On Testing Computability by Small Width OBDDs

Oded Goldreich*
Department of Computer Science
Weizmann Institute of Science
Rehovot, ISRAEL.
oded.goldreich@weizmann.ac.il

April 14, 2010

Abstract

We take another step in the study of the testability of small-width OBDDs, initiated by Ron and Tsur (Random'09). That is, we consider algorithms that, given oracle access to a function $f : \{0,1\}^n \to \{0,1\}$, need to determine whether f can be implemented by some restricted class of OBDDs or is far from any such function.

Ron and Tsur showed that testing whether a function $f : \{0,1\}^n \to \{0,1\}$ is implementable by a width-2 OBDD has query complexity $\Theta(\log n)$. Thus, testing width-2 OBDD functions is significantly easier than learning such functions (which requires $\Omega(n)$ queries). We show that such exponential gaps do not hold for several related classes. Specifically:

1. Testing whether $f : \{0,1\}^n \to \{0,1\}$ is implementable by a width-4 OBDD requires $\Omega(\sqrt{n})$ queries.
2. Testing whether $f : \mathrm{GF}(3)^n \to \mathrm{GF}(3)$ is a linear function with 0-1 coefficients requires $\Omega(\sqrt{n})$ queries. Note that this class of functions is a subset of the class of all linear functions over GF(3), and that each such linear function can be implemented by a width-3 OBDD.
3. There exists a subclass $\mathcal{C}$ of the linear functions from $\mathrm{GF}(2)^n$ to GF(2) such that testing membership in $\mathcal{C}$ has query complexity $\Theta(n)$. Note that each linear function over GF(2) can be implemented by a width-2 OBDD.

Recall that each of these classes has a proper learning algorithm of query complexity $O(n)$.

Keywords: Property Testing, Small Width OBDDs,

*Partially supported by the Israel Science Foundation (grants No. 1041/08).

Contents

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a couple of recent surveys [18, 19]). Loosely speaking, property testing typically refers to super-fast probabilistic algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by making suitable queries. The current work belongs to the study of property testing, but pursues somewhat different themes than the standard ones.

1.1 Testing membership in complexity classes

In the foregoing description, objects are viewed as functions, and so properties are sets of functions. Given this perspective, it is most natural to ask whether various *traditional complexity classes* are testable. Interestingly, this question was not addressed till [20].[1] Instead, whenever (before [20]) computational devices were referred to in the context of property testing, the perspective was that each fixed *computational device* defines a set of strings and the testing problem studied was of membership of the input string in this set (cf. [2, 16, 14]). In contrast, following Ron and Tsur [20], we fix a *complexity class* and study the testing problem that refers to whether the input function is in this class.

To illustrate the difference recall that Alon *et al.* [2] fix any regular set, and study the problem of testing whether a given (input) string is in the set. In contrast, Ron and Tsur [20] consider the complexity class of width-2 OBDDs,[2] and study the problem of testing whether a given (input) function belongs to this complexity class.

The main result of [20] is that testing width-2 OBDD has query complexity $\Theta(\log n)$, where n denotes the length of the argument to the function being tested (i.e., the question is whether $f : \{0,1\}^n \to \{0,1\}$ can be implemented by a width-2 OBDD). This should be compared to the query complexity of *learning* this very class, which is $\Theta(n)$. Thus, testing this complexity class is significantly easier than learning this class. Two natural questions arise:

1. What about width-w OBDDs, for any fixed $w > 2$?

   That is, is testing width-w OBDDs significantly easier (i.e., (poly)logarithmically easier) than learning width-w OBDDs? (Recall that learning width-w OBDDs requires $\Omega(n)$ queries, whereas proper learning is possible with $O(n)$ queries.)

2. What about testing subclasses of width-w OBDDs, for any fixed $w \geq 2$ (i.e., testing whether a given function belongs to a fixed subclass of width-w OBDDs)? Specifically, is every subclass of width-2 OBDDs testable in query complexity $O(\log n)$ or $\text{poly}(\log n)$?[3]

We provide rather gloomy answers to both questions: We prove that even at low computational complexity levels such as constant-width OBDDs, testing may not be significantly easier than learning; that is, these complexities are polynomially related rather than being exponentially related. Specifically:

[1] Indeed, this is a controversial statement, which relies on not viewing the classes of dictatorship functions, juntas, monomials, and constant-term DNFs as traditional complexity classes. The testability of these classes was studied in various works; see, for example [17, 9, 6].

[2] OBDDs are ordered binary decision diagrams, which are a restricted type of read-once branching programs in which the variables are read in a fixed order (across all possible computation paths). See definition in Section 1.4.

[3] Note that the query complexity of testing such a subclass need not be smaller that the query complexity of testing the class.

Theorem 1 (see Theorem 4.2): *Testing width-4 OBDD requires $\Omega(\sqrt{n})$ queries.*

We conjecture that the actual query complexity is $\Theta(n)$.

Theorem 2 (see Theorem 2.1): *There exists a subclass of width-2 OBDDs such that testing this subclass requires $\Omega(n)$ queries. Furthermore, this subclass is a class of linear functions* (over GF(2)).

1.2 Subclasses of linear and quadratic functions

A different perspective on our results is best illustrated by a question of Shafi Goldwasser, who asked whether there is more to algebraic property testing than testing low degree. (Actually, this was a rhetorical question; she meant to advocate such studies.) We mention that a clear example of such a study was provided by Rubinfeld [22] in the mid 1990s, and that various properties of polynomials (e.g., dictatorship functions [17], juntas [9, 4], sparse polynomials [6, 7]) were studied in the last decade (although these studies were not viewed from this perspective).

