Nearly Tight Bounds for Testing Function Isomorphism*

Noga Alon † [1], Eric Blais [2], Sourav Chakraborty ‡ [3],
David García-Soriano ‡ [4], and Arie Matsliah § [5]

[1]Schools of Mathematics and Computer Science, Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel. Email: nogaa@tau.ac.il.
[2]School of Computer Science, Carnegie Mellon University, Pittsburgh 15213, USA. Email: eblais@cs.cmu.edu.
[3]Chennai Mathematical Institute, India. Email: sourav@cmi.ac.in
[4]CWI Amsterdam, The Netherlands. Email: david@cwi.nl
[5]IBM Research and Technion, Haifa, Israel. Email: arie.matsliah@gmail.com

Abstract

We study the problem of testing isomorphism (equivalence up to relabeling of the input variables) between Boolean functions. We prove that:

- For most functions $f : \{0,1\}^n \to \{0,1\}$, the query complexity of testing isomorphism to f is $\Omega(n)$. Moreover, the query complexity of testing isomorphism to most k-juntas $f : \{0,1\}^n \to \{0,1\}$ is $\Omega(k)$.
- Isomorphism to any k-junta $f : \{0,1\}^n \to \{0,1\}$ can be tested with $O(k \log k)$ queries.
- For some k-juntas $f : \{0,1\}^n \to \{0,1\}$, testing isomorphism to f with **one-sided error** requires $\Omega(k \log(n/k))$ queries. In particular, testing if $f : \{0,1\}^n \to \{0,1\}$ is a k-parity with one-sided error requires $\Omega(k \log(n/k))$ queries.
- The query complexity of testing isomorphism between two unknown functions $f, g : \{0,1\}^n \to \{0,1\}$ (both given by oracles) is $\widetilde{\Theta}(2^{n/2})$.

These bounds are tight up to logarithmic factors, and they significantly strengthen the bounds proved by Fischer et al. (FOCS 2002) and Blais and O'Donnell (CCC 2010).

We also obtain results closely related to isomorphism testing, answering questions posed by Diakonikolas et al. (FOCS 2007):

- Testing whether a function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a circuit of size $\leq s$ requires $s^{\Omega(1)}$ queries.
- Testing if the Fourier degree of a function $f : \{0,1\}^n \to \{0,1\}$ is $\leq d$ requires $\Omega(d)$ queries.

All of our lower bounds apply to general (adaptive) testers.

*This article is a joint full version of [AB10] and [CGM11b].

†Research supported in part by an ERC Advanced grant, by a USA-Israeli BSF grant and by the Hermann Minkowski Minerva Center for Geometry at Tel Aviv University.

‡Research performed while at CWI in Amsterdam and supported by the Netherlands Organization for Scientific Research through a VICI grant.

§Research supported in part by an ERC-2007-StG grant number 202405.

Contents

1 Introduction

1.1 Background

The field of property testing, originally introduced by Rubinfeld and Sudan [RS96], has been extremely active over the last few years – see, e.g., the recent surveys [Ron08, Ron10, RS11].

In this paper we focus on testing properties of Boolean functions. Despite the progress in the study of the query complexity of many properties of Boolean functions (e.g., monotonicity [DGL$^+$99, FLN$^+$02, GGL$^+$00], juntas [FKR$^+$04, CG04], having concise representations [DLM$^+$07], halfspaces [MORS09a, MORS09b]), our overall understanding of the testability of Boolean function properties still lags behind our understanding of the testability of graph properties, whose study was initiated by Goldreich, Goldwasser, and Ron [GGR98].

A notable example that illustrates the gap between our understanding of graph and Boolean function properties is *isomorphism*. Two graphs are isomorphic if they are identical up to relabeling of the vertices, while two Boolean functions are isomorphic if they are identical up to relabeling of the input variables. There are three main variants to the isomorphism testing problem. (In the following list, an "object" refers to either a graph or a Boolean function.)

1. **Testing isomorphism to a given object $\mathcal{O}$.** The query complexity required to test isomorphism in this variant depends on the object $\mathcal{O}$; the goal for this problem is to characterize the query complexity for *every* graph or Boolean function.

2. **Testing isomorphism to the hardest known object.** A less fine-grained variant of the first problem asks to determine the maximum query complexity of testing isomorphism to $\mathcal{O}$ over objects of a given size.

3. **Testing isomorphism of two unknown objects.** In this variant, the testing algorithm has query access to two unknown objects $\mathcal{O}_1$ and $\mathcal{O}_2$ and must distinguish between the cases where they are isomorphic to each other or far from isomorphic to each other.

Answering these questions, as suggested by [FKR$^+$04] and [BO10], is an important step in the research program of characterizing testable properties of Boolean functions.

The problem of testing graph isomorphism was first raised by Alon, Fischer, Krivelevich, and Szegedy [AFKS00] (see also [Fis01]), who used a lower bound on testing isomorphism of two unknown graphs to give an example of a non-testable first-order graph property of a certain type. Fischer [Fis05] studied the problem of testing isomorphism to a given graph G and characterized the class of graphs to which isomorphism can be tested with constant number of queries. Tight asymptotic bounds on the (worst-case) query complexity of the problem of testing isomorphism to a known graph and testing isomorphism of two unknown graphs were then obtained by Fischer and Matsliah [FM08]. As a result, the graph isomorphism testing problem is well understood[1]. Additionally, Babai and Chakraborty [BC10] proved query-complexity lower bounds for (generalizations of) the problem of testing isomorphism between two uniform hypergraphs.

[1] To summarize,

- Graphs to which isomorphism can be tested with constant number of queries are those that can be approximated by a simple algebra of constantly many cliques [Fis05];
- The worst-case query complexity of testing isomorphism to a given graph on n nodes is $\widetilde{\Theta}(\sqrt{n})$ [FM08].

The picture is much less complete in the setting of Boolean functions. The first question above is particularly interesting because testing many function properties, like those of being a dictatorship, a k-monomial, a k-parity and more, are equivalent to testing isomorphism to some fixed function f. More general properties can often be reduced to testing isomorphism to several functions (as a simple example, notice that testing whether g depends on a single variable can be done by first testing if g is isomorphic to $f(x) \equiv x_1$, then testing if g is isomorphic to $f(x) \equiv 1 - x_1$, and accepting if one of the tests accepts). The "Testing by Implicit Learning" approach of Diakonikolas et al. [DLM$^+$07] can also be viewed as a clever reduction from the task of testing a wide class of properties to testing function isomorphism against a number of functions. We elaborate more on [DLM$^+$07] and how our work relates to it in the following section.

There are several classes of functions for which testing isomorphism is trivial. For instance, if f is symmetric (invariant under permutations of variables), then f-isomorphism can be tested with constant number of queries[2]. More interesting functions are also known to have testers with constant query complexity. Specifically, the fact that isomorphism to dictatorship functions and k-monomials can be tested with $O(1)$ queries follows from the work of Parnas et al. [PRS02].

The question of testing isomorphism against a known function f was first formulated explicitly by Fischer, Kindler, Ron, Safra, and Samorodnitsky [FKR$^+$04]. They gave a general upper bound on the problem showing that for every function f that depends on k variables (that is, for every k-junta), the problem of testing isomorphism to f is solvable with $\mathrm{poly}(k/\epsilon)$ queries. Conversely, they showed that when f is a parity function on $k < o(\sqrt{n})$ variables, testing isomorphism to f requires $\Omega(\log k)$ queries[3]. No other progress was made on the problem of testing isomorphism on Boolean functions until recently, when Blais and O'Donnell [BO10] showed that for any $k \leq n - \omega(1)$ and function f that "strongly" depends on k variables[4], testing isomorphism to f requires $\Omega(\log k)$ non-adaptive queries, which implies a general lower bound of $\Omega(\log \log k)$. They also proved that there is a k-junta (namely, a majority on k variables) testing isomorphism to which requires $\Omega(k^{1/12})$ queries non-adaptively, and therefore $\Omega(\log k)$ queries in general.

Taken together, the results in [FKR$^+$04, BO10] give only an incomplete solution to the problem of testing isomorphism to a given Boolean function and provide only weak bounds on the other two versions of the isomorphism testing problem.

In this paper we settle the last two questions up to logarithmic factors, and report some progress towards answering the first one.

1.2 Recent developments

Concurrently to the preliminary versions of this work ([AB10] and [CGM11b]), Goldreich [Gol10] has published a proof of $\Omega(\sqrt{n})$ lower bound on the number of queries required for testing isomorphism to a parity on $n/2$ variables[5]. This bound was subsequently improved to $\Omega(n)$ (and more generally to $\Omega(k)$ for testing isomorphism to k-parities) by Blais, Brody and Matulef [BBM11].

[2]Since all permutations of a symmetric f are the same, the problem reduces to testing (strict) equivalence to a given function.

[3]This was shown via an $\Omega(\sqrt{k})$ lower bound for non-adaptive testers.

[4]This also means that for some $c = O(1)$ f is far from all $k - c$ juntas.

[5]A higher lower bound of $\Omega(n)$ queries was proved for non-adaptive testing.

2 Our results

2.1 Lower bounds for testing function isomorphism

It is easy to show that isomorphism to any $f : \{0,1\}^n \to \{0,1\}$ can be ϵ-tested with $O(\frac{n \log n}{\epsilon})$ queries, using Occam's razor. For constant ϵ, which is the primary focus here, this is $\widetilde{O}(n)$; our first result is a nearly matching lower bound of $\Omega(n)$ that applies for *almost all* functions f. In fact we provide a lower bound of $\Omega(k)$ on the query complexity of testing (adaptively, with two-sided error) isomorphism to k-juntas.

Theorem 2.1 *Fix a constant $0 < \epsilon < \frac{1}{4}$ and let $k \leq n$. For a $1 - o(1)$ fraction of the k-juntas $f : \{0,1\}^n \to \{0,1\}$, any algorithm for ϵ-testing isomorphism to f must make $\Omega(k)$ queries.*

We present the proof of Theorem 2.1 in Section 7.1, after proving the special case for $k = n$ in Section 6.2. The proof is non-constructive, but we also show that the hardest functions to test isomorphism to may have relatively simple descriptions, such as belonging to nonuniform $\mathcal{NC}$ or being a polynomial over $\mathbb{F}_2$ of degree logarithmic in k. As a corollary we obtain the following lower bounds, resolving open problems from [DLM+07]:

Corollary 2.2 *Let $\epsilon < 1/4$. There is a constant $c \geq 1$ such that for all $s \leq n^c$, testing size-s Boolean circuits requires $\Omega(s^{1/c})$ queries.*

Corollary 2.3 *For any $0 < \epsilon < 1/4$ and $d < n/2 - 2\sqrt{n}$, the query complexity of ϵ-testing whether a function $f : \{0,1\}^n \to \{0,1\}$ is a polynomial $p \in \mathbb{F}[x_1, \ldots, x_n]$ of total degree at most d over any field $\mathbb{F}$ is $\Omega(d)$.*[6]

If $\mathbb{F} = \mathbb{Q}$ then it means that the query complexity for testing if the Fourier degree of a Boolean function is at most d is $\Omega(d)$.

The proofs of the corollaries appear in Section 7.

Remark 2.1 *While the lower bound of Theorem 2.1 is near best possible and applies to most functions, the proof has the disadvantage of not being constructive. This is not the case in the aforementioned lower bounds from [Gol10] and [BBM11], which apply to testing isomorphism to linear functions.*

2.2 Upper bounds for testing function isomorphism

Our second result (Theorem 2.4) is a nearly matching upper bound for testing isomorphism to any fixed k-junta (with constant ϵ):

Theorem 2.4 *Isomorphism to any k-junta can be ϵ-tested with $O(\frac{k \log k}{\epsilon} + \frac{1}{\epsilon^2})$ queries.*

[6] These bounds apply to the degree over *any* field. Better bounds are known for small enough finite fields; c.f. [AKK+03, JPRZ04, KR04].

This generalizes the aforementioned $O(n \log n)$ upper bound and improves upon the $\widetilde{O}(k^4)$ upper bound of [FKR$^+$04] . One consequence of our techniques, which is of independent interest, is the following (see Proposition 9.2 for a formal statement):

Let $\epsilon > 0$ and suppose we are given oracle access to a k-junta $g : \{0,1\}^n \to \{0,1\}$. Then, after a preprocessing step that makes $O(k \log k/\epsilon)$ queries to g, we can draw uniformly random samples $(x, a) \in \{0,1\}^k \times \{0,1\}$ labelled by $\mathsf{core}(g) : \{0,1\}^k \to \{0,1\}$ – the function of k variables lying at the "core" of g, such that for each sample (x, a), $\mathsf{core}(g)(x) = a$ with probability at least $1 - \epsilon$. Furthermore, obtaining each sample requires making only one query to g.

Generating such samples is one of the main ingredients in the general framework of [DLM$^+$07]; while the procedure therein makes $\Omega(k)$ queries to g for obtaining each sample (executing k independence tests of Fischer et al. [FKR$^+$04]), our procedure requires only *one* query to g per sample.

Remark 2.2 *In subsequent work [CGM11a], a variation of this sampler is used to significantly improve the query-complexity of the testers from [DLM$^+$07] for various Boolean function classes.*

2.3 Testing function isomorphism with one-sided error

Our third result concerns testing function isomorphism with one-sided error. The fact that the one-sided error case is strictly harder than the two-sided error case was established in [FKR$^+$04]. In particular, they showed the impossibility of testing isomorphism to 2-juntas with one-sided error using a number of queries independent of n (their lower bound is $\Omega(\log \log n)$, which follows from an $\Omega(\log n)$ lower bound on non-adaptive testers). In this paper we obtain nearly tight lower bounds for the problem:

Theorem 2.5

- *The query complexity of testing isomorphism to any k-junta with one-sided error is $O(k \log n)$.*
- *For every $2 \leq k \leq n$ and constant $0 < \epsilon \leq \frac{1}{2}$, there is a k-junta $f : \{0,1\}^n \to \{0,1\}$ for which testing f-isomorphism with one-sided error requires $\Omega(k \log(n/k))$ queries.*

The range of k in the theorem is optimal: when $k = 1$, as we mentioned in the introduction, testing isomorphism to any 1-junta with one-sided error can be done with a constant number of queries [PRS02].

The lower bound in Theorem 2.5 follows from the following result: for any $2 \leq k \leq n - 2$, the query complexity of testing with one-sided error whether a function is a k-parity (i.e, an XOR of *exactly* k indices of its input) is $\Theta(\log \binom{n}{k})$. This is in stark contrast to the problem of testing with one-sided error whether a function is a k-parity for *some* k, which can be done with a constant number of queries by the well-known BLR test [BLR90].

2.4 Testing isomorphism between two unknown functions

Finally, we examine the problem of testing two unknown functions for the property of being isomorphic. A simple algorithm can ϵ-test isomorphism in this setting with $\tilde{O}(2^{n/2}/\sqrt{\epsilon})$ queries. We give a lower bound establishing that no algorithm can do much better.

Testing problem	Prior bounds		This work	
	Adaptive	Non-adaptive		
Isomorphism to k-juntas	$\Omega(\log k)$ [FKR+04, BO10] $\widetilde{O}(k^4)$ [FKR+04, DLM+07]	$\Omega(\sqrt{k})$ for $k \ll n$ [FKR+04] $\Omega(k^{1/12})$ for $k \ll n$ [BO10]	$\Omega(k)$ $O(k \log k)$	(Thm. 2.1) (Thm. 2.4)
Isomorphism to k-juntas with 1-sided error	$\Omega(\log \log n)$ [FKR+04]	$\Omega(\log n)$ [FKR+04]	$\Omega(k \log (n/k))$ $O(k \log n)$	(Thm. 2.5)
Having circuits of size s	$\widetilde{\Omega}(\log s)$ [DLM+07] $\widetilde{O}(s^6)$ [DLM+07]		$s^{\Omega(1)}$	(Coro. 2.2)
Having Fourier degree $\leq d$	$\Omega(\log d)$ [DLM+07] $2^{O(d)}$ [DLM+07]	$\Omega(\sqrt{d})$ [DLM+07]	$\Omega(d)$	(Coro. 2.3)
Isomorphism between two unknown functions			$\Omega(2^{n/2}/n^{1/4})$ $O(2^{n/2}\sqrt{n \log n})$	(Thm. 2.6)

Table 1: Summary of results

Theorem 2.6 *The query complexity of testing isomorphism of two unknown functions in* $\{0,1\}^n \to \{0,1\}$ *is* $\tilde{\Theta}(2^{n/2})$ *for constant* ϵ*.*

Again, this bound holds for all testing algorithms (adaptive or non-adaptive, with 1-sided or 2-sided error).

2.5 Summary

In Table 1 we summarize our main results, and compare them to prior work. A few remarks are in order:

- Some of the lower bounds from prior work were obtained via exponentially larger lower bounds for non-adaptive testers, and some of them held only for limited values of k. The third column contains the details. Our lower bounds apply to general (adaptive, two-sided error) testers, and hold for all $k \leq n$.
- In the case of testing for being a k-parity with 1-sided error, the lower bound of $\Omega(k \log(n/k))$ (Thm. 2.5) is asymptotically tight.
- The exponent in our $s^{\Omega(1)}$ bound for testing circuit size depends on the size of the smallest circuit that can generate s^4-wise independent distributions (see details in Section 7.4). In particular, standard textbook constructions show that the exponent is at least $1/8$.

Organization of the rest of the paper. After the necessary preliminaries, we give a brief overview of the main proofs in Section 4. The proofs for one-sided-error testing are given in Section 5. In Section 6 we present the $\Omega(n)$ lower bound on the query complexity of testing isomorphism, which is then extended to the $\Omega(k)$ lower bound for k-juntas in Section 7.1. The lower bounds for testing whether a function has Fourier degree at most d or a circuit of size s are given in Section 7.4. The algorithm for testing isomorphism to k-juntas is given in Section 8. In Section 10 we prove the bounds for testing isomorphism in the setting where both functions have to be queried.

3 Preliminaries

3.1 Generalities

Throughout the paper, f and g represent Boolean functions $\{0,1\}^n \to \{0,1\}$. Tilde notation is used to hide polylogarithmic factors – for example $r(n) = \tilde{\Theta}(t(n))$ if there is a positive constant c such that $r(n) \geq \Omega(\frac{t(n)}{\log^c t(n)})$ and $r(n) \leq O(t(n)\log^c t(n))$.

