T be a (one-sided error) relaxed POT with detection probability ϱ for Π, and consider the following oracle machine: On input n and access to the oracle f, the machine selects $i \in [\lceil \log_2 n \rceil]$ with probability $1/(i+1)^2$, invokes $T^f(n, 2^{-i})$, and outputs whatever T outputs.[41] Note that if $f \in \Pi_n$, then the new tester always outputs 1 (regardless of its choice of i). On the other hand, if $f : D_n \to R_n$ is not in Π_n, then $\delta_\Pi(f) \geq 1/n$ and $i = \lceil \log_2(1/\delta_\Pi(f)) \rceil$ is selected with probability at least $(1 + \lceil \log_2(1/\delta_\Pi(f)) \rceil)^{-2}$. In this case, T outputs 1 with probability at most $1 - \varrho(2^{-i}) \leq 1 - \varrho(\delta_\Pi(f)/2)$, where the inequality is due to the monotonicity of ϱ. ■

Remark A.3 *The proof of Proposition A.2 does not extend to the case that the threshold probability is smaller than 1* (i.e., $c < 1$). *The problem is that in such a case, not all functions in* Π *are guaranteed to be accepted with probability* exactly c. *The proof does extend to two-sided error POTs that satisfy the latter condition, and such POTs do exist also for properties having no one-sided error POTs* (cf., e.g., [19, Thm. 2.3] and [19, Thm. 3.1]).

[41] With probability $1 - \sum_{i \in [\log_2 n]} (i+1)^{-2} > 0$, the machine just outputs 1 (or alternatively it selects $i \in [\log_2 n]$ arbitrarily and behaves accordingly).

A.2 Relating various notions of sampling

In this section, we relate various notions of sampling. In all cases we are given a number of samples from a domain D_n as well as the size of this domain $n = |D_n|$. Definition 2.3 refers to the natural case in which we are given a fixed number of samples, where each sample point is uniformly and independently distributed in D_n. Thus, the given sample may contain repetitions, and in fact it will contain repetitions with a fair (or very high) probability if its size exceeds $\sqrt{|D_n|}$.

It is often more convenient to consider a sample of a fixed size that is drawn without repetitions; that is, a set of a fixed size, denoted m, that is selected uniformly among all m-subsets of D_n Likewise, we sometime consider a sample of D_n generated by picking each element with some probability p (e.g., $p = m/n$) independently of all other choices. In this section we relate these three notions, showing that each one can emulate the others, up-to small deviations that are typically insignificant. We first relate sampling with and without repetitions.

Construction A.4 (sampling with repetitions implies sampling without repetitions): *Suppose that $m \leq n$ and that we are given m uniformly and independently distributed elements of D_n, denoted $r_1, \ldots, r_m$. Then, with probability at least $1 - \exp(-m)$, this sample contains at least $m/2$ distinct elements. In this case, we let $t \geq m$ be the smallest integer such that $|\{i \in [t] : r_i\}| = m/2$, and output the $m/2$-set $\{i \in [t] : r_i\}$* (in random order).

Proposition A.5 (sampling without repetitions implies sampling with repetitions): *Suppose that $m \leq n$ and that we are given a set of m element selected uniformly among all m-subsets of D_n. Let $(e_1, \ldots, e_m)$ be a random ordering of the elements of this set. Consider a selection of $(r_1, \ldots, r_m)$ such that, for every $i \in [m]$, with probability $(n - i + 1)/n$ we select $r_i = e_i$ and otherwise we select r_i uniformly in $\{e_1, \ldots, e_{i-1}\}$. Then, $(r_1, \ldots, r_m)$ is a sequence of m uniformly and independently distributed elements of D_n.*

Proof: We shall show that for every $i \in [m]$ and for every $v_1, \ldots, v_i \in D_n$, it holds that

$$\Pr\Big[r_i = v_i \Big| (r_1, \ldots, r_{i-1}) = (v_1, \ldots, v_{i-1})\Big] \; = \; 1/n.$$

In fact, for each sequence of $i - 1$ distinct $u_1, \ldots, u_{i-1} \in D_n$, we consider the probability

$$\Pr\Big[r_i = v_i \Big| (r_1, \ldots, r_{i-1}) = (v_1, \ldots, v_{i-1}) \;\&\; (e_1, \ldots, e_{i-1}) = (u_1, \ldots, u_{i-1})\Big].$$

Now, if $v_i \in \{u_1, \ldots, u_{i-1}\}$, then $r_i = v_i$ with probability $(1 - \frac{n-i+1}{n}) \cdot \frac{1}{i-1} = \frac{1}{n}$. On the other hand, if $v_i \notin \{u_1, \ldots, u_{i-1}\}$, then $r_i = v_i$ with probability $\frac{n-i+1}{n} \cdot \frac{1}{n-(i-1)} = \frac{1}{n}$ (where $\frac{1}{n-(i-1)}$ represents the probability that $e_i = v_i$ when e_i is selected uniformly in $D_n \setminus \{u_1, \ldots, u_{i-1}\}$). ■

Two versions of sampling without repetitions. In the foregoing, "sampling without repetitions" referred to selecting a random subset of a fixed size within D_n. An alternative notion of sampling without repetitions arises when each element is selected with some fixed probability, independently of all other choices.