In any case, taking this perspective, we view Theorem 2 as saying that a certain property of linear functions (from $\mathrm{GF}(2)^n$ to GF(2)) cannot be tested significantly faster than learning (i.e., cannot be tested with $o(n)$ queries). More generally, we present a full hierarchy of properties (or classes) of linear functions arranged by their query complexity:

Theorem 3 (see Theorem 2.3): *For every function $t : \mathsf{N} \to \mathsf{N}$ that is at most linear, there exists a property of linear functions* (over GF(2)) *such that testing this property has query complexity $\Theta(t + \epsilon^{-1})$. Furthermore, learning each of the corresponding concept classes requires $\Omega(n)$ queries.*

This leads to the question of how natural are these properties, which build on the property used in the proof of Theorem 2. Since the property is not very natural, we also prove the following.

Theorem 4 (see Theorem 2.7): *Testing the set of linear functions from* $\mathrm{GF}(2)^n$ *to* GF(2) *with at most $n/2$ influential variables requires $\Omega(\sqrt{n})$ queries.*

Here too, we conjecture that the actual query complexity is $\Theta(n)$. Another natural property of linear functions is the subject of the following result.

Theorem 5 (see Theorem 3.2): *Testing the class of linear functions from* $\mathrm{GF}(3)^n$ *to* GF(3) *that have 0-1 coefficients requires $\Omega(\sqrt{n})$ queries.*

Again, we conjecture that the actual query complexity is $\Theta(n)$. (Note that the foregoing class is implemented by width-3 OBDDs.) Lastly, we mention that the proof of Theorem 1 actually establishes also the following.

Theorem 6 (see end of Section 4): *Testing the class of linear functions from* $\mathrm{GF}(2)^n$ *to* GF(2) *that have no consecutive influential variables requires $\Omega(\sqrt{n})$ queries.*

And, again, we conjecture that the actual query complexity is $\Theta(n)$.

1.3 Techniques

The proofs of all the foregoing lower bounds, with the exception of Theorem 2, follow a common theme and cope with a similar difficulty. The common theme is that in all these cases the analysis reduces to upper-bounding the ability of query-bounded observers to distinguish two specific distributions of linear functions. In each case, these two distributions are very natural, and the difficulty is in analyzing the corresponding answer distributions (i.e., the distributions of the sequence of answers obtained by querying each function distribution).

To illustrate the difficulty, consider the set of linear functions from $\mathrm{GF}(2)^n$ to $\mathrm{GF}(2)$, denoted $\mathcal{L}$. It is well known that if f is uniformly distributed in $\mathcal{L}$, then its values on a sequence of t linearly independent vectors are uniformly distributed over $\mathrm{GF}(2)^t$. But it is less clear what happens when f is uniformly distributed in some natural subset $\mathcal{L}' \subset \mathcal{L}$. In particular, what happens when $\mathcal{L}'$ is the set of all linear functions that depend on exactly $n/2$ variables? Furthermore, what if these t strings are selected adaptively?

Our proofs deal with these types of problems. For example, in the case of the set of linear functions that depend on either $(n-1)/2$ or $(n+1)/2$ variables, we prove that the deviation of t non-adaptive queries is at most t/n (cf. Proposition 2.10). For t adaptive queries we only prove an upper bound of $O(t^2/n)$ (cf. Lemma 2.8 and the proof of Theorem 2.7).

1.4 Preliminaries: OBDDs and Property Testing

In this section we review the quite standard definitions used in this paper. We merely stress that when we talk of OBDDs, we assume (as in [20]) that the order of the variables is fixed (and known).

1.4.1 OBDDs: Ordered Binary Decision Diagrams

Several different definitions of this notion appear in the literature, and we adopt the one that calls for a fixed ordering of the variables (knows as "strict"). That is, an ordered binary decision diagram (OBDD) is a read-once branching program in which the order in which the variables are read is fixed for all computing devices in the model. Specifically, we shall assume, without loss of generality, that the i^{th} variable is always read at the i^{th} level. This yields the following definition.

Definition 7 *An* OBDD *is a directed acyclic graph with labeled edges and marked sinks that satisfies the following conditions:*

1. *The graph contains a single source vertex.*
2. *Each sink vertex in the graph is marked either 0 or 1.*
3. *Each non-sink vertex has two out-going edges* (which may be parallel) *one labeled 0 and the other labeled 1.*
4. *The graph edges connect vertices in consecutive levels, where the* level *of a vertex is its distance from the source.*
5. *All sinks have the same level, called the graph* length.

The width *of an OBDD is the maximum number of vertices that have the same level. An OBDD of length n computes the function $f : \{0,1\}^n \to \{0,1\}$ such that, for every $x \in \{0,1\}^n$ it holds that the sink that is reached from the source by following the path with edge labels x is marked $f(x)$.*

Indeed, we may view $x = x_1 \cdots x_n$ as a sequence of variables, and observe that in the i^{th} step (i.e., when moving from the $i-1^{\text{st}}$ level to the i^{th} level) the OBDD branches according to the value of x_i.

We mention that in a subsequent work, Ron and Tsur [21] considered OBDDs with a variable ordering of the variables. Indeed, in such a case, one should specify the ordering, and in more general models that allow different variables to be queried along different computation paths it is necessary to specify the variable queried at each non-sink vertex (by marking the non-sink vertices with variable names).

1.4.2 Property testing

We merely recall the standard definition.

Definition 8 *Let* $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$*, where* Π_n *contains functions defined over the domain* D_n (and range R_n)*. A* tester for a property Π *is a probabilistic oracle machine* T *that satisfies the following two conditions:*

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

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

Almost all our results are lower bounds on the query complexity of property testing tasks, and they are obtained for fixed values of the proximity parameter ϵ (i.e., $\epsilon = 1/16$ will do in all). In these cases we omit mention of the proximity parameter.