Let $n, k \in \mathbb{N}$ and $x \in \{0,1\}^n$. We use the following standard notation:

- $[n] = \{1, \ldots, n\}$ and $[k, n] = \{i \in [n] : k \leq i \leq n\}$;
- $|x| = |\{i \in [n] : x_i = 1\}|$ (the *Hamming weight* of input $x \in \{0,1\}^n$);

For a set S and $k \in \mathbb{N}$, $\binom{S}{k}$ is the collection of all k-sized subsets of S and $\binom{S}{\leq k}$ is the collection of all subsets of size at most k; a similar notation is used for binomial coefficients $\binom{m}{\leq k}$.

Given a subset $I \subseteq [n]$ of cardinality k, $x\restriction_I$ denotes the k-bit binary string obtained by restricting of x to the indices in I, according to the natural order of $[n]$. We also write $f\restriction_S$ for the restriction of a function to a set S. For $y \in \{0,1\}^{|I|}$, $x_{I \leftarrow y}$ denotes the string obtained by taking x and substituting its values in I with y.

We also write

$$\{0,1\}^n_{\frac{n}{2} \pm h} \triangleq \{x \in \{0,1\}^n : \tfrac{n}{2} - h \leq |x| \leq \tfrac{n}{2} + h\}.$$

3.2 Permutations

The group of permutations $\pi : [n] \to [n]$ is denoted $\mathcal{S}_n$. For a permutation $\pi \in \mathcal{S}_n$ and $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, we write, with some abuse of notation, $\pi(x) = (x_{\pi(1)}, \ldots, x_{\pi(n)})$. The map sending $x \in \{0,1\}^n$ to $\pi(x) \in \{0,1\}^n$ is a permutation of $\{0,1\}^n$, which we denote also by π. (The corresponding permutation of $\{0,1\}^n$ can be viewed as the natural action of π^{-1} on $\{0,1\}^n$). Clearly there are $n!$ permutations of $\{0,1\}^n$ arising this way.

The function $g^\pi : \{0,1\}^n \to \{0,1\}$ represents the function defined by $g^\pi(x) = g(\pi(x))$ for every $x \in \{0,1\}^n$. Two functions f and g are *isomorphic* (in short, $f \cong g$) if there is a permutation $\pi \in \mathcal{S}_n$ such that $f = g^\pi$.

3.3 Property testing

A *property* $\mathcal{P}$ of Boolean functions is simply a subset of those functions. Given a pair $f, g : D \to \{0,1\}$ of Boolean functions defined on D, the *distance* between them is $\mathrm{dist}(f,g) \triangleq \Pr_{x \in D}[f(x) \neq g(x)]$. (Throughout this paper, $e \in S$ under the probability symbol means that an element e is chosen uniformly at random from a set S.)

The distance of a function f to $\mathcal{P}$ is the minimum distance between f and g over all $g \in \mathcal{P}$, i.e. $\mathrm{dist}(f, \mathcal{P}) = \min_{g \in \mathcal{P}} \mathrm{dist}(f,g)$. For $\epsilon \in \mathbb{R}^+$, f is ϵ-far from $\mathcal{P}$ if $\mathrm{dist}(f, \mathcal{P}) \geq \epsilon$, otherwise it is ϵ-close to $\mathcal{P}$.

A (q, ϵ)-*tester* for the property $\mathcal{P}$ is a randomized algorithm $\mathcal{T}$ that queries an unknown function f on q different inputs in $\{0,1\}^n$ and then (1) accepts f with probability at least $\frac{2}{3}$ when $f \in \mathcal{P}$, and (2) rejects f with probability at least $\frac{2}{3}$ when f is ϵ-far from $\mathcal{P}$. (If the property deals with a pair of input functions, the algorithm may query both.)

The query complexity of a tester $\mathcal{T}$ is the worst-case number of queries it makes before making a decision. $\mathcal{A}$ is *non-adaptive* if its choice of queries does not depend on the outcomes of earlier queries. A tester that always accepts functions in $\mathcal{P}$ has *1-sided error*, otherwise it has *2-sided error*. We assume without loss of generality that testers never query the same input twice. By default, in all testers (and bounds) discussed in this paper we assume adaptivity and two-sided error, unless mentioned otherwise.

The *query complexity* of a property $\mathcal{P}$ for a given $\epsilon > 0$ is the minimum value of q for which there is a (q, ϵ)-tester for $\mathcal{P}$.

Isomorphism testing

The *distance up to permutations of variables* is defined by $\mathrm{distiso}(f, g) \triangleq \min_{\pi \in \mathcal{S}_n} \mathrm{dist}(f^\pi, g)$.

Testing f-isomorphism is defined as the problem of testing the property $\mathsf{Isom}_f \triangleq \{f^\pi : \pi \in \mathcal{S}_n\}$ in the usual property testing terminology (see above). It is thus the task of distinguishing the case $f \cong g$ from the case $\mathrm{distiso}(f, g) \geq \epsilon$.

If $\mathcal{C}$ is a set of functions, then the query complexity for testing isomorphism to $\mathcal{C}$ is the maximum, taken over all $f \in \mathcal{C}$, of the query complexity for testing f-isomorphism.

3.4 Influence, Juntas, Parities

A *parity* is a linear form on $\mathbb{F}_2^n$ i.e. a function $f : \{0,1\}^n \to \{0,1\}$ given by

$$f(x) = \langle x, v \rangle \bmod 2 = \bigoplus_{i \in [n]} x_i v_i$$

for some $v \in \{0,1\}^n$. We say that f is a *k-parity* if its associated vector v has Hamming weight **exactly** k. The set of all k-parities is denoted PAR_k.

For a function $g : \{0,1\}^n \to \{0,1\}$ and a set $A \subseteq [n]$, the *influence* of A on g is defined as

$$\mathrm{Inf_g(A)} \triangleq \Pr_{\mathrm{x} \in \{0,1\}^{\mathrm{n}},\ \mathrm{y} \in \{0,1\}^{|\mathrm{A}|}} \left[\mathrm{g(x)} \neq \mathrm{g}(\mathrm{x}_{A \leftarrow y}) \right].$$

Thus $\mathrm{Inf_g(A)}$ measures the probability that the value of g changes after a random modification of the bits in A of a random input x. Note that when $|A| = 1$, this value is half that of the most common definition of influence of one variable; for consistency we stick to the previous definition instead in this case as well. For example, every variable of a k-parity ($k \geq 1$) has influence $\frac{1}{2}$.

An index (variable) $i \in [n]$ is *relevant* with respect to g if $\mathrm{Inf_g(\{i\})} \neq 0$. A *$k$-junta* is a function g that has **at most** k relevant variables; equivalently, there is $S \in \binom{[n]}{k}$ such that $\mathrm{Inf_g([n] \setminus S)} = 0$. Jun_k will denote the class of k-juntas (on n variables), and for $A \subseteq [n]$, Jun_A will denote the class of juntas all of whose relevant variables are contained in A.

3.5 A lemma for proving adaptive lower bounds

Let $\mathcal{P}$ be a property of functions mapping T to $\{0,1\}$. Define

$$\mathcal{R} \triangleq \{f : T \to \{0,1\} \mid \mathrm{dist}(f, \mathcal{P}) \geq \epsilon\}.$$

Any tester for $\mathcal{P}$ should, with high probability, accept inputs from $\mathcal{P}$ and reject inputs from $\mathcal{R}$.

We use the following lemma in various lower bound proofs for two-sided adaptive testing. It is proven implicitly in [FNS04], and a detailed proof appears in [Fis01]. Here we use a somewhat stronger version of it, but still, the original proof works as is (we reproduce it here for completeness).

Lemma 3.1 *Let $\mathcal{P}, \mathcal{R}$ be as in the preceding discussion, and let $\mathcal{F}_{\text{yes}}$ and $\mathcal{F}_{\text{no}}$ be distributions over $\mathcal{P}$ and $\mathcal{R}$, respectively. If q is such that for all $Q \in \binom{T}{q}$ and $a \in \{0,1\}^Q$ we have*

$$\alpha \Pr_{f \in \mathcal{F}_{\text{yes}}}[f\restriction_Q = a] < \Pr_{f \in \mathcal{F}_{\text{no}}}[f\restriction_Q = a] + \beta \cdot 2^{-q}$$

for some constants $0 \le \beta \le \alpha \le 1$, then any tester for $\mathcal{P}$ with error probability $\le (\alpha - \beta)/2$ must make more than q queries.

Observe that for any fixed $\alpha < 1$ and $\beta > 0$ this implies a lower bound of $\Omega(q)$ queries, since even if $(\alpha - \beta)/2 < 1/3$, the error probability can be reduced from $1/3$ to $(\alpha - \beta)/2$ by a constant (depending on α, β) number of repetitions.

Proof. Assume towards a contradiction that there is such a tester $\mathcal{T}$ making $\le q$ queries; without loss of generality it makes exactly q queries. Define a distribution $\mathcal{D}$ obtained by selecting one of $\mathcal{F}_{\text{yes}}$ and $\mathcal{F}_{\text{no}}$ with probability $\frac{1}{2}$, and then drawing f from the selected distribution. Fix a random seed so that the tester correctly works for $f \in \mathcal{D}$ with probability at least $1 - \frac{\alpha-\beta}{2}$; now the behaviour of the tester can be described by a deterministic decision tree of height q. Each leaf corresponds to a set $Q \in \binom{T}{q}$, along with an evaluation $a : Q \to \{0,1\}$; the leaf is reached if and only if f satisfies the evaluation. Consider the set L corresponding to accepting leaves; f is accepted if and only if there is $(Q,a) \in L$ such that $f\restriction_Q = a$. These $|L|$ events are disjoint, so the probability of acceptance of f is $\sum_{(Q,a)\in L} \Pr[f\restriction_Q = a]$.

Let $p = \Pr_{f \in \mathcal{F}_{\text{yes}}}[f \text{ is accepted}]$, $r = \Pr_{f \in \mathcal{F}_{\text{no}}}[f \text{ is accepted}]$. Applying the hypothesis to each term of the sum $\sum_{(Q,a)\in L} \Pr[f\restriction_Q = a]$ yields $\alpha p < r + \beta$, so $p - r < (1-\alpha)p + \beta \le 1 - \alpha + \beta$. But then the overall success probability of $\mathcal{T}$ when f is taken from $\mathcal{D}$ is $\frac{1}{2} + \frac{p-r}{2} < 1 - \frac{\alpha-\beta}{2}$, contradicting our assumption. □

In practice we sometimes make use of slightly different claims; their proof is still the same.

- The same conclusion holds if instead the inequality

  $$\alpha \Pr_{f \in \mathcal{F}_{\text{no}}}[f\restriction_Q = a] < \Pr_{f \in \mathcal{F}_{\text{yes}}}[f\restriction_Q = a] + \beta \cdot 2^{-q}$$

  is satisfied for all Q, a.

- If $\mathcal{F}_{\text{yes}}$ and $\mathcal{F}_{\text{no}}$ are distributions of functions such that $\Pr_{g\sim\mathcal{F}_{\text{yes}}}[f \in \mathcal{P}], \Pr_{g\sim\mathcal{F}_{\text{no}}}[f \in \mathcal{R}] = 1 - o(1)$, the lemma is not quite applicable as stated. However in that case the success probability of the tester can be no larger than $(1 + p - r + o(1))/2 < 1 - \frac{\alpha-\beta}{2} + o(1)$ (where p and r are as in the proof of the lemma), so an $\Omega(q)$ lower bound still follows.

4 Brief overview of the main proofs

4.1 Overview of the lower bounds

The proof of Theorem 7.1 is done in two steps. First we establish the result for $k = n$, and then we prove that it implies the general case by "padding" the hard-to-test functions obtained before (this requires showing that for any $f, g : \{0,1\}^k \to \{0,1\}$ and their extensions (paddings) $f', g' : \{0,1\}^n \to \{0,1\}$, $\mathrm{distiso}(f', g') = \Omega(\mathrm{distiso}(f, g))$ holds).

A few words concerning the first (and main) step. We fix a function f enjoying some regularity properties; its existence is established via a probabilistic argument. Then we introduce two distributions $\mathcal{F}_{\mathrm{yes}}$ and $\mathcal{F}_{\mathrm{no}}$ such that a function $g \sim \mathcal{F}_{\mathrm{yes}}$ is isomorphic to f and a function $g \sim \mathcal{F}_{\mathrm{no}}$ is ϵ-far from isomorphic to f with overwhelming probability, and then proceed to show indistinguishability of the two distributions with $o(n)$ adaptive queries.

A first idea for $\mathcal{F}_{\mathrm{no}}$ may be to make it the uniform distribution over all Boolean functions $\{0,1\}^n \to \{0,1\}$. However, it is possible for a tester to collect a great deal of information from looking at inputs with very small or very large weight. In particular, just by querying strings $\bar{0}$ and $\bar{1}$ we would obtain a tester that succeeds with probability $3/4$ in distinguishing $\mathcal{F}_{\mathrm{yes}}$ from $\mathcal{F}_{\mathrm{no}}$ if $\mathcal{F}_{\mathrm{no}}$ were completely uniform. To prevent an algorithm from gaining information by querying inputs of very small or very large weight, the functions appearing in both distributions are the same outside the middle layers of the hypercube. We remark that such "truncation" is essential for this result to hold – as Proposition A.1 says (see Section A in the Appendix), random permutations of *any* f can be distinguished from completely random functions with $\widetilde{O}(\sqrt{n})$ queries and *arbitrarily high* constant success probability.

Although it may seem that such an indistinguishability result might be obtained via straightforward probabilistic techniques, the actual proof has to overcome some technical difficulties. We borrow ideas from the work of Babai and Chakraborty [BC10], who proved query-complexity lower bounds for testing isomorphism of uniform hypergraphs. However, in order to be applicable to our problem, we have to extend the method of [BC10] in several ways. One of the main differences is that, because of the need to consider truncated functions, we have to deal with general sets of permutations to prove that a random permutation "shuffles" the values of a function uniformly. To compensate for this lack of structure, we show that any large enough set of permutations that are "independent" in some technical sense has the regularity property we need. Then the result for general sets is established by showing that any large enough set of permutations can be decomposed into a number of such large "independent" sets. This can be deduced from the celebrated theorem of Hajnal and Szemerédi [HS69] on equitable colorings.

Another difference is that for the proof of Corollary 2.2 we need a hard-to-test f that has a circuit of polynomial size, rather than just a random f. To address the second issue we relax the notion of uniformity to $\mathrm{poly}(n)$-wise independence, and then apply standard partial derandomization techniques.

4.2 Overview of the upper bounds

The main ingredient in the proof of Theorem 2.4 is the nearly-optimal junta tester introduced in [Bla09]. In fact, a significant part of the proof deals with analyzing this junta tester, and proving that it satisfies stronger conditions than what was required for testing juntas.

Let us briefly describe the resulting isomorphism tester: The algorithm begins by calling the junta tester, which may either reject (meaning that g is not a k-junta), or otherwise provide a set of $k' \leq k$ blocks (subsets of indices) such that if g is close to some k-junta, then with high probability it is also close to some k'-junta h' that has at most one relevant variable in each of the k' blocks. Using these k' blocks we define an extension h of h' (if $k' < k$), and a noisy sampler that provides random samples $(x, a) \in \{0,1\}^k \times \{0,1\}$, such that $\Pr[h(x) \neq a]$ is sufficiently small. Finally, we use the (possibly correlated) samples of the noisy sampler to test if h is $\epsilon/10$-close to the core function of f or $9\epsilon/10$-far from it.

We note that our approach resembles the high-level idea in the powerful "Testing by Implicit Learning" paradigm of Diakonikolas et al. [DLM$^+$07]. Furthermore, an upper bound of roughly $O(k^4)$ queries to our problem follows easily from the general algorithm of [DLM$^+$07].

Apart from addressing a less general problem, there are several additional reasons why our algorithm attains a better upper bound of $O(k \log k)$. First, in our case the known function is a proper junta, and not just approximated by one. (However, in [CGM11a] is shown that this requirement can be disposed of if the approximation is good enough). Second, in simulating random samples from the core of the unknown function g, we allow a small, possibly correlated, fraction of the samples to be incorrectly labelled. This enables us to generate a random sample with just one query to g, sparing us the need to perform the independence tests of [FKR$^+$04]. Then we perform the final test (the parallel of Occam's razor from [DLM$^+$07]) with a tester that is tolerant (i.e. accepts even if the distance to the defined property is small) and resistant against (possibly correlated) noise.

4.3 Overview of the one-sided-error lower bound

As mentioned earlier, the lower bound, which is the interesting part of Theorem 2.5, is obtained via a lower bound for testing isomorphism to k-parities with one-sided error.

We start with the simple observation that testing isomorphism to k-parities is equivalent to testing isomorphism to $(n-k)$-parities. Since testing 0-parities (constant zero functions) takes $O(1)$ queries, and testing 1-parities (dictatorship functions) takes $O(1)$ queries as well (by Parnas et al. [PRS02]), we are left with the range $2 \leq k \leq n/2$.

We split this range into three parts: small (constant) k, medium k and large k. For small k's a lower bound of $\Omega(\log n)$ is quite straightforward. For the other two ranges, we use the combinatorial theorems of Frankl–Wilson and Frankl–Rödl, which bound the size of families of subsets with restricted intersection sizes. (The reason for this technical case distinction is to comply with the hypotheses of the respective theorems). We obtain lower bounds of $\Omega(k \log(n/k))$.

In all three cases we follow the same methodology: suppose that we want to prove a lower bound of $q = q(n,k)$. We define a function g that is either a k'-parity (for a suitably chosen $k' \neq k$) [7] or a constant, and depends only on n and k. This function is fixed (independent of the tester), and has the property that for all $x^1, \ldots, x^q \in \{0,1\}^n$ there exists a k-parity f satisfying $f(x^i) = g(x^i)$ for all $i \in [q]$. Hence, no matter what the answers to the (adaptive) queries made are, any one-sided error tester of PAR_k making $\leq q$ queries is forced to accept g, even though it is $(1/2)$-far from any k-parity. [8]

[7] Note that not every choice of k' works, even if k and k' are very close to each other. For example, if $k' = k+1$, it is easy to tell PAR_k from $\mathsf{PAR}_{k'}$ by simply querying the all-ones vector.