Construction A.6 (individual selection implies sampling a set of fixed size): *Suppose that each point is selected to be included in the sample with probability $p \in (0, 1]$ independently of all other choices. Then, with probability at least $1 - \exp(-pn)$, at least $m \stackrel{\rm def}{=} pn/2$ points are selected. In this case, we output a random m-subset of the set of selected points* (in random order).

Construction A.7 (sampling a set of fixed size implies individual selection): *Suppose that $m \leq n$ and that we are given a set S of m element selected uniformly among all m-subsets of D_n. Setting $p = m/2n$, let $B_k(n,p)$ denote the probability of obtaining k successes when making n independent experiments with success probability p each. Consider selecting i with probability $B_i(n,p)$, and outputting a random i-subset of S, if $i \leq |S|$, and outputting nothing otherwise. Then, the output distribution is $\exp(-m)$-close to the distribution of a sample that is selected by including each point in the sample with probability p independently of all other choices.*

Sampling according to the binomial distribution $B(n,p)$ (i.e., selecting i with probability $B_i(n,p)$) can be approximate up to ϵ in $\text{poly}(\log(n/\epsilon))$-time; cf., Knuth [26, Sec. 3.4.1] (and [15, Apdx. A.1]).

A.3 On the sample complexity of testing monomials

The following result is implicit in [3].

Proposition A.8 *Let $\Pi = \bigcup_{\ell \in \mathbb{N}} \Pi_{2^\ell}$, where Π_{2^ℓ} is the set of all ℓ-variable Boolean functions that are monomials. Then, any sample-based tester for* Π (under the uniform distribution) *has sample complexity $s(2^\ell, \epsilon) = \Omega(\epsilon^{-1} \log \ell)$, provided that $\epsilon > 2^{-(\ell - \ell^{0.01})}$. Indeed, this lower bound holds also for two-sided error testers.*

Proof: Let us consider first the case $\epsilon = 1/3$. In this case we consider the task of distinguishing a random Boolean function $f : \{0,1\}^\ell \to \{0,1\}$ from a uniformly distributed dictatorship function (i.e., $d_i(x) = x_i$ for $i \in [\ell]$) based on $s = 0.5 \log_2 \ell$ labeled samples, denoted $r_1, \ldots, r_s$. (This is exactly the task analyzed in [3, Thm. VI.8].)[42] The labels assigned by a random function are uniformly and independently distributed in $\{0,1\}$, since the probability of a repetition (i.e., $|\{r_1, \ldots, r_s\}| < s$) in such a small sample is negligible. We now analyze the distribution of s samples that are labeled by d_i for a uniformly distributed $i \in [\ell]$.

We consider an iterative process of assigning these labels, while initializing $I = [\ell]$. Specifically, for $j = 1, 2 \ldots, s$, the label of the j^{th} sample is determined by selecting i_j uniformly in I, labeling the j^{th} sample according to d_{i_j}, and resetting I to equal $\{i \in I : d_i(r_j) = d_{i_j}(r_j)\}$. Denoting the set I after the j^{th} iteration by I_j, note that $I_j = \{i \in I_{j-1} : r_{j,i} = r_{j,i_j}\}$, where $r_{j,i}$ is the i^{th} bit of r_j. Hence, for every $j \in [s]$, with probability at least $1 - \exp(-\Omega(|I_{j-1}|^{1/3})$, it holds that $|I_j| = (0.5 \pm |I_{j-1}|^{-1/3}) \cdot |I_{j-1}|$. It follows that the deviation of these labels from the uniform distribution is upper-bounded by $\exp(-\omega(\ell/2^s)^{1/3}) = \exp(-\omega(\ell^{1/6}))$, and $s(2^\ell, 1/3) \geq 0.5 \log_2 \ell$ follows, since (w.v.h.p.) a random function is 1/3-far from Π.

For the general case (of $\epsilon < 1/6$), we designate $t \overset{\text{def}}{=} \log_2(3/\epsilon)$ of the variables and apply the foregoing construction to the remaining $\ell - t$ variables. For each function $f' : \{0,1\}^{\ell - t} \to \{0,1\}$ (as considered above), we consider the function $f(x_1, \ldots, x_\ell) = \bigwedge_{i \in [t]} x_i \wedge f'(x_{t+1}, \ldots, x_\ell)$. Note that if f' is a dictator function then f is a monomial, whereas if f' is 1/3-far from Π then f is ϵ-far from Π. We thus obtain a lower bound of $s(2^\ell, \epsilon) = \Omega(\epsilon^{-1} \cdot s(2^{\ell - t}, 1/3))$, and the claim follows. ■

[42] In fact, the furthermore clause of [3, Thm. VI.8] refers explicitly to distinguishing dictatorship functions from random functions. Still, for sake of self-containment, we reproduce the argument here.