2 Testing Subclasses of Width 2 OBDDs

We consider various subclasses of linear functions over GF(2), which in particular are realizable by width-2 OBDDs. For a set of strings $S \subseteq \{0,1\}^n$ we denote by $\mathcal{L}_S$ the set of linear functions $\{f_v : v \in S\}$, where $f_v : \{0,1\}^n \to \{0,1\}$ satisfies $f_v(x) \stackrel{\text{def}}{=} \langle v, x\rangle = \sum_{i=1}^n v_i x_i \bmod 2$.

We present a hierarchy of properties of linear functions arranged according to the query complexity of testing them. Our starting point is a property of linear functions having maximal query complexity, and the hierarchy can be derived using any such property. (This is indeed reminiscent of [11].) After establishing the said hierarchy (and since it refers to somewhat unnatural properties), we also consider the natural property of linear function having a bounded number of influential variables.

2.1 A hierarchy of classes of linear functions

We start by presenting a class of linear functions that is hard to test, and then exhibit the full hierarchy by combining any such class with the class of all linear functions.

2.1.1 Linear functions with coefficients from a small-bias space

Let $S \subset \{0,1\}^n$ be a small bias sample space [15, 1], say, of size $2^{0.99n}$ and bias $2^{-0.3n}$. Then, testing $\mathcal{L}_S$ requires $\Omega(n)$ queries, even if we allow two-sided error and adaptive testers. More generally, we have the following.

Theorem 2.1 (Theorem 2, restated): *Let $S \subset \{0,1\}^n$ be a δ-bias sample space; that is, for every $c \in \{0,1\}^n \backslash \{0^n\}$, it holds that $|\Pr_{v \in S}[\langle c,v\rangle = 1] - 0.5| \leq \delta$. Then, testing $\mathcal{L}_S$ requires $\log_2((1-\rho)/3\delta)$ queries, where $\rho = |S|/2^n$.*

Typically (e.g., in the following example), ρ is small (i.e., $\rho \leq 1/2$), and so the lower bound simplifies to $\log_2(1/6\delta)$. An appealing example consists of the set of all n-bit long strings having a number of 1 that is a multiple of 3 (i.e., $S = \{v \in \{0,1\}^n : \sum_{i=1}^n v_i \equiv 0 \pmod 3\}$), which has exponentially small bias and density $\approx 1/3$ (see Proposition A.1). Thus, we get

Corollary 2.2 *Let S be the set of all n-bit strings having a number of 1-entries that is divisible by three. Then, testing $\mathcal{L}_S$ requires $\Omega(n)$ queries.*

Proof of Theorem 2.1: The theorem follows by combining the following two observations.

1. A random linear function is unlikely to be in $\mathcal{L}_S$, and thus is 0.5-far from $\mathcal{L}_S$. Specifically, with probability $1-\rho$, a random linear function is 0.5-far from $\mathcal{L}_S$.
2. A random linear function and a function uniformly selected in $\mathcal{L}_S$ cannot be distinguished with $\log_2(1/\delta) - O(1)$ queries. Specifically, distinguishing these two distributions with a gap of δ' requires $\log_2(\delta'/\delta)$ queries. This holds because for every sequence of queries and every sequence of potential answers, the probability that this specific answer sequence occurs under a function selected uniformly in $\mathcal{L}_S$ deviates by at most δ from the corresponding probability that refers to a random linear function (see Item 1 of Lemma A.4).

Now, on the one hand, the probability that a tester accepts a random linear function is at most $\rho \cdot \mu + (1-\rho) \cdot \frac{1}{3}$, where $\mu \geq \frac{2}{3}$ denotes the probability that the test accepts a function uniformly distributed in $\mathcal{L}_S$. (Indeed, we assume here that $\epsilon < 1/2$, which implies that the tester accepts linear functions that are not in $\mathcal{L}_S$ with probability at most 1/3). On the other hand, if the test distinguishes random linear functions from functions in $\mathcal{L}_S$ with gap at most δ', then it must accept a random linear function with probability at least $\mu - \delta'$. We infer that $(1-\rho)(\mu - (1/3)) \leq \delta'$, which implies $\delta' \geq (1-\rho)/3$. Combing this with the query lower bound of $\log_2(\delta'/\delta)$, the claim follows. ■

2.1.2 The Hierarchy

The following hierarchy theorem follows by combining any set of hard-to-test linear functions (from $\mathrm{GF}(2)^t$ to $\mathrm{GF}(2)$) with the class of all linear functions (from $\mathrm{GF}(2)^{n-t}$ to $\mathrm{GF}(2)$).

Theorem 2.3 (Theorem 3, restated): *For every function $t : \mathsf{N} \to \mathsf{N}$ that is at most linear, there exist sets $S \subseteq \{0,1\}^n$ such that testing $\mathcal{L}_S$ has query complexity $\Theta(t + \epsilon^{-1})$. Furthermore, learning $\mathcal{L}_S$ requires $\Omega(n)$ queries.*

Proof: Letting $t = t(n)$, we start with an arbitrary set $H \subset \{0,1\}^t$ such that $\mathcal{L}_H$ is a property of linear functions from $\mathrm{GF}(2)^t$ to $\mathrm{GF}(2)$ that requires $\Omega(t)$ queries for testing. Indeed, such a property is provided by Corollary 2.2. Next, we consider an arbitrary set $G \subseteq \{0,1\}^{n-t}$ such that $\mathcal{L}_G$ is a property of linear functions from $\mathrm{GF}(2)^{n-t}$ to $\mathrm{GF}(2)$ that can be tested in $O(1/\epsilon)$ queries (with one-sided error) but requires $\Omega(n-t)$ queries for learning. Indeed, the set $S = \{0,1\}^{n-t}$ will do (and other alternatives are provided by Theorem 2.4). Combining these two properties, we consider the set $S = H \times G$, and the corresponding property $\mathcal{L}_S$. Note that each $f \in \mathcal{L}_S$ can be written as the sum of some $h \in \mathcal{L}_H$ and some $g \in \mathcal{L}_G$ such that

$$f(x_1, ..., x_t, x_{t+1}, ..., x_n) = h(x_1, ..., x_t) + g(x_{t+1}, ..., x_n). \tag{1}$$