[8] This is because for two parities p_1 and p_2 of different sizes, $p_1^{\pi} \oplus p_2$ is always a parity of non-zero size and hence

4.4 Overview of the remaining parts

The upper bounds in Theorem 2.5 (testing with one-sided error) and Theorem 2.6 (testing of two unknown functions) are fairly straightforward. The testers start by random sampling, and then perform exhaustive search over all possible permutations, checking if one of them defines an isomorphism that is consistent with the samples. Their analysis is essentially the same as that of Occam's razor.

The lower bound in the setting where both functions are unknown is proved by defining two distributions on *pairs* of functions, the first supported on isomorphic pairs and the second on pairs that are far from being isomorphic. Then Yao's principle is applied via Lemma 3.1, which gives bounds for adaptive testers.

To prove that any function $f : \{0,1\}^n \to \{0,1\}$ is distinguishable from a completely random function (without the truncation) with $\widetilde{O}(\sqrt{n})$ queries (Proposition A.1) we borrow the ideas from [FM08], using which we reduce our problem to testing closeness of distributions, and then we apply the distribution tester of Batu et al. [BFF$^+$01].

5 Proof of Theorem 2.5 – Testing isomorphism with one-sided error

We prove here Theorem 2.5. Note that if $f \in \mathsf{PAR}_k$, then testing isomorphism to f is the same as testing membership in PAR_k. Hence the lower bound in Theorem 2.5 for any $2 \le k \le n$ follows from the next proposition. (If $k \ge n/2$, the $\Omega(n)$ lower bound for k-juntas follows from the $\Omega(n)$ lower bound for $k' \triangleq \lfloor n/2 \rfloor$ because any k'-junta is also a k-junta).

Proposition 5.1 *Let $\epsilon \in (0, \frac{1}{2}]$ be fixed. The following holds for all $n \in \mathbb{N}$:*

- *For any $k \in [2, n-2]$, the query complexity of testing PAR_k with one-sided error is $\Theta(\log \binom{n}{k})$. Furthermore, the upper bound is obtainable with a non-adaptive tester, while the lower bound applies to adaptive tests, and even to the certificate size for proving membership in PAR_k.*
- *For any $k \in \{0, 1, n-1, n\}$, the query complexity of testing PAR_k with one-sided error is $\Theta(1)$.*

For every $f : \{0,1\}^n \to \{0,1\}$ let Isom_f denote the set of functions isomorphic to f. The upper bound in Theorem 2.5 follows from the next proposition.

Proposition 5.2 *Isomorphism to any given $f : \{0,1\}^n \to \{0,1\}$ can be tested with $O(\log |\mathsf{Isom}_f|/\epsilon)$ queries.*

This immediately implies the desired upper bound, since $|\mathsf{Isom}_f| \le \binom{n}{k} \cdot k!$ for any $k \in [n]$ and k-junta f. This also implies the upper bound in the first item of Proposition 5.1, since for a k-parity f, $|\mathsf{Isom}_f| = |\mathsf{PAR}_k| = \binom{n}{k}$.

takes the value 1 on precisely half the inputs.

5.1 Proof of Proposition 5.1 (parity lower bound)

We begin with the following observation, which is immediate from the fact that p is a k-parity if and only if $p(x) \oplus x_1 \oplus \ldots \oplus x_n$ is an $(n-k)$-parity:

Observation 5.1 *Let $\epsilon \in (0, \frac{1}{2}]$, $n \in \mathbb{N}$ and $k \in [0, n]$. Any ϵ-tester for PAR_k can be converted into an ϵ-tester for PAR_{n-k}, while preserving the same query complexity, type of error, and adaptivity.*

As mentioned earlier, the upper bound in the first item of Proposition 5.1 follows by Proposition 5.2. It is also easy to verify that the second item holds for $k = 0$. For $k = 1$, the bound follows from [PRS02], who show that one-sided-error testing of functions for being a 1-parity (monotone dictatorship) can be done with $O(1)$ queries. So, according to Observation 5.1 we only have to prove the lower bound in the first item of Proposition 5.1 for $k \in [2, \lfloor n/2 \rfloor]$.

To this end we make a distinction between three cases. First we prove a lower bound of $\Omega(\log n)$ for any $k \in [2, \lfloor n/2 \rfloor]$. Then a lower bound of $\Omega(\log \binom{n}{k})$ is shown for $k \in [5, \alpha n]$, where $\alpha n \triangleq \lfloor n/2^{12} \rfloor$. Finally we prove a lower bound of $\Omega(k)$ queries that works for $k \in [\alpha n, \lfloor n/2 \rfloor]$. Combining the three bounds will complete the proof.

In all three cases we follow the argument sketched in the overview (Section 4.3).

5.1.1 Lower bound of $\Omega(\log n)$ for $2 \leq k \leq \lfloor n/2 \rfloor$

Let $q = \lfloor \log n \rfloor - 1$, and let $x^1, \ldots, x^q \in \{0,1\}^n$ be the set of queries. For any $k \in [2, \lfloor n/2 \rfloor]$ we let g be the parity on the last $k-2$ variables: $g(x) = x_{n-k+3} \oplus \cdots \oplus x_n$ (in case $k = 2$, g is simply the constant zero function). By the pigeonhole principle, it is possible to find $j, j' \in [n-k+2]$, $j \neq j'$ such that $x^i_j = x^i_{j'}$ for all $i \in [q]$; this is because $2^q < n-k+2$. Let f be the k-parity corresponding to $\{j, j'\} \cup [n-k+3, n]$. Then $f(x^i) = g(x^i)$ for all $i \in [q]$, so the tester must accept g, even though it is $1/2$-far from any k-parity.

This simple idea can only yield lower bounds of $\Omega(\log n)$. We need to generalize it in order to obtain lower bounds that grow with k.

5.1.2 Lower bound of $\Omega(\log \binom{n}{k})$ for $5 \leq k \leq \alpha n$

Let $q = \lfloor \frac{1}{20} \log \binom{n}{k} \rfloor$. Given $k \in [5, \lfloor n/2 \rfloor]$, let $k' \geq 1$ be the smallest integer such that $(k-k')/2$ is a prime power; the reason for this requirement will be explained shortly. Note that $k' < k/2$ as $k \geq 5$. We let g be the k'-parity $g(x) = x_{n-k'+1} \oplus \cdots \oplus x_n$. With a slight abuse of notation, let g also denote the n-bit string with ones exactly in the last k' indices. It suffices to show that for any $x^1, \ldots, x^q \in \{0,1\}^n$ there exists $y \in \{0,1\}^n$ such that

- $|y| = k - k'$,
- $y \cap g = \emptyset$ and
- $\langle y, x^i \rangle \triangleq \bigoplus_{j=1}^n (y_j \cdot x^i_j) = 0$ for all $i \in [q]$.

Indeed, if such a y exists, then the k-parity corresponding to $g \cup y$ is consistent with g on $x^1, \ldots, x^q$.

Let $Y = \{y \in \{0,1\}^n : |y| = k-k' \text{ and } y \cap g = \emptyset\}$. Partition Y into disjoint subsets $\{Y_\alpha\}_{\alpha \in \{0,1\}^q}$, such that $y \in Y_\alpha$ if and only if $\langle y, x^i \rangle = \alpha_i$ for all $i \in [q]$. Clearly, one of the sets Y_α must be of size at least $\binom{n-k'}{k-k'}/2^q$. We interpret the elements of this Y_α as ℓ-subsets of $[m]$, where $\ell \triangleq k - k'$ and

$m \triangleq n - k'$, and show that there must be $y^1, y^2 \in Y_\alpha$ such that $|y^1 \cap y^2| = \ell/2 = (k-k')/2$. Once the existence of such a pair is established, the claim will follow by taking y to be the bitwise XOR of y^1 and y^2. Indeed, it is clear that $|y| = k - k'$ and $y \cap g = \emptyset$, and it is also easy to verify that $\langle y, x^i \rangle = \langle y^1, x^i \rangle \oplus \langle y^2, x_i \rangle = 0$ for all $i \in [q]$.

At this point we appeal to the Frankl-Wilson Theorem:

Theorem 5.3 ([FW81], Thm. 7b; see also [FR87], p. 3) *Let $m \in \mathbb{N}$ and let $\ell \in [m]$ be even, such that $\ell/2$ is prime power. If $\mathcal{F} \subseteq \binom{[m]}{\ell}$ is such that for all $F, F' \in \mathcal{F}$, $|F \cap F'| \neq \ell/2$, then $|\mathcal{F}| \leq \binom{m}{\ell/2}\binom{3\ell/2-1}{\ell}/\binom{3\ell/2-1}{\ell/2}$.*

Let us check that the hypothesis on the size of $\mathcal{F}$ is satisfied when $\mathcal{F} = Y_\alpha$. Let $c \triangleq n/k$; observe that $c \leq m/\ell \leq 2c$. In the following we use the bounds $b(\log(a/b)) \leq \log \binom{a}{b} \leq b(\log(a/b) + 2)$.

We have

$$\begin{aligned}
\log |Y_\alpha| &\geq \log \left(\frac{\binom{n-k'}{k-k'}}{2^q} \right) \geq \log \binom{m}{\ell} - \frac{1}{20} \log \binom{n}{k} \\
&\geq \ell(\log(m/\ell)) - \frac{1}{20} k(\log(n/k) + 2) \\
&\geq \ell(\log c) - \frac{1}{10} \ell(\log c + 2) \\
&= \ell \left(\frac{9}{10} \log c - \frac{1}{5} \right).
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\log \left(\frac{\binom{m}{\ell/2}\binom{3\ell/2-1}{\ell}}{\binom{3\ell/2-1}{\ell/2}} \right) &\leq \frac{\ell}{2}(\log(m/\ell) + 3) + 3\ell/2 \\
&\leq \ell \left(\frac{1}{2} \log c + \frac{7}{2} \right).
\end{aligned}$$

Since $c \geq 2^{12}$, these inequalities together with Theorem 5.3 imply that there must be $y^1, y^2 \in Y_\alpha$ such that $|y^1 \cap y^2| = \ell/2$, as desired.

5.1.3 Lower bound of $\Omega(k)$ for $\alpha n \leq k \leq \lfloor n/2 \rfloor$

The reasoning in this case is very similar, but since for large k the previous method does not work, we have to change a few things. One of them is switching to the related theorem of Frankl and Rödl, using which we can prove a lower bound of $\Omega(k)$ (instead of $\Omega(\log \binom{n}{k})$), but for the current range of k they are asymptotically the same.

Theorem 5.4 ([FR87], Thm. 1.9) *There is an absolute constant $\delta > 0$ such that for any even k the following holds: Let $\mathcal{F}$ be a family of subsets of $[2k]$ such that no two sets in the family have intersection of size $k/2$. Then $|\mathcal{F}| \leq 2^{(1-\delta)2k}$.*

Algorithm 1 (Non-adaptive one-sided-error tester for the known-unknown setting)

1: Let $q \leftarrow \frac{1}{\epsilon}(2 + \ln |\mathsf{Isom}_f|)$.
2: **for** $i = 1$ to q **do**
3: Pick $x^i \in \{0,1\}^n$ uniformly at random.
4: Query g on x^i.
5: **end for**
6: Accept if and only if there exists $h \in \mathsf{Isom}_f$ such that $g(x^i) = h(x^i)$ for all $i \in [q]$.

Let n be large enough with respect to α and δ. Given $k \in [\alpha n, \lfloor n/2 \rfloor]$, we set $q = \delta k$. Assume first that k is even – we mention the additional changes required for odd k below.

We set g to be the zero function, and show that for any $x^1, \ldots, x^q \in \{0,1\}^n$ there exists $y \in \{0,1\}^n$ such that

- $|y| = k$ and
- $\langle y, x^i \rangle = 0$ for all $i \in [q]$.

Let $Y = \{y \in \{0,1\}^n : y \subseteq [2k] \text{ and } |y| = k\}$. As in the previous case, partition Y into disjoint subsets $\{Y_\alpha\}_{\alpha \in \{0,1\}^q}$, such that $y \in Y_\alpha$ if and only if $\langle y, x^i \rangle = \alpha_i$ for all $i \in [q]$. One of the sets Y_α must be of size at least $\binom{2k}{k}/2^q = 2^{2k-1-q}$, which is greater than $2^{(1-\delta)2k}$ for large enough n (and hence k). We interpret the elements of this Y_α as k-subsets of $[2k]$ in the natural way. Thus, by Theorem 5.4, there must be $y^1, y^2 \in Y_\alpha$ such that $|y^1 \cap y^2| = k/2$. Take y to be the bitwise XOR of y^1 and y^2. Clearly $|y| = k$, and $\langle y, x^i \rangle = 0$ for all $i \in [q]$.

For an odd k, we use the 1-parity $g(x) = x_n$ instead of the zero function. We follow the same steps to find $y \subseteq [2k-2]$ of size $|y| = k-1$ such that $\langle y, x^i \rangle = 0$ for all $i \in [q]$. Then, the vector $y \cup \{n\}$ corresponds to a function in PAR_k that is consistent with g on the q queries.

5.2 Proof of Proposition 5.2 (general upper bound)

Consider the simple tester described in Algorithm 1. It is clear that this is a non-adaptive one-sided error tester, and that it only makes $O(\log |\mathsf{Isom}_f|/\epsilon)$ queries to g. So we only need to show that for any f and any g that is ϵ-far from f, the probability of acceptance is small. Indeed, for a fixed $h \in \mathsf{Isom}_f$ the probability that $g(x^i) = h(x^i)$ for all $i \in [q]$ is at most $(1-\epsilon)^q$. Applying the union bound on all functions $h \in \mathsf{Isom}_f$, we can bound the probability of acceptance by $|\mathsf{Isom}_f|(1-\epsilon)^q \leq |\mathsf{Isom}_f| e^{-\epsilon q} < 1/3$. □

An upper bound of $O(\log \binom{n}{k})$ for testing PAR_k follows from Proposition 5.2, but in fact something much stronger holds in this case. Since the distance between any two parity functions is $1/2$, the algorithm from Proposition 5.2 (which can be thought of as a learning algorithm) can actually *decode* the parity bits of the tested function with the same number of queries:

Fact 5.5 *There is a non-adaptive algorithm A that, given n, k and oracle access to $g : \{0,1\}^n \to \{0,1\}$, satisfies the following:*

- *if g is a k-parity then A outputs the k parity indices of g with probability 1;*
- *if g is ϵ-far from being k-parity then A rejects with probability at least $2/3$;*

- *A makes* $O(\log \binom{n}{k})$ *queries to* g.

Furthermore, if we drop the requirement of the second item, A can be even made deterministic.

This contrasts with the matching lower bound that applies even for the much simpler task of deciding whether the size of a given parity is k.

The fact that for all n and k there is such deterministic algorithm can be seen by taking q twice as large as that in Proposition 5.2, arguing that with high probability no two parities agree on all q samples, and fixing a set of samples with this property; alternately, it follows from the existence of binary linear codes of word length n, distance $2k$ and $O(k \log(n/k))$ parity check equations, for k up to $\Omega(n)$. The existence of a uniform algorithm (whose running time is $\mathrm{poly}(n^k)$) is then implied by standard derandomization techniques, such as the method of conditional expectations (c.f. [AS92, Juk01]) applied to the expression

$$\mathop{\mathbb{E}}_{x_1,\ldots,x_q} \mathop{\mathbb{E}}_{f \in \mathsf{PAR}_{2k}} \mathbb{I}[f(x^1) = f(x^2) \ldots = f(x^i) = 0].$$

6 $\Omega(n)$ lower bound for testing isomorphism to most functions

6.1 Definitions and basic results

To prove lower bounds for testing isomorphism to a function f, it suffices to show the stronger claim that one can choose g with $\mathrm{distiso}(f, g) \geq \epsilon$ and such that no tester can reliably distinguish between the cases where a function h is a random permutation of f or a random permutation of g.

Definition 6.1 *Let* $f, g : \{0,1\}^n \to \{0,1\}$ *be Boolean functions and* $\epsilon > 0$. *Consider the distribution* $\mathcal{D}$ *obtained by choosing a random permutation of* f *with probability half, and a random permutation of* g *with probability half.*

We say that the pair (f, g) *is* (q, ϵ)***-hard*** *if* $\mathrm{distiso}(f, g) \geq \epsilon$ *and no tester with oracle access to* $h \sim \mathcal{D}$ *can determine if* $h \cong f$ *or* $h \cong g$ *with overall success probability* $\geq 2/3$ *unless it makes more than* q *queries.*

The existence of a q-hard pair f, g implies a lower bound of q on the query complexity of testing isomorphism to f (or to g, for that matter). The function g will be defined to agree with f on all *unbalanced* inputs, as defined below.

Definition 6.2 *A query* $x \in \{0,1\}^n$ *is* ***balanced*** *if* $\frac{n}{2} - 2\sqrt{n} \leq |x| \leq \frac{n}{2} + 2\sqrt{n}$. *Otherwise, we say that* x *is an* ***unbalanced*** *query.*

Note that the fraction of unbalanced inputs is $2^{-n} \sum_{|i-n/2|>2\sqrt{n}} \binom{n}{i} < 2\exp(-8) < 1/1000$ by standard estimates on the tails of the binomial distribution.

Definition 6.3 *For every* f, *a* ***random*** f***-truncated function*** *is a random function uniformly drawn from the set of all* $g : \{0,1\}^n \to \{0,1\}$ *satisfying* $g(x) = f(x)$ *for all unbalanced* x.

Proposition 6.4 *Fix* $0 < \epsilon < \frac{1}{2}(1 - 10^{-3})$. *For any function* $f : \{0,1\}^n \to \{0,1\}$, *a random* f*-truncated function* g *is* ϵ*-close to isomorphic to* f *with probability at most* $o(1)$.