Learning $\mathcal{L}_S$ requires $\Omega(n)$ queries, since recovering f requires recovering both h and g. Formally, we can reduce learning h (resp., g) to learning f, by fixing g (resp., h). Similarly (i.e., by fixing g (resp., h)), we can reduce testing $\mathcal{L}_H$ (resp., $\mathcal{L}_G$) to testing $\mathcal{L}_S$, and conclude that the query complexity of the latter task is $\Omega(t+\epsilon^{-1})$. It is thus left to show that $\mathcal{L}_S$ can be tested in $O(t+\epsilon^{-1})$ queries. This is shown by presenting an algorithm that, on input n and proximity parameter $\epsilon > 0$, proceeds as follows.

1. *Testing if f is linear*: The algorithm repeats the basic BLR Test for $O(1/\epsilon)$ times, where in each repetition the algorithm selects uniformly $a, b \in \mathrm{GF}(2)^n$, and rejects if $f(a) + f(b) \neq f(a+b)$. The algorithm continues to the next steps only if none of these checks has rejected, and so we will assume in these steps that f is ϵ-close to linear.

   Let $h : \mathrm{GF}(2)^t \to \mathrm{GF}(2)$ and $g : \mathrm{GF}(2)^{n-t} \to \mathrm{GF}(2)$ be linear functions such that $h(x_1..., x_t) + g(x_{t+1}, ..., x_n)$ is the linear function closest to $f(x_1..., x_t, x_{t+1}, ..., x_n)$.

2. *Reconstructing the function h*: Using $O(t)$ queries, the algorithm reconstructs h; by using self-correction, see details bellow. The algorithm rejects if $h \notin \mathcal{L}_H$.

   For starters, consider a naive algorithm that recovers each coefficient of h with success probability at least $1 - (1/10n)$ by making $O(\log n)$ queries. Specifically, for every $i \in [t]$, the i^{th} coefficient is reconstructed by taking a majority vote of $O(\log n)$ experiments, where in each experiment we select uniformly $a \in \mathrm{GF}(2)^n$, and compute $f(a) + f(a + 0^{i-1}10^{n-i})$. Below, we shall describe a more efficient reconstruction procedure, which uses $O(t)$ queries rather than $O(t \log n)$ queries.

3. *Testing the residual function g*: Actually, for a random $a = (a_1, ..., a_t) \in \mathrm{GF}(2)^t$, the algorithm tests whether the residual function f_a defined as $f_a(x_{t+1}, ..., x_n) = f(a_1, ..., a_t, x_{t+1}, ..., x_n) - h(a)$ belongs to $\mathcal{L}_G$. This is done by using the tester of $\mathcal{L}_G$.

We first observe that this algorithm accepts any $f \in \mathcal{L}_S$ with probability 1, since $f = h + g$ passes the linearity test (of Step 1) with probability 1, Step 2 always reconstructs h, and Step 3 always accepts g (assuming that the tester of $\mathcal{L}_G$ has one-sided error). Thus, we turn to analyze the behavior of this algorithm when f is ϵ-far from $\mathcal{L}_S$.

We may assume that f is ϵ-close to being linear, since otherwise Step 1 rejects with high constant probability (say, probability at least 2/3). Considering h and g as defined at the end of Step 1, we note that either $h \notin \mathcal{L}_H$ or $g \notin \mathcal{L}_G$. In the first case (i.e., $h \notin \mathcal{L}_H$) Step 2 rejects with high probability, since (with high probability) the reconstructed function will be h. In the second case, we consider for every $a = (a_1, ..., a_t) \in \mathrm{GF}(2)^t$, the linear function that is closest to f_a (where $f_a(x_{t+1}, ..., x_n) = f(a_1, ..., a_t, x_{t+1}, ..., x_n) - h(a)$), and note that for at least $1 - 4\epsilon$ of the choices of $a \in \mathrm{GF}(2)^t$ this linear function equals g (since f is ϵ-close to $h + g$).[4] Assuming that $\epsilon \leq 0.01$ (or

[4] For a uniformly distributed a, the expected relative distance of f_a from g is at most ϵ. If f_a is closer to some linear function other than g, then its relative distance to g must be at least 1/4.

else we reset $\epsilon \leftarrow 0.01$), we infer that Step 3 rejects with probability at least $0.96 \cdot 0.9 > 2/3$, where we assume (without loss of generality) that the $\mathcal{L}_G$-tester has error probability at most 0.1.

It is left to provide a more efficient implementation of Step 2. Indeed, instead of recovering each coefficient of h with error probability of $1/10n$, we recover each bit in the "encoding of h's coefficients" (via a good linear error-correcting code) with probability at least 0.9, and obtain h by using an error-correcting decoder. Specifically, we use a good linear error-correcting code $C : \mathrm{GF}(2)^t \rightarrow \mathrm{GF}(2)^T$, where $T = O(t)$, and let $\ell_1, ..., \ell_T : \mathrm{GF}(2)^t \rightarrow \mathrm{GF}(2)$ denote the corresponding linear functions; that is, $C(z) = \ell_1(z) \cdots \ell_T(z)$. Viewing each ℓ_i as an element of $\mathrm{GF}(2)^t$, we obtain $h(\ell_i)$ via self-correction; that is, we select uniformly $a \in \mathrm{GF}(2)^n$, and compute $f(a) + f(a + \ell_i 0^{n-t})$. Thus, we obtain each $h(\ell_i)$, which is a linear combination of h's coefficients, with probability at least $1 - 2\epsilon > 0.9$, and by using error correction this yields the values of $h(10^{t-1}), ..., h(0^{t-1}1)$ (with overwhelmingly high probability).[5] ■

2.1.3 Linear functions in a fixed linear space

Recall that the standard linearity property (i.e., the set of all linear functions over GF(2)) is testable by $O(1/\epsilon)$ non-adaptive queries. Here we point out that this is not the only property of linear functions having $\Theta(1/\epsilon)$ testing complexity, but is merely a special case of a larger class of properties. Specifically, we consider arbitrary classes $\mathcal{L}_S$ such that S is a linear space. That is, let $S = \{Gs : s \in \{0,1\}^k\}$, where G is an k-dimensional generator matrix. Thus, for every $s \in \{0,1\}^k$, we define the function $g_s \in \mathcal{L}_S$ as $g_s(x) = f_{Gs}(x) = \langle Gs, x\rangle$, and note that $\langle Gs, x\rangle = \langle s, G^\top x\rangle$.

Theorem 2.4 *Let $S \subseteq \{0,1\}^n$ be a linear space, and $\mathcal{L}_S = \{f_v : v \in S\}$. Then, $\mathcal{L}_S$ can be tested with $O(1/\epsilon)$ non-adaptive queries.*

Proof: The case of $S = \{0,1\}^n$ corresponds to linearity testing, which is handled by the BLR linearity test [5], and so we focus on the case that $S \subset \{0,1\}^n$. We actually present a proximity-oblivious tester (cf. [13]). When given oracle access to a function for f, we perform the following two checks.

1. *BLR Linearity Check*: Uniformly select $a, b \in \{0,1\}^n$, and reject if $f(a) + f(b) \neq f(a + b)$.

2. *Checking* (via self correcting) *that the kernel of $G^\top$ evaluates to zero*: Uniformly select $a \in \{0,1\}^n$ and $b \in \{x : G^\top x = G^\top a\}$, and reject if $f(a) \neq f(b)$. (This is a self-correction of checking for a random $c \in \{x : G^\top x = 0\}$ whether $f(c) = 0$.)

The test accept only if none of the foregoing checks rejected. Clearly, any $f \in \mathcal{L}_S$ passes both checks with probability 1. Thus, we focus on analyzing the probability that a function $f \notin \mathcal{L}_S$ is rejected, denoting by δ the distance of f to the set of all linear functions.

We first note that f is rejected by the first check with probability at least δ (cf. [3]). Denoting the linear function closest to f by g, we note that if $g \notin \mathcal{L}_S$ then there exists x such that $G^\top x = 0$ and $g(x) \neq 0$, since otherwise g is constant on each set $S_\alpha \stackrel{\rm def}{=} \{x : G^\top x = \alpha\}$ and it follows that $g(x)$ is linear in $G^\top x$ (since g is linear and only depends on $G^\top x$). Furthermore, at most half of the kernel of G evaluates to 0 under g, since these vectors form a subgroup. Thus, in this case (i.e.,

[5]Indeed, our reasoning interchanges the roles of function and argument between h and its argument, but recalling that h is linear it is actually the case that the roles of function and argument are ficticious, when we associated the linear function h with its coefficient sequence, denoted u. Indeed, if $h(z) = \langle u, z\rangle = \sum_{i=1}^t u_i z_i \bmod 2$, then u and z actually play the same role. Our reconstruction of the bits of u, viewed as $h(10^{t-1}), ..., h(0^{t-1}1)$, by obtaining a noisy version of $C(u) = \ell_1(u) \cdots \ell_T(u)$, where each $\ell_i(u)$ equals $\langle u, \ell_i\rangle = h(\ell_i)$.

$g \notin \mathcal{L}_S$), the second check rejects with probability at least $0.5 - 2 \cdot \delta$. It follows that if f is ϵ-far from $\mathcal{L}_S$, then it is rejected with probability at least $\min(\epsilon, 1/6) \geq \epsilon/6$, where the first term is due to the case that $g \in \mathcal{L}_S$ (since in this case f is rejected with probability at least $\delta \geq \epsilon$) and the second term is due to the case that $g \notin \mathcal{L}_S$ (since in this case f is rejected with probability at least $\max(\delta, 0.5 - 2\delta) \geq 1/6$). ■

2.2 Linear functions with at most ρn influential variables

For any constant $\rho > 0$, let W_ρ denote the class of linear functions with at most ρn influential variables. That is, $W_\rho = \mathcal{L}_S$ for $S = \{v : \mathrm{wt}(v) \leq \rho n\}$, where $\mathrm{wt}(v) = |\{i : v_i = 1\}|$.

Conjecture 2.5 *Testing $W_{0.5}$ requires $\Omega(n)$ queries, even when allowing adaptive testers of two-sided error.*

If true, then (by using techniques as in the proof of Theorem 2.3) it will follow that, for any function $\rho : \mathsf{N} \to [0,1]$, testing W_ρ requires $\Omega(\rho(n) \cdot n)$ queries. We present two partial results that support Conjecture 2.5: the first is an $\Omega(n)$ lower bound for non-adaptive testers and the second is an $\Omega(\sqrt{n})$ lower bound for general (adaptive) testers. In particular, this establishes Theorem 4.

2.2.1 Linear lower bound for non-adaptive testers

We show that Conjecture 2.5 holds when restricted to non-adaptive testers.