Proof. Let $N \triangleq |\{0,1\}^n_{\frac{n}{2}\pm 2\sqrt{n}}| = \Omega(2^n)$ and $\eta \triangleq 1 - (2^{n+1}/N)\epsilon > 0$. For any $\pi \in \mathcal{S}_n$, note that $\mathrm{dist}_{n/2\pm 2\sqrt{n}}(f^\pi, g) = (2^n/N)\mathrm{dist}(f^\pi, g)$, where the term on the left hand side denotes the relative distance when the domain is $\{0,1\}^n_{n/2\pm 2\sqrt{n}}$. Then, by the Chernoff bound,

$$\Pr[\mathrm{dist}_{n/2\pm 2\sqrt{n}}(f^\pi, g) < (2^n/N)\epsilon] = \Pr[\mathrm{dist}_{n/2\pm 2\sqrt{n}}(f^\pi, g) < (1-\eta)/2] \leq \exp(-N\eta^2/4) \leq o(\tfrac{1}{n!}).$$

Taking the union bound over all choices of $\pi \in \mathcal{S}_n$ completes the proof. □

In the rest of this section and all its subsections, we assume $\epsilon < \frac{1}{2}(1 - 10^{-3})$. See Remark 6.1 in Section 6.2 for the details on how to deal with any $\epsilon < \frac{1}{2}$.

Let $\mathcal{T}$ denote any deterministic non-adaptive algorithm that attempts to test f-isomorphism with at most q queries to an unknown function g (where $q = \Omega(n)$ is a parameter to be determined later). Let $Q \subseteq \{0,1\}^n$ be the set of queries performed by $\mathcal{T}$ on f. We partition the queries in Q in two: the set Q_b of balanced queries, and the set Q_u of unbalanced queries.

The tester cannot distinguish f from g by making only unbalanced queries. Some unbalanced queries, however, could conceivably yield useful information to the tester and let it distinguish f from g with only a small number of balanced queries. The next proposition shows that this is not the case, and that little information is conveyed by the responses to unbalanced queries.

Definition 6.5 *For a fixed function $f : \{0,1\}^n \to \{0,1\}$, a set Q of queries, and $a : Q \to \{0,1\}$, the **set of permutations of f compatible with Q and a** is*

$$\Pi_f(Q, a) = \{\pi \in \mathcal{S}_n : f^\pi\restriction_Q = a\}$$

Proposition 6.6 *For any function $f : \{0,1\}^n \to \{0,1\}$, any set Q of queries, and any $0 < t < 1$,*

$$\Pr_{\pi\in\mathcal{S}_n}\left[\left|\Pi_f(Q, f^\pi\restriction_Q)\right| < t \cdot \frac{n!}{2^{|Q|}}\right] < t.$$

This implies that when the unknown function g is truncated according to f, with high probability the set $\Pi_f(Q_u, g^\pi\restriction_{Q_u})$ is large, which will be useful later.

Proof. For every $a \in \{0,1\}^{|Q|}$, let $S_a \subseteq \mathcal{S}_n$ be the set of permutations σ for which $g^\sigma\restriction_Q = a$. A set S_a is *small* if $|S_a| < t\frac{n!}{2^{|Q|}}$. The union of all small sets covers less than $2^{|Q|} \cdot t\frac{n!}{2^{|Q|}} = tn!$ permutations, so the probability that a randomly chosen one belongs to a small set is less than t. □

We now examine the balanced queries.

Definition 6.7 *Let $q \in \mathbb{N}$. We say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is q-**regular** if for every set Q_b of at most q* balanced *queries, every function $a : Q_b \to \{0,1\}$ and every set $S \subseteq \mathcal{S}_n$ of permutations of size at least $|S| \geq \frac{1}{3}\frac{n!}{2^{2q}}$,*

$$|\Pr_{\pi\in S}[f^\pi\restriction_{Q_b} = a] - 2^{-q}| < \tfrac{1}{6} \cdot 2^{-q}.$$

It is easy to see that "at most q" may be replaced with "exactly q" in the definition, as long as q does not surpass the total number of unbalanced inputs. Also note that whether f is regular or not depends only on the values it takes on balanced inputs. This restriction is necessary for $\Omega(n)$-regularity to be possible, since the condition implies in particular the existence of $\Omega(2^q)$ elements in the orbit under $\mathcal{S}_n$ of any 1-query set. The requirement that the condition hold for all large enough sets S arises from the need to handle the information obtained from unbalanced queries. [9]

Definition 6.7 is useful because two functions f, g that are both regular and agree on unbalanced inputs will be hard to tell from each other, as they both resemble random functions on balanced inputs. This holds no matter how f is defined on unbalanced inputs. This is formalized in the following lemma:

Lemma 6.8 *If f, g are q-regular, identical on unbalanced inputs, and* $\mathrm{distiso}(f, g) \geq \epsilon$*, then the pair (f, g) is (q, ϵ)-hard.*

Proof. Consider the following two distributions:

- $\mathcal{F}_{\mathrm{yes}}$: pick $\pi \in G$ uniformly at random, and return f^π.
- $\mathcal{F}_{\mathrm{no}}$: pick $\pi \in G$ uniformly at random, and return g^π.

By definition, any $h_1 \in \mathcal{F}_{\mathrm{yes}}$ is isomorphic to f, whereas any $h_2 \in \mathcal{F}_{\mathrm{no}}$ is isomorphic to g and hence ϵ-far from isomorphic to f.

Let $Q = Q_u \cup Q_b$ be any set of at most q queries and $a = (a_u, a_b)$ any set of $|Q|$ responses. We show that

$$\Pr_{\pi \in \mathcal{S}_n}[f^\pi\restriction_Q = a] - \Pr_{\pi \in \mathcal{S}_n}[g^\pi\restriction_Q = a] < \tfrac{1}{3}2^{-q}.$$

There are two cases to consider.

Case 1: $|\Pi_f(Q_u, a_u)| < \frac{1}{3}\frac{n!}{2^{2q}}$. In this case, by Proposition 6.6 we have that $\Pr_\pi[f^\pi\restriction_{Q_u} = a_u] \leq \frac{1}{3}2^{-q}$. This immediately implies that $\Pr_\pi[f^\pi\restriction_Q = a] \leq \frac{1}{3}2^{-q}$ also.

Case 2: $|\Pi_f(Q_u, a_u)| \geq \frac{1}{3}\frac{n!}{2^{2q}}$. Note that

$$\begin{aligned}
\Pr_\pi[f^\pi\restriction_Q = a] &= \Pr_\pi[f^\pi\restriction_{Q_u} = a_u] \cdot \Pr_\pi[f^\pi\restriction_{Q_b} = a_b \mid f^\pi\restriction_{Q_u} = a_u] \\
&= \Pr_\pi[f^\pi\restriction_{Q_u} = a_u] \cdot \Pr_{\pi \in \Pi_f(Q_u, a_u)}[f^\pi\restriction_{Q_b} = a_b] \\
&= (1 \pm \delta)2^{-q} \Pr_\pi[f^\pi\restriction_{Q_u} = a_u],
\end{aligned}$$

where $\delta < 1/6$ and the last line uses the regularity of f.

Similarly, by the regularity of g,

$$\begin{aligned}
\Pr_\pi[g^\pi\restriction_Q = a] &= (1 \pm \delta)2^{-q} \Pr_\pi[g^\pi\restriction_{Q_u} = a_u] \\
&= (1 \pm \delta)2^{-q} \Pr_\pi[f^\pi\restriction_{Q_u} = a_u],
\end{aligned}$$

[9] It would be enough to demand regularity with respect to $S = \mathcal{S}_n$ if all one wanted to show is an $\Omega(n)$ lower bound for *some* functions, as one can render unbalanced queries meaningless by defining f to be zero on all unbalanced inputs.

because f and g are defined identically on unbalanced inputs. (We can choose the same $\delta < 1/6$ for both). Therefore, for any $a : Q \to \{0,1\}$,

$$\Pr_\pi[f^\pi\restriction_Q = a] - \Pr_\pi[g^\pi\restriction_Q = a] < \tfrac{1}{3}2^{-q}\Pr_\pi[f^\pi\restriction_{Q_u} = a_u] \le \tfrac{1}{3}2^{-q},$$

and an appeal to Lemma 3.1 establishes the claim. □

The main step of the proof of existence of regular functions in the next section is to show that any sufficiently "uniform" family of functions contains regular functions.

Definition 6.9 *A distribution $\mathcal{F}$ of Boolean functions on $\{0,1\}^n$ is r-**uniform** if it is r-independent and uniform on sets of r balanced inputs, i.e. for all $Q_b \in \binom{\{0,1\}^n}{\frac{n}{2}\pm 2\sqrt{n}}$ and $a : Q_b \to \{0,1\}$,*

$$\Pr_{f\in\mathcal{F}}[f\restriction_{Q_b} = a] = 2^{-r}.$$

For example, the uniform distribution over all Boolean functions is 2^n-uniform. The reason we deal with this more general case is to establish the existence of relatively simple functions that are hard to test isomorphism to (see Section 7).

6.2 Existence of regular functions

The main tool we need is the following:

Proposition 6.10 *Let $\mathcal{F}$ be an n^4-uniform distribution over Boolean functions. Then a random function from $\mathcal{F}$ is $(\frac{n}{3} - 2\lceil\log n\rceil)$-regular with probability $1 - o(1)$.*

Before providing the proof, we show how it implies the special case of Theorem 2.1 when $k = n$.

Theorem 6.11 *Fix any $0 < \epsilon < \frac{1}{2}$. Let $f : \{0,1\}^n \to \{0,1\}$ be chosen at random from an n^4-uniform distribution $\mathcal{F}$, and let $g : \{0,1\}^n \to \{0,1\}$ be a random f-truncated function. Then with probability $1 - o(1)$, the pair (f,g) is $(\Omega(n),\epsilon)$-hard.*

Hence, for most functions $f : \{0,1\}^n \to \{0,1\}$, testing f-isomorphism requires $\Omega(n)$ queries.

Proof. Assume $\epsilon < \frac{1}{2}(1 - 10^{-3})$; see Remark 6.1 below to see how to handle larger ϵ. For some $q = \Omega(n)$ we can pick one q-regular function f from $\mathcal{F}$ by Proposition 6.10. The distribution of functions drawn from $\mathcal{F}$ and truncated according to f is also n^4-uniform, so a random such g is also q-regular with probability $1 - o(1)$. Also with probability $1 - o(1)$ we have $\mathrm{distiso}(f,g) = \Omega(1)$. [10] By the union bound some g satisfies both conditions. By Lemma 6.8, the pair f, g is q-hard and f needs more than q queries to test isomorphism to. The "hence" part follows by taking for $\mathcal{F}$ the uniform distribution among all functions. □

Proof of Proposition 6.10. Let $q \triangleq \frac{n}{3} - 2\log n$. Fix a set Q_b of q balanced queries, a function $a : Q_b \to \{0,1\}$, and a set $S \subseteq \mathcal{S}_n$ of size $|S| \ge \frac{1}{3}\frac{n!}{2^{2q}}$. For any $f : \{0,1\}^n \to \{0,1\}$ and $\pi \in S$,

[10] The proof is the same as that of Proposition 6.4, except that we use n^4-independence in place of full independence, and employ the variation of Chernoff bounds stated below in Theorem 6.14. This leads to a bound of $\exp(-\Omega(n^4))$ instead of $\exp(-\Omega(2^n))$, but is still $o(1/n!)$.

define the indicator variable $X(f,\pi) \triangleq \mathbb{I}[f^{\pi}\restriction_{Q_b} = a]$ and define $A(f) \triangleq \Pr_{\pi\in S}[X(f,\pi)=1]$. We aim to compute the probability, over a random function f drawn from $\mathcal{F}$, that $A(f)$ deviates from 2^{-q} by $2^{-q}/6$ or more. Since $q \le n^4$, the n^4-uniform distribution $\mathcal{F}$ is also q-uniform. As a result, $\mathbb{E}_f A(f) = \mathbb{E}_\pi \mathbb{E}_f X(f,\pi) = \mathbb{E}_\pi \Pr_{f\sim\mathcal{F}}[f^{\pi}\restriction_{Q_b} = a] = \mathbb{E}_\pi 2^{-q} = 2^{-q}$.

Consider any pair $\sigma_1, \sigma_2 \in S$ such that $\sigma_1(Q_b) \cap \sigma_2(Q_b) = \emptyset$. Since $2q \le n^4$, a random function from $\mathcal{F}$ assigns values independently to each element of $\sigma_1(Q_b) \cup \sigma_2(Q_b)$, so the random variables $X(f,\sigma_1)$ and $X(f,\sigma_2)$ are independent conditioned on the choice of σ_1, σ_2.

More generally, for any s permutations $\sigma_1, \ldots, \sigma_s$ of S under which the images of Q_b are pairwise disjoint, the variables $X(f,\sigma_1), \ldots, X(f,\sigma_s)$ are $n^4/q \ge n^3$-wise independent. We show that S can be partitioned into a number of large sets of permutations, each of them satisfying the pairwise disjointness property. The proof of this claim uses the celebrated theorem of Hajnal and Szemerédi [HS69].

Theorem 6.12 (Hajnal-Szemerédi) *Let G be a graph on n vertices with maximum vertex degree $\Delta(G) \le d$. Then G has a $(d+1)$-coloring in which all the color classes have size $\lfloor \frac{n}{d+1} \rfloor$ or $\lceil \frac{n}{d+1} \rceil$.*

Lemma 6.13 *Let S be a set of permutations on $[n]$ (with $n \ge 13$) and let Q_b be a set of at most $q < n$ balanced queries. Then there exists a partition $S_1 \dot\cup \cdots \dot\cup S_k$ of the permutations in S such that for $i = 1, 2, \ldots, k$,*

(i) $|S_i| \ge \frac{|S|}{n!}\frac{2^n}{2n^2\sqrt{n}} - 1$, *and*

(ii) The sets $\{\pi(Q_b)\}_{\pi\in S_i}$ are pairwise disjoint.

Proof. Construct a graph G on S where two permutations σ, τ are adjacent iff there exist $x, y \in Q_b$ such that $\sigma(x) = \tau(y)$ or $\sigma(y) = \tau(x)$. By this construction, when T is a set of permutations that form an independent set in G, the sets $\{\pi(Q_b)\}_{\pi\in T}$ are pairwise disjoint.

Let $N \triangleq \binom{n}{n/2-2\sqrt{n}} = \frac{2^n}{\sqrt{n}}\left(\sqrt{\frac{2}{\pi}} + o(1)\right) \ge \frac{2^n}{2\sqrt{n}}$. Note that for any $x, y \in \{0,1\}^n_{\frac{n}{2}\pm 2\lceil\sqrt{n}\rceil}$,

$$\Pr_{\pi\in\mathcal{S}_n}[\pi(x) = y] = \left\{ \begin{array}{ll} 0, & |x| \ne |y| \\ \frac{1}{\binom{n}{|x|}}, & |x| = |y| \end{array} \right\} \le \frac{1}{N}.$$

This holds because the orbit of x under $\mathcal{S}_n$ is the set of all $\binom{n}{|x|}$ strings of the same weight. So by applying the union bound over all choices of $x, y \in Q_b$, we can upper bound the degree of G by $d \triangleq q^2 n!/N < n^2 n!/N$. Therefore, by the Hajnal-Szemerédi Theorem, G can be colored so that each color class has size at least

$$\left\lfloor \frac{|S|}{d+1} \right\rfloor \ge \frac{|S|}{n!}\frac{2^n}{2n^2\sqrt{n}} - 1.$$

□

In our case $|S|/n! \ge 2^{-2q}/3$, and by our choice of q we conclude that each of the elements of the partition has size at least $|S_i| \ge n^3 \cdot 2^q$ for large enough n. Since $A(f)$ is a weighted average of the random variables $Y_i(f) \triangleq \mathbb{E}_{\pi\in S_i} X(f,\pi)$, it is enough to show that with probability $1 - o(1)$

$$|Y_i(f) - 2^{-q}| < 2^{-q}/6$$

holds simultaneously for all $i = 1, \ldots, k$.

Each quantity $Y_i(f)$ is the average of $|S_i|$ random variables that are n^3-wise independent, each satisfying $\mathbb{E}_f X(f, \pi) = 2^{-q}$. We apply the following version of Chernoff bounds:

Theorem 6.14 (Chernoff bounds for k-wise independence.) *[SSS95] Let X be the sum of s k-wise independent random variables in the interval $[0, 1]$, and let $p = \frac{1}{s}\mathbb{E}[X]$. For any $0 \leq \delta \leq 1$,*

$$\Pr[|X - p| \geq \delta p] \leq e^{-\Omega(\min(k, \delta^2 ps))}.$$

Since $2^{-q}|S_i| \geq n^3$ and $k = n^3$, using the above theorem with $\delta = \frac{1}{6}$ we obtain that for all $i \in [k]$,

$$\Pr_f[|Y_i(f) - 2^{-q}| \geq \delta 2^{-q}] \leq 2^{-\Omega(n^3)},$$

hence we can bound

$$\Pr_f[|A(f) - 2^{-q}| \geq \delta 2^{-q}] \leq \Pr_f[\exists i \in [k] : |Y_i(f) - 2^{-q}| \geq \delta 2^{-q}] \leq k 2^{-\Omega(n^3)} = n^3 2^{-\Omega(n^3)}.$$

To conclude the proof we apply the union bound over all possible choices of Q and $a \in \{0, 1\}^Q$, yielding

$$\Pr_f[\exists\, Q, a : |A(f) - 2^{-q}| \geq 2^{-q}/6] \leq \binom{2^n}{q} 2^q n^3 2^{-\Omega(n^3)} = o(1).$$

□

Remark 6.1 *It is not difficult to see that if one replaces $\frac{n}{2} \pm 2\sqrt{n}$ in the definition of balanced inputs with $\frac{n}{2} \pm c\sqrt{n}$ for some other constant $c > 2$, the result still holds for* the same *lower bound q and large enough n. We refrained from doing so and introducing an additional parameter in all the definitions and proofs. The only place where this matters is in claiming the $\Omega(n)$ lower bound for* any *fixed $\epsilon < \frac{1}{2}$. The value $c = 2$ only suffices for $\epsilon < \frac{1}{2}(1 - 10^{-3})$ because of Proposition 6.4, but choosing larger values can prove the theorem for any constant $\epsilon < \frac{1}{2}$.*

7 Proof of Theorem 2.1 and its consequences

Here we prove the following stronger version of Theorem 2.1.