Proposition 2.6 *Testing $W_{0.5}$ requires $\Omega(n)$ non-adaptive queries, even when allowing two-sided error.*

Proof: We consider two classes of linear functions, denoted GOOD and BAD, such that GOOD $\subset W_{0.5}$, whereas BAD $\cap\, W_{0.5} = \emptyset$, which implies that every function in BAD is 0.5-far from $W_{0.5}$. For $m = n/2$, each of these functions will be specified by an index $j_0 \in [m]$ and a sequence of m bits $\sigma_1, ..., \sigma_m \in \{0,1\}$, Specifically, we let $g_{j_0,\sigma_1,...,\sigma_m}$ denote the linear function f_v such that

$$v = \sigma_1 \overline{\sigma}_1 \cdots \sigma_{j_0-1} \overline{\sigma}_{j_0-1} 00 \sigma_{j_0+1} \overline{\sigma}_{j_0+1} \cdots \sigma_m \overline{\sigma}_m, \tag{2}$$

and let GOOD $= \{g_{j_0,\sigma_1,...,\sigma_m} : j_0 \in [m], \sigma_1, ..., \sigma_m \in \{0,1\}\}$. Similarly, we let $b_{j_0,\sigma_1,...,\sigma_m}$ denote the linear function f_v such that

$$v = \sigma_1 \overline{\sigma}_1 \cdots \sigma_{j_0-1} \overline{\sigma}_{j_0-1} 11 \sigma_{j_0+1} \overline{\sigma}_{j_0+1} \cdots \sigma_m \overline{\sigma}_m, \tag{3}$$

and let BAD $= \{b_{j_0,\sigma_1,...,\sigma_m} : j_0 \in [m], \sigma_1, ..., \sigma_m \in \{0,1\}\}$. Note that

$$g_{j_0,\sigma_1,...,\sigma_m}(x) = \sum_{j \neq j_0} (\sigma_j x_{2j-1} + (1-\sigma_j) x_{2j}) \tag{4}$$

$$b_{j_0,\sigma_1,...,\sigma_m}(x) = x_{2j_0-1} + x_{2j_0} + \sum_{j \neq j_0} (\sigma_j x_{2j-1} + (1-\sigma_j) x_{2j}) \tag{5}$$

and that each term in these sums equals $(x_{2j-1} + x_{2j})\sigma_j + x_{2j}$. That is, the value of a generic $g_{j_0,\sigma_1,...,\sigma_m}$ at a query $q \in \{0,1\}^n$ equals $\sum_{j \neq j_0} (q_{2j-1} + q_{2j})\sigma_j + \sum_{j \neq j_0} q_{2j}$.

Note that elements of GOOD can be distinguished from elements of BAD by using $O(\log n)$ *adaptive* queries. Specifically, every query of the form $q_1 \cdots q_n \in \{00, 11\}^m$ is answered by $\sum_{j \neq j_0} q_{2j}$, which allows finding j_0 by a binary search (since $j_0 \in \{j \in [m] : q_{2j} = 1\}$ if and only if the answer to the query $q_1\cdots q_n\in\{00,11\}^m$ differs from $\sum_{j\in[m]}q_{2j}$). Needless to say, once j_0 is found, we distinguish any $g_{j_0,\cdot}$ from any $b_{j_0,\cdot}$ by the query $q=0^{2j_0-1}10^{n-2j_0}$ (since $g_{j_0,\sigma}(q)=0$ whereas $b_{j_0,\sigma}(q)=1$).

Our aim is to prove that $\Omega(n)$ *non-adaptive* queries are required in order to distinguish, with constant probability gap, between a uniformly selected element of GOOD and a uniformly selected element of BAD. Recall that an element in either sets is selected by specifying an index $j_0\in[m]$ and an m-bit string. Fixing any sequence of queries $\overline{q}=(q^{(1)},...,q^{(t)})$, we shall show that for most choices of $j_0\in[m]$ the answers to $\overline{q}$ are distributed identically in the two distributions. The exceptional indices j_0 are called special and defined next.

Definition 2.6.1 *An index $j\in[m]$ is called* special *with respect to a sequence of queries $\overline{q}=(q^{(1)},...,q^{(t)})$ if there exists a linear combination of these queries that yields an n-bit string q such that $q\in\{00,11\}^{j-1}\times\{01,10\}\times\{00,11\}^{m-j}$.*

It will be convenient to use matrix notation in our analysis. We present $\overline{q}$ as a matrix, denoted Q, such that the $i^{\rm th}$ row of Q equals $q^{(i)}$. The condition in Definition 2.6.1 asserts that there exists a t-vector v such that $q=vQ$ is in $\{00,11\}^{j-1}\times\{01,10\}\times\{00,11\}^{m-j}$. Denoting by I_2 an n-by-m binary matrix in which the (i,j) entry is 1 if and only if $j=\lceil i/2\rceil$ (i.e., I_2 maps the row vector $q_1\cdots q_n$ to $p_1\cdots p_m$ such that $p_k=q_{2k-1}+q_{2k}$), the latter condition means that qI_2 is the $j^{\rm th}$ unit vector (i.e., the vector $0^{j-1}10^{m-j}$). Using this observation, we immediately get

Claim 2.6.2 *For any sequence of t queries, $\overline{q}$, there exists at most t indices that are special with respect to $\overline{q}$.*

Proof: For every special index j, there exists a t-vector v such that $vQI_2=0^{j-1}10^{m-j}$. Thus, the number of special indices is a lower bound on the rank of the matrix Q, which is upper bounded by t. □

Claim 2.6.3 *Suppose that j_0 is not special with respect to $\overline{q}=(q^{(1)},...,q^{(t)})$. Then, when $\sigma=(\sigma_1,...,\sigma_m)$ is selected uniformly in $\{0,1\}^m$, the t-tuple $(g_{j_0,\sigma}(q^{(1)}),...,g_{j_0,\sigma}(q^{(t)}))$ is distributed identically to the t-tuple $(b_{j_0,\sigma}(q^{(1)}),...,b_{j_0,\sigma}(q^{(t)}))$.*

Proof: Let I_2' be as I_2 except that the $j_0^{\rm th}$ column is all zeros. Then the value of $g_{j_0,\sigma}$ at any query q (i.e., $\sum_{j\neq j_0}(q_{2j-1}+q_{2j})\sigma_j+\sum_{j\neq j_0}q_{2j}$) can be written as $\langle qI_2',\sigma\rangle+\langle qI_1',1^m\rangle$, where I_1' is an n-by-m binary matrix in which the (i,j) entry is 1 if and only if $i=2j$ and $j\neq j_0$. Likewise, the value of $b_{j_0,\sigma}$ at q is written as $\langle qI_2',\sigma\rangle+\langle qI_1',1^m\rangle+q_{2j_0-1}+q_{2j_0}$, where $q_{2j_0-1}+q_{2j_0}=\langle q(I_2-I_2'),1^m\rangle$. That is, in both cases, the randomness comes from the first term; that is, $\langle qI_2',\sigma\rangle=qI_2'\sigma^\top$, since q is fixed and only σ is random (i.e., it is uniformly distributed in $\{0,1\}^m$). Looking at the entire vector of answers, we have

$$(g_{j_0,\sigma}(q^{(1)}),...,g_{j_0,\sigma}(q^{(t)}))^\top = QI_2'\sigma^\top+QI_1'1^m \qquad (6)$$

$$(b_{j_0,\sigma}(q^{(1)}),...,b_{j_0,\sigma}(q^{(t)}))^\top = QI_2'\sigma^\top+Q(I_1'+I_2-I_2')1^m \qquad (7)$$

where, again, the first term is random and the second term is fixed (but different in the two cases). Our goal is to show that these two vectors of answers are identically distributed.

Considering the matrix Q, we fix an arbitrary maximal set of rows such that for corresponding (generalized) submatrix Q' it holds that $Q'I_2'$ is of full rank, denote t'. (For simplicity, suppose that Q' consists of the first t' rows of Q.) Note that QI_2' has rank t', whereas Q may have rank $t\geq t'$.

We first observe that in both distributions, the corresponding t' answers are uniformly distributed in $\{0,1\}^{t'}$, since $Q'I_2'\sigma^\top \in \{0,1\}^{t'}$ is uniformly distributed. As for each of the other rows, denoted q, it holds that qI_2' is a linear combination of the rows of $Q'I_2'$; that is, $qI_2' = u'Q'I_2'$ for some t'-vector u'. (Again, note that q need not equal $u'Q'$.) The key observation (to be proved below) is that $\langle q(I_2 - I_2'), 1^m\rangle = q_{2j_0-1} + q_{2j_0}$ is obtained by the same linear combination (i.e., u') of the corresponding $(q^{(i)}_{2j_0-1} + q^{(i)}_{2j_0})_{i\in[t']}$; that is, $\langle q(I_2 - I_2'), 1^m\rangle$ equals $\langle u'Q'(I_2 - I_2'), 1^m\rangle$. It follows that

$$\begin{aligned}
g_{j_0,\sigma}(q) &= qI_2'\sigma^\top + qI_1'1^m &(8)\\
&= u'Q'I_2'\sigma^\top + qI_1'1^m &(9)\\
&= u'(g_{j_0,\sigma}(q^{(1)}), ..., g_{j_0,\sigma}(q^{(t')}))^\top - u'Q'I_1'1^m + qI_1'1^m &(10)
\end{aligned}$$

where the second equality uses $qI_2' = u'Q'I_2'$. Similarly,

$$\begin{aligned}
b_{j_0,\sigma}(q) &= qI_2'\sigma^\top + qI_1'1^m + q(I_2 - I_2')1^m &(11)\\
&= u'Q'I_2'\sigma^\top + u'Q'(I_2 - I_2')1^m + qI_1'1^m &(12)\\
&= u'(b_{j_0,\sigma}(q^{(1)}), ..., b_{j_0,\sigma}(q^{(t')}))^\top - u'Q'I_1'1^m + qI_1'1^m &(13)
\end{aligned}$$

where the second equality uses both $qI_2' = u'Q'I_2'$ and $u'Q'(I_2 - I_2') = q(I_2 - I_2')$. Thus, both $g_{j_0,\sigma}(q)$ and $b_{j_0,\sigma}(q)$ are obtained by the same linear transformation (i.e., $x^\top \mapsto u'x^\top + \langle (q - u'Q')I_1', 1^m\rangle$) on the corresponding $(g_{j_0,\sigma}(q^{(1)}), ..., g_{j_0,\sigma}(q^{(t')}))$ and $(b_{j_0,\sigma}(q^{(1)}), ..., b_{j_0,\sigma}(q^{(t')}))$, which in turn are identically distributed.

Thus, it is left to prove that $u'Q'(I_2 - I_2') = q(I_2 - I_2')$. Assume, towards the contradiction that $q(I_2 - I_2') \neq u'Q'(I_2 - I_2')$, which implies $(q - u'Q')(I_2 - I_2') \neq 0^m$. On the other hand, recall that $qI_2' = u'Q'I_2'$ (i.e., $(q - u'Q')I_2' = 0^m$), which implies that $(q - u'Q')I_2 = (q - u'Q')(I_2 - I_2')$ is non-zero and hence equals $0^{j_0-1}10^{m-j_0}$ (since the image of $I_2 - I_2'$ is in $\{0^{j_0-1}\sigma 0^{m-j_0} : \sigma \in \{0,1\}\}$). Denoting by i ($i > t'$) the row index of q in Q, note that $v = u'0^{i-t'-1}10^{t-i}$ satisfies $vQ = u'Q' + q$ and so $vQI_2 = (q - u'Q')I_2 = 0^{j_0-1}10^{m-j_0}$. But this (i.e., the fact that QI_2 spans $0^{j_0-1}10^{m-j_0}$) contradicts the hypothesis that j_0 is not special with respect to $\overline{q}$. □

Combining the claims, we conclude that the probability gap observed by a query sequence $\overline{q}$ is upper-bounded by the probability that j_0 is special with respect to $\overline{q}$. ■

2.2.2 A square root lower bound for adaptive testers

For general (adaptive) testers, we prove a lower bound that is weaker than the one in Conjecture 2.5.