Theorem 7.1 *For every $\epsilon < \frac{1}{4}$ and all large enough $k \leq n$, it holds that for $1 - o(1)$ fraction of all k-juntas $f : \{0, 1\}^n \to \{0, 1\}$, a random f-truncated k-junta g satisfies that (f, g) is $(\Omega(k), \epsilon)$-hard with probability $1 - o(1)$. Moreover, such k-juntas f can have either one of the following properties:*

- *f can be written as a polynomial of degree $O(\log k)$ over $\mathbb{F}_2$;*
- *f can be in non-uniform $\mathcal{NC}$, i.e. computed by bounded fan-in circuits of size* $\mathrm{poly}(k)$ *and depth $O(\mathrm{polylog}(k))$.*

7.1 $\Omega(k)$ lower bound for k-juntas

The lower bound in Theorem 7.1 for testing isomorphism to k-juntas is obtained by combining the $\Omega(n)$ lower bound of Theorem 6.11 with the following lemma, which uses a "preservation of distance under padding" argument to allow us to embed a function on k variables into one on n variables, so that the hardness of testing remains roughly the same. [11]

Lemma 7.2 (extension from $\{0,1\}^k$ to $\{0,1\}^n$) *Let $k, n \in \mathbb{N}$, $k \leq n$, and let $f', g' : \{0,1\}^k \to \{0,1\}$ be a pair of functions. Define $f = \mathrm{pad}(\mathrm{f}')$ to be the padding extension of f', where $f : \{0,1\}^n \to \{0,1\}$ is given by $f(x) = f'(x\restriction_{[k]})$ for all $x \in \{0,1\}^n$. Likewise, define $g = \mathrm{pad}(\mathrm{g}')$. Then the following holds:*

- $\mathrm{distiso}(f', g') \geq \mathrm{distiso}(f, g) \geq \mathrm{distiso}(f', g')/2$.
- *If (f', g') is (q, ϵ)-hard, then (f, g) is $(q, \epsilon/2)$-hard.*

Note that the inequality $\mathrm{distiso}(f, g) \geq \mathrm{distiso}(f', g')/2$ is tight for some functions. Consider, for example, the case where $n = k + 1$, $f'(x) = |x| \bmod 2$ and $g'(x) = 1 - f(x)$.

Proof. Take π for which $\mathrm{distiso}(f, g^\pi) = \epsilon$. The function f is a junta on $[k]$, while g^π is a junta on $\pi^{-1}([k])$. Let $A = \pi^{-1}([k]) \setminus [k]$, $B = \pi^{-1}([k]) \cap [k]$, $C = [k] \setminus B$; note that $|A| = |C|$. Roughly speaking, if f and g^π are close then both must be close to a junta on B because bits outside are irrelevant to either f or g^π.

Every input $x \in \{0,1\}^n$ is the interleaving of strings $a \in \{0,1\}^A, b \in \{0,1\}^B, c \in \{0,1\}^C, r \in \{0,1\}^{[n]-(A \cup B \cup C)}$ in the right order, i.e. $x = \sigma(a, b, c, r)$ for some permutation $\sigma : [n] \to [n]$, where (a, b, c, r) denotes concatenation. Hence there are permutations σ_1, σ_2 of $[k]$ for which $f(x) = f'^{\sigma_1}(b, c)$ and $g^\pi(x) = g'^{\sigma_2}(b, a)$.

For every $b \in \{0,1\}^B$ and $i, j \in \{0,1\}$, let

$$p^b_{ij} \triangleq \Pr_a[f'^{\sigma_1}(b, a) = i \wedge g'^{\sigma_2}(b, a) = j] = \Pr_c[f'^{\sigma_1}(b, c) = i \wedge g'^{\sigma_2}(b, c) = j]$$

[11] It appears likely that one can obtain an $\Omega(k)$ lower bound for any $\epsilon < 1/2$, as opposed to any $\epsilon < 1/4$, by arguing that for two random functions f', g', the isomorphism distance between their extensions is still very close to $1/2$ instead of $\mathrm{distiso}(f', g')/2 \approx 1/4$. Similar remarks apply to the lower bounds for degree and circuit size, but we will not pursue this direction here.

Obviously $p^b_{01} + p^b_{10} = \Pr_a[f'^{\sigma_1}(b,a) \neq g'^{\sigma_2}(b,a)] \leq 1$, so $p^b_{01} + p^b_{10} \geq (p^b_{01} + p^b_{10})^2 \geq 4p^b_{01}p^b_{10}$. As a, b, c are mutually independent for random x, we can compute

$$\begin{aligned}
\Pr_{a,c}[f'^{\sigma_1}(b,c) \neq g'^{\sigma_2}(b,a)] &= \Pr_c[f'^{\sigma_1}(b,c) = 0] \cdot \Pr_a[g'^{\sigma_2}(b,a) = 1] \\
&\quad + \Pr_c[f'^{\sigma_1}(b,c) = 1] \cdot \Pr_a[g'^{\sigma_2}(b,a) = 0] \\
&= (p^b_{00} + p^b_{01})(p^b_{01} + p^b_{11}) + (p^b_{10} + p^b_{11})(p^b_{00} + p^b_{10}) \\
&\geq p^b_{01}(p^b_{00} + p^b_{01} + p^b_{11}) + p^b_{10}(p^b_{00} + p^b_{10} + p^b_{11}) \\
&= p^b_{01}(1 - p^b_{10}) + p^b_{10}(1 - p^b_{01}) \\
&= \frac{p^b_{01} + p^b_{10}}{2} + \frac{p^b_{01} + p^b_{10} - 4p^b_{01}p^b_{10}}{2} \\
&\geq \frac{p^b_{01} + p^b_{10}}{2} \\
&= \frac{\Pr_a[f'^{\sigma_1}(b,a) \neq g'^{\sigma_2}(b,a)]}{2}.
\end{aligned}$$

Hence, by taking expectations over b,

$$\mathrm{dist}(f, g^\pi) = \Pr_{a,b,c}[f'^{\sigma_1}(b,c) \neq g'^{\sigma_2}(b,a)] \geq \frac{1}{2}\Pr_{b,a}[f'^{\sigma_1}(b,a) \neq g'^{\sigma_2}(b,a)] = \frac{1}{2}\mathrm{dist}(f'^{\sigma_1}, g'^{\sigma_2}).$$

This implies $\mathrm{distiso}(f, g) \geq \frac{1}{2}\mathrm{distiso}(f', g')$. (The other inequality is obvious, and is not used in the remainder).

It is clear that if $f' \cong g'$ then $f \cong g$. Let there be an algorithm $\mathcal{A}$ capable of distinguishing a random permutation of f from a random permutation of g using fewer than q queries. Based on $\mathcal{A}$, we can construct an algorithm to distinguish whether $h' : \{0,1\}^k \to \{0,1\}$ is a random permutation of f' or a random permutation of g' in the following manner: pick a uniformly random permutation $\sigma \in \mathcal{S}_n$, and apply $\mathcal{A}$ to $\mathrm{pad(h)}^\sigma$ (clearly, any query to $\mathrm{pad(h')}^\sigma$ can be simulated by one query to h', and the distribution of $\mathrm{pad(h')}^\sigma$ is a random permutation of either f or g). Hence no such $\mathcal{A}$ exists. □

7.2 Low-degree polynomials over $\mathbb{F}_2$

We show in the next lemma that there is an n^4-uniform distribution over low-degree polynomials. Combining this lemma with Theorem 6.11 completes the lower bound in Theorem 7.1 for testing isomorphism to low-degree polynomials over $\mathbb{F}_2$.

Lemma 7.3 *Let $\mathcal{F}_d$ be the set of all polynomials $p : \mathbb{F}_2^n \to \mathbb{F}_2$ of degree at most d. Then the uniform distribution over $\mathcal{F}_d$ is $(2^{d+1} - 1)$-uniform.*

Proof. To prove independence, it is enough to prove the following claim: for any set $S \subseteq \mathbb{F}_2^n$ of size $|S| < 2^{d+1}$, and any function $f : S \to \mathbb{F}_2$, there is a polynomial $q \in \mathcal{F}_d$ such that $q\restriction_S = f$ (this fact has been generalized in the works of [KS05] and [BEHL09]). Indeed, if the claim holds then $\Pr_{p \in \mathcal{F}_d}[p\restriction_S = f] = \Pr_{p \in \mathcal{F}_d}[(p \oplus q)\restriction_S = 0] = \Pr_{p' \in \mathcal{F}_d}[p'\restriction_S = 0]$, since the distributions of p and $p' \triangleq p \oplus q$ are uniform over $\mathcal{F}_d$. Therefore this probability is the same for every f.

We prove now this fact by induction on $|S|+n$; it is trivial for $|S|=n=0$. Suppose that, after removing the first bit of each element of S, we still get $|S|$ distinct vectors; then we can apply the induction hypothesis with S and $n-1$. Otherwise, there are disjoint subsets $A, B, C \subseteq \{0,1\}^{n-1}$ such that $S = \{0,1\} \times A \cup \{0\} \times B \cup \{1\} \times C$, and $A \neq \emptyset$.

We can find, by induction, a polynomial $p_{0A,0B,1C}$ of degree $\leq d$ on $n-1$ variables that computes f on $\{0\} \times A \cup \{0\} \times B \cup \{1\} \times C$. As $|S| = 2|A| + |B| + |C|$, either $|A| + |B|$ or $|A| + |C|$ is at most $\frac{|S|}{2} < 2^d$; assume the latter. Then any function $g : A \cup C \to \mathbb{F}$ can be evaluated by some polynomial $p_{AC}(y)$ of degree $\leq d-1$; consider $g(y) = 0$ if $y \in C$ and $g(y) = f(1,y) - p_{0A,0B,1C}(1,y)$ if $y \in A$. Then the polynomial $p(x,y) = p_{0A,0B,1C}(y) + xp_{AC}(y)$ does the job. □

7.3 Small circuits

To complete the lower bound in Theorem 7.1 for the query complexity of testing isomorphism to functions computable by small circuits, we just need the following fact:

Proposition 7.4 (see, e.g., [AS92]) *There is an n^4-uniform distribution $\mathcal{F}$ over $\mathcal{NC}$ circuits.*

One example of a distribution that proves Proposition 7.4 is the distribution over circuits that computes a uniformly random polynomial of degree n^4 over the finite field of size 2^n and returns the last bit of the result. These circuits are known to belong to $\mathcal{NC}$. The size of these circuits is $O(n^c)$ for some small constant $c \geq 1$.

7.4 Applications to other testing problems

Here we prove the lower bounds for testing size-s Boolean circuits and degree-d Boolean functions using our lower bounds for testing isomorphism.

Proof. [Proof of Corollary 2.2] Fix $r = \Theta(s^{1/c})$. Theorem 7.1 shows that there is a function $f' : \{0,1\}^r \to \{0,1\}$ such that f' can be computed by circuits of size r^c (for some constant c) and for an f'-truncated random function $g' : \{0,1\}^r \to \{0,1\}$, the pair (f', g') is $(\Omega(r), 2\epsilon)$-hard. With overwhelming probability, g' will be far from all circuits of size r^c (and even of size $2^{c'r}$ for some c'). Consider the functions $f, g : \{0,1\}^n \to \{0,1\}$ obtained by the padding extensions $f = \mathrm{pad(f')}$ and $g = \mathrm{pad(g')}$. By Lemma 7.2, the pair (f,g) is also $(\Omega(r), \epsilon)$-hard. Since the extension does not change the size of the Boolean circuit that computes the corresponding functions, the query complexity of testing a function of size-s Boolean circuits is $\Omega(r) = \Omega(s^{1/c})$. □

Proof. [Proof of Corollary 2.3] Any d-junta f can be written as a polynomial of degree at most d over any field. The result follows by the main part of Theorem 7.1 and a simple counting argument showing that nearly all f-truncated functions g are ϵ-far from all polynomials of degree less than $\frac{n}{2} - 2\sqrt{n}$:

Let $g : \{0,1\}^n \to \{0,1\}$ be any function. Because the domain is $\{0,1\}^n$, it can be represented by a unique multilinear polynomial over $\mathbb{F}$

$$g(x) = \sum_{S \subseteq [n]} c_S \cdot \mathrm{And_S(x)}$$

where $\mathrm{And_S}$ is the monomial corresponding to the subset S, and where $c_S \in \mathbb{F}$. It is well-known that

$$c_S = \sum_{T \subseteq S} (-1)^{|T|-|S|} g(T).$$

(Note that we are identifying subsets $T \subseteq [n]$ with elements of $\{0,1\}^n$.) In particular, c_S is determined by the values that g takes on inputs $T \subseteq S$.

Imagine we enumerate all balanced inputs $S_1, \ldots, S_m$ in order of increasing weight, and we pick a random f-truncated function g. By the observations above, at most one choice for $g(S_i)$ can make $c_{S_i} = 0$ given all previous $g(S_j)$, $j < i$. So the probability that $c_{S_i} = 0$, conditioned on all previous assignments to $g(S_j)$ cannot exceed $\frac{1}{2}$. Therefore

$$\Pr_g[g \text{ has degree} < \tfrac{n}{2} - 2\sqrt{n}] \leq \Pr_g[c_{S_i} = 0 \text{ for all } i \in [m]] \leq \frac{1}{2^m}.$$

On the other hand, there are at most $2^{mH(\epsilon 2^n/m)}$ f-truncated Boolean functions at distance $\leq \epsilon$ from any given one, where H denotes Shannon's entropy function. As $H(\epsilon 2^n/m)$ is bounded away from 1,

$$\Pr_g[g \text{ is } \epsilon\text{-close to a polynomial of degree } < \tfrac{n}{2} - 2\sqrt{n}] \leq 2^{2^m H(\epsilon 2^n/m) - 2^m} = 2^{-\Omega(2^n)} = o(1),$$

where we used the union bound and the distribution of $g \oplus e$ being uniform for random g and fixed e. □

8 Proof of Theorem 2.4 – isomorphism testers for k-juntas

In this section we prove that $O(k \log k)$ queries suffice to test isomorphism against any function $f : \{0,1\}^n \to \{0,1\}$ that is a k-junta.

High-level overview of the proof. The first ingredient in our proof is a tolerant, noise-resistant and bias-resistant isomorphism tester RobustIsoTest (Algorithm 2 below). Informally, RobustIsoTest allows us to test isomorphism of an unknown g to a known function f, even if instead of an oracle access to g we are given a sampler that produces pairs (x, a), where

- there is some h that is close to g, and $\Pr[h(x) = a]$ is high;
- the distribution of the x's from the sampled pairs is close to uniform.

The basic idea that allows us to use RobustIsoTest for testing isomorphism to k-juntas is the following: if we could simulate a noisy almost-uniform sampler to the core of h, where $h : \{0,1\}^n \to \{0,1\}$ is the presumed k-junta that is close to $g : \{0,1\}^n \to \{0,1\}$, then we could test whether g is isomorphic to f. What we show is, roughly speaking, that for the aforementioned simulation it suffices to detect k disjoint subsets $J_1, \ldots, J_k \subseteq [n]$ such that each subset contains at most one relevant variable of the presumed k-junta $h : \{0,1\}^n \to \{0,1\}$.

To obtain such sets we use the second ingredient, which is the optimal junta tester of Blais [Bla09]. This tester, in addition to testing whether g is a k-junta, can provide (in case g is close to some k-junta h) a set of $\leq k$ blocks (sets of indices), such that each block contains exactly one of the relevant variables of h. The trouble is that the k-junta h may not be the closest one to g. In fact, even if g is a k-junta itself, h may be some other function that is only close to g. Taking these considerations into account constitutes the bulk of the proof.

Algorithm 2 (RobustIsoTest – tests if $f \cong g$, tolerantly with noise)

1: Let $q \leftarrow \frac{20}{\epsilon^2} + \frac{7\ln(k!)}{\epsilon}$.
2: Obtain q independent samples $(x^1, a^1), \ldots, (x^q, a^q)$ from $\widetilde{g}$.
3: Accept iff there exists a permutation π of $[k]$ such that $\left|\left\{i \in [q] : f^\pi(x^i) \neq a^i\right\}\right| < \epsilon q/2$.

8.1 Testing isomorphism between the cores

In the following we use the term *black-box algorithm* for algorithms that take no input.

Definition 8.1 *Let $g : \{0,1\}^k \to \{0,1\}$ be a function, and let $\eta, \mu \in [0,1)$. An (η,μ)-**noisy sampler for** g is a black-box probabilistic algorithm $\widetilde{g}$ that on each execution outputs $(x,a) \in \{0,1\}^k \times \{0,1\}$ such that*

- *$x \in \{0,1\}^k$ is distributed according to some distribution $\mathcal{D}$ on $\{0,1\}^k$ whose variation distance to the uniform distribution is at most μ; namely, for all $A \subseteq \{0,1\}^k$, $\left|\Pr_{x\sim\mathcal{D}}[x \in A] - |A|/2^k\right| \leq \mu$;*
- *$\Pr[a = g(x)] \geq 1 - \eta$,*

where the probability is taken over the randomness of $\widetilde{g}$, which also determines x.

We stress that the two items are **not** necessarily independent; e.g., it may be that for some $\alpha \in \{0,1\}^k$, $\Pr[a = g(x) \mid x = \alpha] = 0$.

The following is essentially a strengthening of Occam's razor that is both tolerant, noise-resistant and bias-resistant:

Proposition 8.2 *There is an algorithm* RobustIsoTest *that, given $\epsilon \in \mathbb{R}^+$, $k \in \mathbb{N}$, a function $f : \{0,1\}^k \to \{0,1\}$ and a (η,μ)-noisy sampler $\widetilde{g}$ for some $g : \{0,1\}^k \to \{0,1\}$, where $\eta \leq \epsilon/100$ and $\mu \leq \epsilon/10$, satisfies the following:*

- *if* $\mathrm{distiso}(f,g) < \epsilon/10$*, it accepts with probability at least* $9/10$*;*
- *if* $\mathrm{distiso}(f,g) > 9\epsilon/10$*, it rejects with probability at least* $9/10$*;*
- *it draws $O(\frac{k\log k}{\epsilon} + \frac{1}{\epsilon^2})$ samples from $\widetilde{g}$.*

Proof. Consider the tester described in Algorithm 2. It is clear that RobustIsoTest uses $O(\frac{k\log k}{\epsilon} + \frac{1}{\epsilon^2})$ queries.

We first bound the fraction of samples (x^i, a^i) for which $g(x^i) \neq a^i$ using Markov's inequality:

$$\Pr\left[\frac{1}{q}|\{i \in [q] : g(x^i) \neq a^i\}| \geq \epsilon/5\right] \leq \Pr\left[\frac{1}{q}|\{i \in [q] : g(x^i) \neq a^i\}| \geq 20\eta\right] \leq 1/20.$$

For a permutation π, let

$$s_\pi \triangleq \frac{1}{q}|\{i \in [q] : f^\pi(x^i) \neq a_i)\}|$$

be the fraction of samples on which f^π disagrees with a^i. For any i for which $f^\pi(x^i) \neq a_i$, either $f^\pi(x^i) \neq g(x^i)$ or $g(x^i) \neq a_i$; consequently, there is probability at least 19/20 that s_π lies within distance $\epsilon/5$ from

$$d_\pi \triangleq \frac{1}{q}|\{i \in [q] : f^\pi(x^i) \neq g(x^i)\}|.$$

Let $\delta_\pi = \mathrm{dist}(f^\pi, g)$ and let $\Delta_\pi \subseteq \{0,1\}^k$, $|\Delta_\pi| = \delta_\pi 2^k$, be the set of *inputs* on which f^π and g disagree. Since each x^i is an independent random variables distributed according to some distribution $\mathcal{D}$ that is μ-close to uniform, we have

$$\zeta_\pi \triangleq \Pr_{x\sim\mathcal{D}}[x \in \Delta_\pi] = \delta_\pi \pm \mu$$

(by $a = b \pm c$ we mean $|a - b| \leq c$).

Observe that

- If $\zeta_\pi \leq \epsilon/10$, then by the additive Chernoff bound,

$$\Pr[d_\pi \geq 2\epsilon/5] \leq \Pr[d_\pi \geq \zeta_\pi + 3\epsilon/10] \leq \exp(-18q\epsilon^2/100) < 1/20.$$

- If $\zeta_\pi \geq 4\epsilon/5$, then by the multiplicative Chernoff bound,

$$\Pr[d_\pi \leq 3\epsilon/5] \leq \Pr[d_\pi \leq (3/4)\zeta_\pi] \leq \exp(-3q\epsilon/20) < \frac{1}{20(k!)}.$$

Applying the union bound we see that with probability at least 19/20,

- If for some π we have $\zeta_\pi < \epsilon/10$, then for some π we have $d_\pi < 2\epsilon/5$.
- For all π with $\zeta_\pi > 4\epsilon/5$, it holds that $d_\pi > 3\epsilon/5$.

Recalling that with probability at least 19/20, $|d_\pi - s_\pi| \leq \epsilon/10$, we conclude that with probability 9/10,

- If $\mathrm{distiso}(f, g) < \epsilon/10$ then there exists π with $s_\pi \leq d_\pi + \epsilon/10 < \epsilon/2$.
- If $\mathrm{distiso}(f, g) > 9\epsilon/10$, then for all π it holds that $s_\pi \geq d_\pi - \epsilon/10 > \epsilon/2$.

This proves the proposition.

□

8.2 Some definitions and lemmas

Definition 8.3 *Given a k-junta $f : \{0,1\}^n \to \{0,1\}$ we define $\mathsf{core}_k(f) : \{0,1\}^k \to \{0,1\}$ to be the restriction of f to its relevant variables (where the variables are placed according to the natural order of $[n]$). In case f has fewer than k relevant variables, $\mathsf{core}_k(f)$ is extended to a $\{0,1\}^k \to \{0,1\}$ function by adding dummy variables.*

Throughout the rest of this section, a random partition $\mathcal{I} = I_1, \ldots, I_\ell$ of $[n]$ into ℓ sets is constructed by starting with ℓ empty sets, and then putting each coordinate $i \in [n]$ into one of the ℓ sets picked uniformly at random. Unless explicitly mentioned otherwise, $\mathcal{I}$ will always denote a random partition $\mathcal{I} = I_1, \ldots, I_\ell$ of $[n]$ into ℓ subsets, where ℓ is even; and $\mathcal{J} = J_1, \ldots, J_k$ will denote an (ordered) k-subset of $\mathcal{I}$ (meaning that there are $a_1, \ldots, a_k$ such that $J_i = I_{a_i}$ for all $i \in [k]$).

Definition 8.4 (Operators $\mathsf{replicate}$ and $\mathsf{extract}$) *We call $y \in \{0,1\}^n$ $\mathcal{I}$-**constant** if the restriction of y on every set of $\mathcal{I}$ is constant; that is, if for all $i \in [\ell]$ and $j, j' \in I_i$, $y_j = y_{j'}$.*

- *Given $z \in \{0,1\}^\ell$, define $\mathsf{replicate}_{\mathcal{I}}(z)$ to be the $\mathcal{I}$-constant string $y \in \{0,1\}^n$ obtained by setting $y_j \leftarrow z_i$ for all $i \in \ell$ and $j \in I_i$.*
- *Given an $\mathcal{I}$-constant $y \in \{0,1\}^n$ and an ordered subset $\mathcal{J} = (J_1, \ldots, J_k)$ of $\mathcal{I}$ define $\mathsf{extract}_{\mathcal{I},\mathcal{J}}(y)$ to be the string $x \in \{0,1\}^k$ where for every $i \in [k]$: $x_i = y_j$ if $j \in J_i$; and x_i is a uniformly random bit if $J_i = \emptyset$.*

Definition 8.5 (Distributions $\mathcal{D}_{\mathcal{I}}$ and $\mathcal{D}_{\mathcal{J}}$) *For any $\mathcal{I}$ and $\mathcal{J} \subseteq \mathcal{I}$ as above, we define a pair of distributions:*

- *The distribution $\mathcal{D}_{\mathcal{I}}$ on $\{0,1\}^n$: A random $y \sim \mathcal{D}_{\mathcal{I}}$ is obtained by*
  1. *picking $z \in \{0,1\}^\ell$ uniformly at random among all $\binom{\ell}{\ell/2}$ strings of weight $\ell/2$;*
  2. *setting $y \leftarrow \mathsf{replicate}_{\mathcal{I}}(z)$.*
- *The distribution $\mathcal{D}_{\mathcal{J}}$ on $\{0,1\}^{|\mathcal{J}|}$: A random $x \sim \mathcal{D}_{\mathcal{J}}$ is obtained by*
  1. *picking $y \in \{0,1\}^n$ at random, according to $\mathcal{D}_{\mathcal{I}}$;*
  2. *setting $x \leftarrow \mathsf{extract}_{\mathcal{I},\mathcal{J}}(y)$.*

Lemma 8.6 (Properties of $\mathcal{D}_{\mathcal{I}}$ and $\mathcal{D}_{\mathcal{J}}$)

1. *For all $\alpha \in \{0,1\}^n$, $\Pr_{\mathcal{I}, y \sim \mathcal{D}_{\mathcal{I}}}[y = \alpha] = 1/2^n$;*
2. *Assume $\ell > 4|\mathcal{J}|^2$. For every $\mathcal{I}$ and $\mathcal{J} \subseteq \mathcal{I}$, the total variation distance between $\mathcal{D}_{\mathcal{J}}$ and the uniform distribution on $\{0,1\}^{|\mathcal{J}|}$ is bounded by $2|\mathcal{J}|^2/\ell$. Moreover, the L_∞ distance between the two distributions is at most $4|\mathcal{J}|^2/(\ell 2^{|\mathcal{J}|})$.*

Proof.

1. Each choice of $z \in \{0,1\}^\ell$, $|z| = \ell/2$, in Definition 8.5 splits $\mathcal{I}$ into two equally-sized sets: $\mathcal{I}^0$ and $\mathcal{I}^1$; and the bits corresponding to indices in $\mathcal{I}^b$ (where $b \in \{0,1\}$) are set to b in the construction of y. For each index $i \in [n]$, the block it is assigned to is chosen independently at random from $\mathcal{I}$, and therefore falls within $\mathcal{I}^0$ (or $\mathcal{I}^1$) with probability $1/2$, independently of other $j \in [n]$. (This actually shows that the first item of the lemma still holds if z is an arbitrarily fixed string of weight $\ell/2$, rather than a randomly chosen one).

2. Let $k = |\mathcal{J}|$. Let us prove the claim about the Ł_∞ distance, which implies the other one. We only need to take care of the case were all sets J_i in $\mathcal{J}$ are non-empty; having empty sets can only decrease the distance to uniform. Let $w \in \{0,1\}^k$. The choice of $y \sim \mathcal{D}_\mathcal{I}$, in the process of obtaining $x \sim \mathcal{D}_\mathcal{J}$, is independent of $\mathcal{J}$; thus, for every $i \in [k]$ we have

$$\Pr_{x \sim \mathcal{D}_\mathcal{J}}[x_i = w_i \mid x_j = w_j \ \forall j < i] \leq \frac{\ell/2}{\ell - k} < \frac{1}{2} + \frac{k}{\ell},$$

and

$$\Pr_{x \sim \mathcal{D}_\mathcal{J}}[x_i = w_i \mid x_j = w_j \ \forall j < i] \geq \frac{\ell/2 - k}{\ell - k} > \frac{1}{2} - \frac{k}{\ell}.$$

Using the inequalities $1 - my \leq (1-y)^m$ for all $y < 1, m \in \mathbb{N}$ and $(1+y)^m \leq e^{my} \leq 1 + 2my$ for all $m \geq 0, 0 \leq my \leq 1/2$, we conclude

$$\Pr_{x \sim \mathcal{D}_\mathcal{J}}[x = w] = \left(\frac{1}{2} \pm \frac{k}{\ell}\right)^k = \frac{1}{2^k}\left(1 \pm \frac{4k^2}{\ell}\right).$$

whereas a truly uniform distribution U should satisfy $\Pr_{x \sim U}[x = w] = 1/2^k$.

□

Definition 8.7 (Black-box algorithm sampler) *Given $\mathcal{I}, \mathcal{J}$ as above and oracle access to $g : \{0,1\}^n \to \{0,1\}$, we define a probabilistic black-box algorithm $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ that on each execution produces a pair $(x, a) \in \{0,1\}^{|\mathcal{J}|} \times \{0,1\}$ as follows: it picks a random $y \sim \mathcal{D}_\mathcal{I}$ and outputs the pair $(\mathsf{extract}_{\mathcal{I},\mathcal{J}}(y), g(y))$.*

Note that just one query is made to g in every execution of $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$. Notice also that the x in the pairs $(x, a) \in \{0,1\}^{|\mathcal{J}|} \times \{0,1\}$ produced by $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ is distributed according to distribution $\mathcal{D}_\mathcal{J}$ defined above.

8.3 From junta testers to noisy samplers

Given a function $g : \{0,1\}^n \to \{0,1\}$, we denote by $g^* : \{0,1\}^n \to \{0,1\}$ the k-junta that is closest to g (if there are several k-juntas that are equally close, break ties using some arbitrarily fixed scheme). Clearly, if g is itself a k-junta then $g^* = g$.

We make repeated use of the following lemma:

Lemma 8.8 *[FKR+04] For any $f : \{0,1\}^n \to \{0,1\}$ and $A \subseteq [n]$*

$$\mathrm{dist}(f, \mathsf{Jun}_A) \leq \mathrm{Inf_f}([\mathrm{n}] \setminus \mathrm{A}) \leq 2 \cdot \mathrm{dist}(\mathrm{f}, \mathsf{Jun}_\mathrm{A}).$$

We also use the fact (see [FKR+04, Bla09] for a proof) that influence is monotone and subadditive; namely, for all $f : \{0,1\}^n \to \{0,1\}$ and $A, B \subseteq [n]$,

$$\mathrm{Inf_f}(\mathrm{A}) \leq \mathrm{Inf_f}(\mathrm{A} \cup \mathrm{B}) \leq \mathrm{Inf_f}(\mathrm{A}) + \mathrm{Inf_f}(\mathrm{B}).$$

For the following definition and lemma the reader should keep in mind the distributions $\mathcal{D}_\mathcal{I}$ and $\mathcal{D}_\mathcal{J}$ from Definition 8.5.

Definition 8.9 *Given $\delta > 0$, function $g : \{0,1\}^n \to \{0,1\}$, partition $\mathcal{I} = I_1, \ldots, I_\ell$ of $[n]$ and a k-subset $\mathcal{J}$ of $\mathcal{I}$ (where $\ell > 4k^2$), we call the pair $(\mathcal{I}, \mathcal{J})$* δ-good *(with respect to g) if there exists a k-junta $h : \{0,1\}^n \to \{0,1\}$ such that the following conditions are satisfied.*

1. *Conditions on h:*
   (a) *Every relevant variable of h is also a relevant variable of g^* (recall that g^* denotes the k-junta closest to g);*
   (b) $\mathrm{dist}(g^*, h) < \delta$.
2. *Conditions on $\mathcal{I}$:*
   (a) *For all $j \in [\ell]$, I_j contains at most one variable of $\mathsf{core}_k(g^*)$;* [12]
   (b) $\Pr_{y \sim \mathcal{D}_\mathcal{I}}[g(y) \neq g^*(y)] \leq 10 \cdot \mathrm{dist}(g, g^*)$;
3. *Conditions on $\mathcal{J}$:*
   (a) *The set $\bigcup_{I_j \in \mathcal{J}} I_j$ contains* all *relevant variables of h;*

Lemma 8.10 *Let $\delta, g, \mathcal{I}, \mathcal{J}$ be as in the preceding definition. If the pair $(\mathcal{I}, \mathcal{J})$ is δ-good with respect to g, then $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ is an (η, μ)-noisy sampler for some permutation of $\mathsf{core}_k(g^*)$, with $\eta \leq 2\delta + 4k^2/\ell + 10 \cdot \mathrm{dist}(g, g^*)$ and $\mu \leq 4k^2/\ell$.*

Proof. By item 2b in Definition 8.9, it suffices to prove that

$$\Pr_{y \sim \mathcal{D}_\mathcal{I}}[g^*(y) \neq \mathsf{core}_k(g^*)^\pi(\mathsf{extract}_{\mathcal{I},\mathcal{J}}(y))] < 2\delta + 4k^2/\ell$$

for some π.

Let h be the k-junta that witnesses the fact that the pair $(\mathcal{I}, \mathcal{J})$ is δ-good. Let $V \subseteq [n]$ be the set of k variables of $\mathsf{core}_k(g^*)$. (Recall that V may actually be a superset of the relevant variables of g^*.) Let $\mathcal{J}' \triangleq \{I_j \in \mathcal{I} : I_j \cap V \neq \emptyset\}$ be an ordered subset respecting the order of $\mathcal{J}$, and let π be the permutation whose inverse maps the i-th relevant variable of g^* (in the standard order) to the index of the element of $\mathcal{J}'$ in which it is contained. We assume without loss of generality that π is the identity map.

It follows from Definition 8.9 that $|\mathcal{J}'| = |V| = k$, since each block in $\mathcal{I}$ contains at most one variable of $\mathsf{core}_k(g^*)$. For any $\mathcal{I}$-uniform $y \in \{0,1\}^n$, let $x \triangleq \mathsf{extract}_{I,\mathcal{J}}(y)$ and $x' \triangleq \mathsf{extract}_{I,\mathcal{J}'}(y)$ denote the k-bit strings corresponding to $\mathcal{J}$ and $\mathcal{J}'$. By definitions, we have the equalities

(1) $g^*(y) = \mathsf{core}_k(g^*)(x')$,

(2) $\mathsf{core}_k(h)(x) = \mathsf{core}_k(h)(x')$.

The first equality is by Definition 8.4, and the second one follows from items 1a and 3a in Definition 8.9. From item 1b we also have

(3) $\Pr_{r \in \{0,1\}^k}[\mathsf{core}_k(g^*)(r) \neq \mathsf{core}_k(h)(r)] < \delta$,

where r is picked uniformly at random. However, by the second item of Lemma 8.6, the distribution

[12] Note that this, along with 1a, implies that every block I_j contains at most one relevant variable of h, since the variables of $\mathsf{core}_k(g^*)$ contain all relevant variables of g^*.

$\mathcal{D}_\mathcal{J}$ is $2k^2/\ell$ close to uniform [13]; combining this with (3) we also get

(4) $\Pr_{y\sim\mathcal{D}_\mathcal{I}}[\mathsf{core}_k(g^*)(x) \neq \mathsf{core}_k(h)(x)] < \delta + 2k^2/\ell$.

Likewise, we have

(5) $\Pr_{y\sim\mathcal{D}_\mathcal{I}}[\mathsf{core}_k(g^*)(x') \neq \mathsf{core}_k(h)(x')] < \delta + 2k^2/\ell$,

thus, using $(2, 4, 5)$ and the union bound we get

(6) $\Pr_{y\sim\mathcal{D}_\mathcal{I}}[\mathsf{core}_k(g^*)(x') \neq \mathsf{core}_k(g^*)(x)] < 2\delta + 4k^2/\ell$.

Combining (1) and (6) we conclude that

$$\Pr_{y\sim\mathcal{D}_\mathcal{I}}[g^*(y) \neq \mathsf{core}_k(g^*)(x)] < 2\delta + 4k^2/\ell,$$

and the claim follows. □

Corollary 8.11 *If the pair $(\mathcal{I}, \mathcal{J})$ is δ-good (with respect to g), then $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ is (η, μ)-noisy sampler for a permutation of $\mathsf{core}_k(g^*)$, with $\eta \leq 2\delta + 4k^2/\ell + 10 \cdot \mathrm{dist}(g, g^*)$ and $\mu \leq 4k^2/\ell$.*

Proof. Recall that $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ is a probabilistic black-box algorithm that on each execution produces a pair $(x, a) \in \{0,1\}^k \times \{0,1\}$ as follows: it picks a random $y \sim \mathcal{D}_\mathcal{I}$ and outputs the pair $(x, a) \triangleq (\mathsf{extract}_{\mathcal{I},\mathcal{J}}(y), g(y))$.

To be an (η, μ)-noisy sampler for $\mathsf{core}_k(g^*)^\pi$, $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ has to satisfy the following:

- the distribution of $x \in \{0,1\}^k$ in its pairs should be μ close to uniform (in total variation distance);
- $\Pr_{(x,a)\leftarrow\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)}\left[a = \mathsf{core}_k(g^*)^\pi(x)\right] \geq 1 - \eta$.

The first item follows from the second item of Lemma 8.6. The second item follows from Lemma 8.10. □

Now we set up a version of the junta tester from [Bla09] that is needed for our algorithm. A careful examination of the proof in [Bla09] yields the following:

Theorem 8.12 (Corollary to [Bla09]) *The property Jun_k can be tested with one-sided error using $O(k \log k + k/\epsilon)$ queries.*

Moreover, the tester $\mathrm{T}^\star$ can take a (random) partition $\mathcal{I} = I_1, \ldots, I_\ell$ of $[n]$ as input, where $\ell = \ell_{[Bla09]}(k, \epsilon) = \Theta(k^9/\epsilon^5)$ is even, and output (in case of acceptance) a k-subset $\mathcal{J}$ of $\mathcal{I}$ such that for any g the following conditions hold (the probabilities below are taken over the randomness of the tester and the construction of $\mathcal{I}$):

- *if g is a k-junta, $\mathrm{T}^\star$ always accepts;*
- *if g is $\epsilon/2400$-far from Jun_k, then $\mathrm{T}^\star$ rejects with probability at least $9/10$;*
- *for* any *g, with probability at least $4/5$ either $\mathrm{T}^\star$ rejects, or it outputs $\mathcal{J}$ such that the pair $(\mathcal{I}, \mathcal{J})$ is $\epsilon/600$-good (as per Definition 8.9). (In particular, if g is a k-junta then with probability at least $4/5$, $\mathrm{T}^\star$ outputs a set $\mathcal{J}$ such that $(\mathcal{I}, \mathcal{J})$ is $\epsilon/600$-good.)*

[13] Recall that $\mathcal{D}_\mathcal{J}$ is a distribution on $\{0,1\}^k$, where a random $x \sim \mathcal{D}_\mathcal{J}$ is obtained by picking a random $y \sim \mathcal{D}_\mathcal{I}$ and setting $x \leftarrow \mathsf{extract}_{\mathcal{I},\mathcal{J}}(y)$.

Proof. In view of the results stated in [Bla09], only the last item needs justification. [14]

We start with a brief description of how $\mathrm{T}^\star$ works. Given the partition $\mathcal{I}$, $\mathrm{T}^\star$ starts with an empty set $S = \emptyset$, and iteratively finds indices $j \in [\ell] \setminus S$ such that for some pair of inputs $y, y' \in \{0,1\}^n$, $y_{\restriction_{[n]\setminus I_j}} = y'_{\restriction_{[n]\setminus I_j}}$ but $g(y) \neq g(y')$. In other words, it finds j such that I_j contains at least one influential variable (let us call such a block I_j *relevant*). Then j is joined to S, and the algorithm proceeds to the next iteration. $\mathrm{T}^\star$ stops at some stage, and rejects if and only if $|S| > k$. If g is not rejected (i.e. if $\mathrm{T}^\star$ terminates with $|S| \leq k$), then

$(*)$ with probability at least 19/20 the set S satisfies $\mathrm{Inf_g}\Big([\mathrm{n}] \setminus (\bigcup_{\mathrm{j}\in\mathrm{S}} \mathrm{I_j})\Big) \leq \epsilon/4800$.

We will use this S to construct the subset $\mathcal{J} \subseteq \mathcal{I}$ as follows:

- for every $j \in S$, we put the block I_j into $\mathcal{J}$;
- if $|S| < k$ then we extend $\mathcal{J}$ by putting in it $k - |S|$ additional "dummy" blocks from $\mathcal{I}$ (some of them possibly empty), obtaining a set $\mathcal{J}$ of size exactly k.

Now we go back to proving the third item of Theorem 8.12. Recall that g^* denotes the closest k-junta to g. Let $R \in \binom{[n]}{\leq k}$ denote the set of the relevant variables of g^*, and let $V \in \binom{[n]}{k}$, $V \supseteq R$, denote the set of the variables of $\mathsf{core}_k(g^*)$. Assume that $\mathrm{dist}(g, \mathsf{Jun}_k) \leq \epsilon/2400$, [15] and $\mathrm{T}^\star$ did not reject. In this case,

- by $(*)$, with probability at least 19/20 the set $\mathcal{J}$ satisfies
$$\mathrm{Inf_g}\Big([\mathrm{n}] \setminus (\bigcup_{\mathrm{I_j}\in\mathcal{J}} \mathrm{I_j})\Big) \;\leq\; \mathrm{Inf_g}\Big([\mathrm{n}] \setminus (\bigcup_{\mathrm{j}\in\mathrm{S}} \mathrm{I_j})\Big) \leq \epsilon/4800;$$
- since $\ell \gg k^2$, with probability larger than 19/20 all elements of V fall into different blocks of the partition $\mathcal{I}$;
- by Lemma 8.6, $\mathrm{Pr}_{\mathcal{I},y\sim\mathcal{D}_\mathcal{I}}\Big[g(y) = g^*(y)\Big] = \mathrm{dist}(g, g^*)$; hence by Markov's inequality, with probability at least 9/10 the partition $\mathcal{I}$ satisfies $\mathrm{Pr}_{y\sim\mathcal{D}_\mathcal{I}}[g(y) \neq g^*(y)] \leq 10 \cdot \mathrm{dist}(g, g^*)$.

So with probability at least 4/5, all three of these events occur. Now we show that conditioned on them, the pair $(\mathcal{I}, \mathcal{J})$ is $\epsilon/600$-good.

Let $U = R \cap (\bigcup_{I_j\in\mathcal{J}} I_j)$. Informally, U is the subset of the relevant variables of g^* that were successfully "discovered" by $\mathrm{T}^\star$. Since $\mathrm{dist}(g, g^*) \leq \epsilon/2400$, we have $\mathrm{Inf_g}([\mathrm{n}] \setminus \mathrm{V}) \leq \epsilon/1200$ (by Lemma 8.8). By the subadditivity and monotonicity of influence we get

$$\mathrm{Inf_g}([\mathrm{n}] \setminus \mathrm{U}) \leq \mathrm{Inf_g}([\mathrm{n}] \setminus \mathrm{V}) + \mathrm{Inf_g}(\mathrm{V} \setminus \mathrm{U}) \leq \mathrm{Inf_g}([\mathrm{n}] \setminus \mathrm{V}) + \mathrm{Inf_g}\Big([\mathrm{n}] \setminus (\bigcup_{\mathrm{I_j}\in\mathcal{J}} \mathrm{I_j})\Big) \leq \epsilon/960,$$

where the second inequality follows from $V \setminus U \subseteq [n] \setminus (\bigcup_{I_j\in\mathcal{J}} I_j)$. This means, by Lemma 8.8, that there is a k-junta h in Jun_U satisfying $\mathrm{dist}(g, h) \leq \epsilon/960$, and by triangle inequality, $\mathrm{dist}(g^*, h) \leq \epsilon/2400 + \epsilon/960 < \epsilon/600$. Based on this h, we can verify that the pair $(\mathcal{I}, \mathcal{J})$ is $\epsilon/600$-good by going over the conditions in Definition 8.9. □

[14] The somewhat different constants can be easily achieved by increasing (by a constant factor) the number of iterations and partition sizes of the algorithm.

[15] For other g's the third item follows from the second item.

Algorithm 3 (Tests isomorphism to a k-junta f)

1: Let $\ell = \ell_{[\text{Bla09}]}(k, \epsilon) = \Theta(k^9/\epsilon^5)$.
2: Randomly partition $[n]$ into $\mathcal{I} = (I_1, \ldots, I_\ell)$.
3: Test g for being a k-junta, using $\mathrm{T}^\star$ with $\mathcal{I} = I_1, \ldots, I_\ell$. (See Theorem 8.12)
4: **if** $\mathrm{T}^\star$ rejects **then**
5: Reject.
6: **end if**
7: Let $\mathcal{J} \subseteq \mathcal{I}$ be the set output by $\mathrm{T}^\star$.
8: Construct $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$. (See Section 8.2)
9: Accept iff RobustIsoTest($\mathsf{core}_k(f), \mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$) accepts. (See Section 8.1)

9 Proof of Theorem 2.4

Consider the tester described in Algorithm 3. Theorem 2.4 follows from the next proposition:

Proposition 9.1 *Algorithm 3 satisfies the following conditions:*

1. *if $g \cong f$ then it accepts with probability at least $2/3$;*
2. *if $\mathrm{distiso}(f, g) \geq \epsilon$ then it rejects with probability at least $2/3$;*
3. *its query complexity is $O(k \log k/\epsilon + 1/\epsilon^2)$.*

Proof of item 1. Assume $g \cong f$, and hence $\mathsf{core}_k(g) \cong \mathsf{core}_k(f)$. Since g is a k-junta, Algorithm 3 does not reject on line 5, because $\mathrm{T}^\star$ has one-sided error. So in this case, by Theorem 8.12, with probability at least $4/5$ the pair $(\mathcal{I}, \mathcal{J})$ is $\epsilon/600$-good. If so, by Corollary 8.11, $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ is a (η, μ)-noisy sampler for a function isomorphic to $\mathsf{core}_k(g^*) = \mathsf{core}_k(g)$, where $\eta \leq 2\epsilon/600 + 4k^2/\ell + 10 \cdot 0 < \epsilon/100$ and $\mu \leq 4k^2/\ell < \epsilon/10$, and hence RobustIsoTest accepts with probability at least $9/10$. Thus the overall acceptance probability is at least $2/3$.

Proof of item 2. If $\mathrm{distiso}(f, g) \geq \epsilon$ then one of the following must hold:

- either g is $\epsilon/2400$-far from Jun_k,
- or $\mathrm{dist}(g, \mathsf{Jun}_k) = \mathrm{dist}(g, g^*) \leq \epsilon/2400$ and $\mathrm{distiso}(\mathsf{core}_k(f), \mathsf{core}_k(g^*)) \geq \epsilon - \epsilon/2400 > 9\epsilon/10$.

If the first case holds, then $\mathrm{T}^\star$ rejects with probability greater than $2/3$ and we are done. So assume that the second case holds.

By the third item of Theorem 8.12, with probability at least $4/5$, $\mathrm{T}^\star$ either rejects g, or the pair $(\mathcal{I}, \mathcal{J})$ is $\epsilon/600$ good. If $\mathrm{T}^\star$ rejects then we are done. Otherwise, if an $\epsilon/600$-good pair is obtained, then by Corollary 8.11, $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ is a (η, μ)-noisy sampler for a function isomorphic to $\mathsf{core}_k(g^*)$, where $\eta \leq 2\epsilon/600 + 4k^2/\ell + 10 \cdot \epsilon/2400 < \epsilon/100$ and $\mu \leq 4k^2/\ell < \epsilon/10$, and hence RobustIsoTest rejects with probability at least $9/10$. Thus the overall rejection probability is at least $2/3$.

Proof of item 3. As for the query complexity, it is the sum of $O(k \log k + k/\epsilon)$ queries made by $\mathrm{T}^\star$, and additional $O(k \log k/\epsilon^2)$ queries made by RobustIsoTest.

This completes the proof of Theorem 2.4. □

9.1 Query-efficient procedure for drawing random samples from the core

We conclude this section by observing that the tools developed above can be used for drawing random samples from the core of a k-junta g, so that generating each sample requires only one query to g.

Proposition 9.2 *Let $\gamma > 0$ be an arbitrary constant. There is a randomized algorithm A, that given oracle access to any k-junta $g : \{0,1\}^n \to \{0,1\}$ satisfies:*

- *Algorithm A has two parts: preprocessor A_P and sampler A_S. A_P is executed only once; it makes $O(k \log k)$ queries to g, and produces a state $\alpha \in \{0,1\}^{\mathrm{poly}(n)}$. The sampler A_S can then be called on demand, with the state α as an argument; in each call, A_S makes only one query to g and outputs a pair $(x, a) \in \{0,1\}^k \times \{0,1\}$.*
- *With probability at least $4/5$, the state α produced by A_P is such that for some permutation $\pi : [k] \to [k]$,*

$$\Pr_{(x,a) \leftarrow A_S(\alpha)} [\mathsf{core}(g)^\pi(x) = \alpha] \geq 1 - \gamma.$$

Furthermore, the x's generated by the sampler A_S are independent random variables, distributed uniformly on $\{0,1\}^k$.

Proof. The preprocessor A_P starts by constructing a random partition $\mathcal{I}$ and calling the junta tester $\mathrm{T}^\star$ with $\epsilon \triangleq \gamma$. Then A_P encodes in the state α the partition $\mathcal{I}$ and the subset $\mathcal{J} \subseteq \mathcal{I}$ output by $\mathrm{T}^\star$ (see Theorem 8.12).

The sampler, given $\alpha = (\mathcal{I}, \mathcal{J})$, obtains a pair $(x, a) \in \{0,1\}^k \times \{0,1\}$ by executing $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ (once). Then, with probability p_x (defined bellow), A_P outputs (x, a); and with probability $1 - p_x$ it draws a uniformly random $z \in \{0,1\}^k$ and outputs $(z, 0)$.

By Theorem 8.12 (third item), since g is a k-junta, with probability at least $4/5$, the pair $\mathcal{I}, \mathcal{J}$ is $\epsilon/600$-good. So, by Corollary 8.11, $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ is a (η, μ)-noisy sampler for a function isomorphic to $\mathsf{core}_k(g^*) = \mathsf{core}_k(g)$, where $\eta \leq 2\epsilon/600 + 4k^2/\ell + 10 \cdot 0 < \epsilon/100$ and $\mu \leq 4k^2/\ell < \epsilon/100$. Moreover, the distribution of x in the pairs produced by $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ is $2^{-k}\mu < \epsilon 2^{-k}/100$ close to uniform in L_∞ norm. Since we need this distribution to be uniform, we use rejection sampling, with the only difference being that since $\mu \leq \epsilon/100 \ll 1$, we can stop after one execution of $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$ at the cost of a small increase in the error probability.

Concretely, after drawing sample (x, a) from $\mathsf{sampler}_{\mathcal{I},\mathcal{J}}(g)$, we accept it with probability

$$p_x \triangleq \frac{\Pr_{x_1 \sim U}[x_1 = x]}{(1+\mu)\Pr_{x_2 \sim D_\mathcal{J}}[x_2 = x]};$$

and with probability $1 - p_x$ we reject the sample (and output a uniformly random pair $(z, 0)$ instead). It is easy to verify that the overall acceptance probability is $\mathbb{E}_{x \sim D_\mathcal{J}} p_x = 1/(1+\mu)$ and thus, conditioned on acceptance, the distribution of x is uniform. In the case of rejection (which occurs with probability $\mu/(1+\mu)$) it is uniform by definition; hence the overall distribution of x is uniform too, and $\Pr[a \neq g(x)] \leq \epsilon/100 + \mu/(1+\mu) < \epsilon/50 < \gamma$. □

Algorithm 4 (Non-adaptive one-sided error tester for the unknown-unknown setting)

1: Generate a set Q by including every $x \in \{0,1\}^n$ in Q w.p. $\sqrt{\frac{n \ln n}{\epsilon 2^n}}$ independently at random.
2: **if** $|Q| > 10\sqrt{\frac{2^n}{\epsilon} n \ln n}$ **then**
3: Accept.
4: **end if**
5: Query both f and g on all inputs in Q.
6: Accept iff there exists π such that for all $x \in Q$, either $f(x) = g(\pi(x))$ or $\pi(x) \notin Q$.

10 Proof of Theorem 2.6 – Testing isomorphism of two unknown functions

Recall that an ϵ-tester for function isomorphism in the unknown-unknown setting is a probabilistic algorithm $\mathcal{A}$ that, given oracle access to two functions $f, g : \{0,1\}^n \to \{0,1\}$, satisfies the following conditions: (1) if $f \cong g$ it accepts with probability at least 2/3; (2) if $\text{distiso}(f,g) \geq \epsilon$ it rejects with probability at least 2/3.

In the rest of the section we prove the following restatement of Theorem 2.6.

Theorem 10.1 *For any fixed $\epsilon > 0$,*

1. *There exists a non-adaptive ϵ-tester with one-sided error for function isomorphism in the unknown-unknown setting that has query complexity $O(2^{n/2}\sqrt{n \log n/\epsilon})$; and*
2. *Any adaptive tester for function isomorphism in the unknown-unknown setting must have query complexity $\Omega(2^{n/2}/n^{1/4})$.*

10.1 Proof of the upper bound

In this section we show that isomorphism of a pair of unknown functions can be tested with a one-sided-error non-adaptive tester that makes $O(2^{n/2}\sqrt{n \log n})$ queries. The tester is described in Algorithm 4.

It is clear that Algorithm 4 is non-adaptive, has one-sided error and makes $O(2^{n/2}\sqrt{n \log n})$ queries. Let f and g be ϵ-far up to isomorphism; we prove that the probability of the tester accepting is $o(1)$. We may assume that the event $|Q| \leq 10\sqrt{2^n n \ln n/\epsilon}$ holds, since it occurs with probability $1 - o(1)$. For any permutation $\pi \in \mathcal{S}_n$ there are at least $\epsilon 2^n$ inputs $x \in \{0,1\}^n$ for which $f(x) \neq g(\pi(x))$. When x satisfies this condition, the probability that both x and $\pi(x)$ belong to Q is at least $\frac{n \ln n}{\epsilon 2^n}$, so the permutation π passes the acceptance condition in the last line of Algorithm 4 with probability no more than $(1 - n \ln n/(\epsilon 2^n))^{\epsilon 2^n} \leq e^{-n \ln n} = n^{-n} = o(1/n!)$. The claim follows by taking the union bound over all $n!$ permutations.

10.2 Proof of the lower bound

In this section we prove that any two-sided adaptive tester in the unknown-unknown setting must make $\widetilde{Omega}(2^{n/2})$ queries.

We define two distributions $\mathcal{F}_{\text{yes}}$ and $\mathcal{F}_{\text{no}}$ on pairs of functions such that any pair of functions drawn from $\mathcal{F}_{\text{yes}}$ are isomorphic, while any pair drawn from $\mathcal{F}_{\text{no}}$ is 1/8-far from isomorphic with probability $1-o(1)$. The distribution $\mathcal{F}_{\text{yes}}$ is constructed by letting the pair of functions be (f, f^π), where $f \in \mathcal{F}_{\frac{n}{2}\pm 2\sqrt{n}}$ is a random truncated function on $\{0,1\}^n$ (see Definition 6.3) and $\pi \in \mathcal{S}_n$ is a uniformly random permutation.

For the distribution $\mathcal{F}_{\text{no}}$ the pair of functions are two independently chosen random truncated functions f and g; with probability $1-o(1)$, distiso$(f,g) \geq 1/8$ (Proposition 6.4). For any set $Q = \{x^1, \ldots, x^t\} \subseteq \{0,1\}^n$ of t queries and any $p, q \in \{0,1\}^t$ let $\Pr_{(f,g)\in\mathcal{F}_{\text{yes}}}[(f,g)\restriction_Q = (p,q)]$ be the probability that for all $1 \leq i \leq t$, $f(x^i) = p_i$ and $g(x^i) = q_i$ when f and g are drawn according to $\mathcal{F}_{\text{yes}}$. Similarly we define $\Pr_{(f,g)\in\mathcal{F}_{\text{no}}}[(f,g)\restriction_Q = (p,q)]$.

Without loss of generality we may assume that $|x^i| \in [\frac{n}{2} - 2\sqrt{n}, \frac{n}{2} + 2\sqrt{n}]$ for all $i \in [t]$, since functions drawn from $\mathcal{F}_{\text{yes}}$ or $\mathcal{F}_{\text{no}}$ always take the value 0 on all other inputs. If the pair f, g is drawn from $\mathcal{F}_{\text{no}}$, the answers to the queries will be uniformly distributed by definition, so for any $p, q \in \{0,1\}^t$, we have

$$\Pr_{(f,g)\in\mathcal{F}_{\text{no}}}[(f,g)\restriction_Q = (p,q)] = 1/2^{2t}.$$

Now let the pair be drawn according to $\mathcal{F}_{\text{yes}}$ and let π be the permutation that defined the pair. Let E_Q denote the event that $\pi(Q)$ and Q are disjoint, i.e., that for all $i, j \in [t]$, the inequality $\pi(x^i) \neq x^j$ holds. Conditioned on E_Q, the answers to the queries will again be distributed uniformly, that is

$$\Pr_{(f,g)\in\mathcal{F}_{\text{yes}}}[(f,g)\restriction_Q = (p,q) \mid E_Q] = \Pr_{(f,g)\in\mathcal{F}_{\text{no}}}[(f,g)\restriction_Q = (p,q)].$$

(Note that the event in question is independent of E_Q when the pairs is drawn from $\mathcal{F}_{\text{no}}$.)

Let us now show that E_Q occurs with probability at least $\frac{3}{4}$. For $t \leq 2^{n/2}/(3n^{1/4})$ and any fixed $i, j \in [t]$, we have that $\Pr_\pi[\pi(x^i) = x^j] \leq 1/\binom{n}{k} \leq \frac{2\sqrt{n}}{2^n}$ since $\frac{n}{2} - 2\sqrt{n} \leq |x^i| \leq \frac{n}{2} + 2\sqrt{n}$. So by the union bound, since $t \leq 2^{n/2}/(3n^{1/4})$ we have that

$$\Pr[E_Q] \geq 1 - \frac{2t^2\sqrt{n}}{2^n} \geq \frac{3}{4}.$$

Therefore,

$$\begin{aligned}\Pr_{(f,g)\in\mathcal{F}_{\text{yes}}}[(f,g)\restriction_Q = (p,q)] &\geq \Pr[E_Q] \cdot \Pr_{(f,g)\in\mathcal{F}_{\text{yes}}}[(f,g)\restriction_Q = (p,q) \mid E_Q] \\ &\geq \tfrac{3}{4} \cdot \Pr_{(f,g)\in\mathcal{F}_{\text{no}}}[(f,g)\restriction_Q = (p,q)].\end{aligned}$$

By Lemma 3.1, this implies that the success probability of any tester that makes fewer than $2^{n/2}/(3n^{1/4})$ queries is at most $5/8 + o(1) < 2/3$ and completes the proof of the lower bound in Theorem 10.1.

Acknowledgements.

We thank Ronald de Wolf for many valuable comments. In addition, E.B. thanks Ryan O'Donnell for much valuable advice throughout the course of this research and Michael Saks for enlightening discussions.

References

[AB10] Noga Alon and Eric Blais. Testing boolean function isomorphism. In *Proc. RANDOM-APPROX*, pages 394–405, 2010.

[AFKS00] Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 20:451–476, 2000. 10.1007/s004930070001.

[AKK+03] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing low-degree polynomials over GF(2). In *Proc. RANDOM-APPROX*, pages 188–199, 2003.

[AS92] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley, New York, 1992.

[BBM11] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing via communication complexity. *Proc. CCC*, 2011.

[BC10] Laszlo Babai and Sourav Chakraborty. Property testing of equivalence under a permutation group action. *To appear in the ACM Transactions on Computation Theory (ToCT)*, 2010.

[BEHL09] Ido Ben-Eliezer, Rani Hod, and Shachar Lovett. Random low degree polynomials are hard to approximate. In *Proc. RANDOM-APPROX*, pages 366–377, 2009.

[BFF+01] T. Batu, L. Fortnow, E. Fischer, R. Kumar, R. Rubinfeld, and P. White. Testing random variables for independence and identity. *Proc. IEEE Symposium on Foundations of Computer Science*, 0:442, 2001.

[Bla09] Eric Blais. Testing juntas nearly optimally. In *Proc. ACM symposium on the Theory of computing*, pages 151–158, New York, NY, USA, 2009. ACM.

[BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, STOC '90, pages 73–83, New York, NY, USA, 1990. ACM.

[BO10] Eric Blais and Ryan O'Donnell. Lower bounds for testing function isomorphism. In *IEEE Conference on Computational Complexity*, pages 235–246, 2010.

[CG04] Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, 90:301–305, June 2004.

[CGM11a] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Efficient sample extractors for juntas with applications. In *Proc. ICALP*, 2011.

[CGM11b] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Nearly tight bounds for testing function isomorphism. In *Proc. SODA*, 2011.

[DGL+99] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Proc. RANDOM-APPROX*, pages 97–108, 1999.

[DLM+07] Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. *Proc. IEEE Symposium on Foundations of Computer Science*, 0:549–558, 2007.

[Fis01] Eldar Fischer. The art of uninformed decisions. *Bulletin of the EATCS*, 75:97, 2001.

[Fis05] Eldar Fischer. The difficulty of testing for isomorphism against a graph that is given in advance. *SIAM J. Comput.*, 34(5):1147–1158, 2005.

[FKR+04] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753 – 787, 2004. Special Issue on FOCS 2002.

[FLN+02] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *STOC*, pages 474–483, 2002.

[FM08] Eldar Fischer and Arie Matsliah. Testing graph isomorphism. *SIAM J. Comput.*, 38(1):207–225, 2008.

[FNS04] Eldar Fischer, Ilan Newman, and Jiří Sgall. Functions that have read-twice constant width branching programs are not necessarily testable. *Random Struct. Algorithms*, 24(2):175–193, 2004.

[FR87] P. Frankl and V. Rödl. Forbidden intersections. *Trans. Amer. Math. Soc. 300*, pages 259–286, 1987.

[FW81] P. Frankl and M. Wilson. Intersection theorems with geometric consequences. *Combinatorica 1*, pages 357–368, 1981.

[GGL+00] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.

[GGR98] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45:653–750, July 1998.

[Gol10] Oded Goldreich. On testing computability by small width obdds. In *APPROX-RANDOM*, pages 574–587, 2010.

[HS69] András Hajnal and Endre Szemerédi. Proof of a conjecture of Paul Erdős. In *Combinatorial Theory and its Applications*, pages 601–623, 1969.

[JPRZ04] Charanjit S. Jutla, Anindya C. Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. *Proc. IEEE Symposium on Foundations of Computer Science*, 0:423–432, 2004.

[Juk01] Stasys Jukna. *Extremal Combinatorics: with applications in computer science.* Springer, 2001.

[KR04] Tali Kaufman and Dana Ron. Testing polynomials over general fields. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 413–422, Washington, DC, USA, 2004. IEEE Computer Society.

[KS05] Peter Keevash and Benny Sudakov. Set systems with restricted cross-intersections and the minimum rank of inclusion matrices. *SIAM J. Discrete Math.*, 18(4):713–727, 2005.

[MORS09a] Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing halfspaces. In *SODA*, pages 256–264, 2009.

[MORS09b] Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing ± 1-weight halfspaces. In *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '09 / RANDOM '09, pages 646–657, Berlin, Heidelberg, 2009. Springer-Verlag.

[PRS02] Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM J. Discrete Math.*, 16(1):20–46, 2002.

[Ron08] Dana Ron. Property testing: A learning theory perspective. *Found. Trends Mach. Learn.*, 1:307–402, March 2008.

[Ron10] Dana Ron. Algorithmic and analysis techniques in property testing. *Found. Trends Theor. Comput. Sci.*, 5:73–205, February 2010.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25:252–271, February 1996.

[RS11] Ronitt Rubinfeld and Asaf Shapira. Sublinear time algorithms. *Electronic Colloquium on Computational Complexity (ECCC)*, 11(013), 2011.

[SSS95] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM J. Discrete Math.*, 8(2):223–250, 1995.

Appendix

A Distinguishing two random functions with $\widetilde{O}(\sqrt{n})$ queries

In light of the fact that two trimmed random functions are hard to distinguish with fewer than n queries (roughly), we may ask whether the restriction to trimmed functions is necessary. In this section we show that without such a restriction, the aforementioned task can be completed with only $\widetilde{O}(\sqrt{n})$ queries. We prove the following proposition, which says in particular that any function can be distinguished from a completely random function using $\widetilde{O}(\sqrt{n})$ queries.

Proposition A.1 *Let $p < 1$ be an arbitrary constant. For* any *function f and* any *distribution $\mathcal{D}_y$ over functions isomorphic to f, it is possible to distinguish $g \in \mathcal{D}_y$ from $g \in U$ with probability at least p using $\widetilde{O}(\sqrt{n})$ queries.*

Note that querying g only on inputs of Hamming weights $0, 1, n-1, n$ is only of limited help. By querying the all-zero and all-one inputs, we can distinguish between the two cases only with probability $3/4$; notice that this success probability cannot be amplified, since the probability is taken over the choice of functions, rather than the randomness of the tester. When considering singletons (and likewise, inputs of weight $n-1$), then f, g are isomorphic only if $|\{|x| = 1 : f(x) = 1\}| = |\{|x| = 1 : g(x) = 1\}|$. So a natural (and only) approach is to test the equality of these measures by sampling. But notice that for most f, with very high probability (over the choice of g), these two measures will be at most $O(\sqrt{n})$ away from each other, which means that distinguishing the two cases requires at least $\Omega(n)$ samples.

We show that $\widetilde{O}(\sqrt{n})$ queries into inputs of weight ≤ 2 are sufficient for distinguishing $g \in \mathcal{D}_y$ from $g \in U$ with high probability. One way to do this is to interpret the restriction of f and g to $\binom{[n]}{2}$ as adjacency functions of graphs on n vertices. It is not hard to prove that for any f and a randomly chosen g, the corresponding graphs G_f, G_g are $1/3$-far from being isomorphic with overwhelming probability. On the other hand, if f is isomorphic to g then G_f is obviously isomorphic to G_g. Hence, we can use the isomorphism tester of [FM08] (in the appropriate setting) to distinguish between the two cases.

But in fact, the graph case is more complicated, since it is concerned with the worst case scenario (i.e., it should work for any pair of graphs). In our case, we only wish to distinguish a (possibly random) permutation of some given f from a random function g. Indeed, it turns out that we can reduce our problem directly to the task of testing equivalence of a samplable distribution to an explicitly given one. Then we can use an algorithm of Batu et al. [BFF$^+$01] that solves exactly this problem with $\widetilde{O}(\sqrt{n})$ queries. We work out the formal details below.

Proof. Let $\ell = 2 \log n$. Given a function $f : \{0,1\}^n \to \{0,1\}$ and $i \in [n]$ we define $\alpha(f, i) \in \{0,1\}^\ell$ as follows: the j'th bit of $\alpha(f, i)$ is one if and only if $f(\{i, j\}) = 1$, where we allow j to range from 1 to ℓ only (rather than the full range of $[n]$). We then define the distribution D_f over $\{0,1\}^\ell$, where the probability of $\beta \in \{0,1\}^\ell$ under D_f is $\frac{1}{n}|\{i \in [n] : \alpha(f, i) = \beta\}|$. Clearly, if $f = g$ then $D_f = D_g$. Now we claim something similar for f and g that are isomorphic.

Let Π be a set of permutations of $[n]$, such that there is one-to-one correspondence between the elements of Π and the possible injections $I : [\ell] \to [n]$ as follows. Each $\pi \in \Pi$ is associated with an injection $I_\pi : [\ell] \to [n]$, such that

$$\pi(i) = \left\{ \begin{array}{lcl} I_\pi(i) & , & i \in [\ell] \\ i & , & i \in [n] \setminus [\ell] \text{ and } I_\pi^{-1}(i) = \emptyset \\ I_\pi^{-1}(i) & , & i \in [n] \setminus [\ell] \text{ and } I_\pi^{-1}(i) \neq \emptyset \end{array} \right\}.$$

Clearly, $|\Pi| \leq n^\ell$.

Claim A.1 *If f is isomorphic to g, then for some $\pi \in \Pi$, $D_f = D_{g^\pi}$. On the other hand, for any function f,*

$$\Pr_g \Big[|D_f - D_{g^\pi}| \leq 1/4 \text{ for some } \pi \in \Pi \Big] = 1 - o(1).$$

Proof. The first statement is straightforward: Let f and g be isomorphic, i.e. $f = g^\sigma$ for some $\sigma : [n] \to [n]$. Take $\pi \in \Pi$ such that $\sigma(i) = \pi(i)$ for all $i \in [\ell]$. Then $D_f = D_{g^\pi}$.

Now, fix f, and let g be chosen uniformly at random. We would like to show that for all $\pi \in \Pi$, $\Pr_g \Big[|D_f - D_{g^\pi}| \leq 1/4 \Big] = 1 - o(1/|\Pi|)$, so that we can apply the union bound. But notice that it is sufficient to prove this inequality when π is the identity, because the function g is chosen uniformly at random.

Fix $i \in [n]$. For every $j \in [n]$,

$$\Pr_g \Big[\alpha(f,i) = \alpha(g,j) \Big] = 2^{-\ell},$$

hence

$$\Pr_g \Big[\alpha(f,i) = \alpha(g,j) \text{ for some } j \in [n] \Big] \leq n 2^{-\ell} = 1/n.$$

Therefore, the expected intersection size between the multisets[16] $\{\alpha(f,i) : i \in [n]\}$ and $\{\alpha(g,i) : i \in [n]\}$ is $O(1)$. But notice that in order for the distributions D_f and D_g to be close, the intersection of these multisets must be of size $\Omega(n)$. Using the fact that the events

$$E_i \triangleq \mathbb{I}\Big[\alpha(f,i) = \alpha(g,j) \text{ for some } j \in [n] \Big]$$

are independent, we can apply standard concentration bounds to conclude that

$$\Pr_g \Big[|D_f - D_g| \leq 1/4 \Big] = 1 - 2^{-\Omega(n)} = 1 - o(1/|\Pi|),$$

completing the proof. □

Notice that the distribution D_f can be constructed exactly given f. On the other hand, given an oracle access to g, we can obtain a random sample from D_g by picking a random $i \in [n]$ and querying g on ℓ inputs $\{i,1\}, \ldots, \{i,\ell\}$. This observation, together with Claim A.1, suggests that we use the following lemma from [BFF$^+$01], which states that $\widetilde{O}(\sqrt{n})$ samples are sufficient for testing equivalence between a samplable distribution and an explicitly given one.

[16] Intersection here can be a multiset as well. For example, $\{a,a,b,c,c,c\} \cap \{a,a,b,b,c\} = \{a,a,b,c\}$.

Lemma A.2 *There is a tester T_{Dist} that for any two distributions D_K,D_U over $\{0,1\}^*$, each having support of size at most n, and where D_K is given explicitly and D_U is given as a black box that allows sampling, satisfies the following: If $D_K = D_U$ then the T_{Dist} accepts with probability at least $1 - n^{-3\log n}$; and if $|D_K - D_U| \geq 1/4$ then T_{Dist} rejects with probability at least $1 - n^{-3\log n}$. In any case, T_{Dist} uses $\widetilde{O}(\sqrt{n})$ samples.*

Actually, this is an amplified version of the lemma from [BFF$^+$01], which can be achieved by independently repeating the algorithm provided there polylog(m) many times and taking the majority vote.

To conclude, we can reduce our problem to testing equivalence of distributions as follows. Given f and oracle access to g, go over all permutations $\pi \in \Pi$ and test, with T_{Dist}, if D_f and D_{g^π} are equal. If T_{Dist} accepts for some π, accept; otherwise reject.

By Claim A.1, if f is isomorphic to g then for some $\pi \in \Pi$ we have $D_f = D_{g^\pi}$, and so T_{Dist} will with high probability accept while checking that particular π. On the other hand, every π for which $|D_f - D_{g^\pi}| \geq 1/4$ is accepted with probability at most $n^{-3\log n} = o(1/|\Pi|)$. Thus, for randomly chosen g, T_{Dist} rejects with probability $1 - o(1)$.

As for the query complexity, the amplified version of Lemma A.2 allows us to reuse the same $\widetilde{O}(\sqrt{n})$ samples for checking all permutations in Π. Therefore, since simulating a random sample from D_{g^π} requires $\ell = 2\log n$ queries to g, the bound on the query complexity is $\widetilde{O}(\sqrt{n})$. □