Theorem 2.7 (Theorem 4, restated): *Testing $W_{0.5}$ requires $\Omega(\sqrt{n})$ queries, even when allowing adaptive testers of two-sided error.*

Recalling that the (structured) distributions used in the proof of Proposition 2.6 can be distinguished by $O(\log n)$ adaptive queries, we consider instead random permutations of the strings in both distributions. This destroys the structure used by the aforementioned adaptive distinguisher, and yields a proof of Theorem 2.7. The key to the proof is provided by the following lemma, which is of independent interest.

Lemma 2.8 *Let $t < \sqrt{n}/6$ and let Q be a t-by-n full rank matrix such that its rows do not span the vector 1^n. Suppose that v is uniformly distributed among all n-bit binary vectors having weight*

$m = n/2$. Then, with probability at least $1 - (18t^2/n)$, the vector Qv is uniformly distributed over $\{0,1\}^t$; that is, there exists a set G that is a subspace of the probability space Ω that underlies the choice of v (i.e., $v = v(\omega) \in \{0,1\}^n$ for every $\omega \in \Omega$) *such that*

1. $|G| \geq (1 - (18t^2/n)) \cdot |\Omega|$.
2. *For every $\alpha \in \{0,1\}^t$, it holds that $\Pr_{\omega \in G}[Qv = \alpha] = 2^{-t}$, where $v = v(\omega)$.*

Furthermore, if G' is a set as guaranteed for the matrix Q' obtained by omitting a row of Q, then there exists a set $G \subseteq G'$ that satisfies the foregoing conditions with respect to Q.

Note that the requirement that Q is full rank and does not span 1^n is essential; specifically, for any v of weight m it holds that $\langle 1^n, v\rangle = m \bmod 2$.

Proof: We view the uniform distribution over $\{v \in \{0,1\}^n : \mathrm{wt}(v) = m\}$ as generated by the following two-step random process:

1. Select uniformly a partition π of $[n]$ into m ordered pairs, let $\pi(j)$ denote the j^{th} pair, and $\pi_1(j)$ (resp., $\pi_2(j)$) denote the first (resp., second) element of the j^{th} pair (i.e., $\pi(j) = (\pi_1(j), \pi_2(j))$).
2. Select uniformly a string $v = (v_1, ..., v_n) \in \{0,1\}^n$ such that $v_{\pi_1(j)} = 1 - v_{\pi_2(j)}$ holds for every $j \in [m]$. That is, we select uniformly $\sigma = (\sigma_1, ..., \sigma_m) \in \{0,1\}^m$ and determining v such that $v_{\pi_1(j)} = \sigma_j$ (and $v_{\pi_2(j)} = 1 - \sigma_j$).

For π as selected in Step 1 (and the corresponding π_1, π_2), we let I'_π (resp., I''_π) be an n-by-m binary matrix such that entry (i,j) in I'_π (resp., I''_π) equals 1 if and only if $i = \pi_1(j)$ (resp., $i = \pi_2(j)$). Then, for v and σ as above, it holds that $v = I'_\pi\sigma + I''_\pi(1^m + \sigma)$, which implies that

$$Qv = QI_\pi\sigma + QI''_\pi 1^m \tag{14}$$

where $I_\pi = I'_\pi + I''_\pi$. Noting that $QI''_\pi 1^m$ is a fixed vector, it follows that the deviation of Qv from the uniform distribution over $\{0,1\}^t$ equals the deviation of $QI_\pi\sigma$ from the uniform distribution. Lastly, the latter distance is upper-bounded by the probability that QI_π is not full rank. The rest of the proof is devoted to upper-bounding this probability.

We upper-bound the probability that QI_π is not full rank by the sum taken over all $c \in \{0,1\}^t \setminus \{0^t\}$ of the probability that cQI_π equals the all-zero vector. Recall that, by the hypothesis, the vector cQ is neither the all-zero vector nor the all-one vector. Furthermore, when we vary c in $\{0,1\}^t \setminus \{0^t\}$ and consider any t linearly independent columns of Q, we see all possible $2^t - 1$ non-zero patterns. It follows that, for every $k \in [t]$, the cardinality of $\{c \in \{0,1\}^t \setminus \{0^t\} : \mathrm{wt}(cQ) \leq k\}$ is upper-bounded by $\sum_{i\in[k]} \binom{t}{i}$. Similarly, for every $k \in [t]$, the cardinality of $\{c \in \{0,1\}^t \setminus \{0^t\} : n - \mathrm{wt}(cQ) \leq k\}$ is upper-bounded by $1 + \sum_{i\in[k]} \binom{t}{i}$, where the added 1 is due to the case that the pattern 1^t appears in these k columns (but even then $cQ \neq 1^n$). Hence, for every $k \in [t]$:

$$|\{c \in \{0,1\}^t \setminus \{0^t\} : \min(\mathrm{wt}(cQ), n - \mathrm{wt}(cQ)) \leq k\}| \;\leq\; 1 + 2\sum_{i\in[k]} \binom{t}{i}; . \tag{15}$$

Next, fixing any $c \in \{0,1\}^t \setminus \{0^t\}$, we upper-bound the probability that cQI_π is all-zeros. Note that cQI_π is all-zeros if and only if all pairs in the partition π are "monochromatic" (i.e., for every $j \in [m]$ it holds that the $\pi_1(j)^{\text{th}}$ and $\pi_2(j)^{\text{th}}$ positions in cQ have the same value, where

$\pi(j) = (\pi_1(j), \pi_2(j)))$. Letting $w = \mathrm{wt}(cQ)$, and denoting by #pairs(x) the number of partitions of x elements to pairs, we have

$$\Pr_\pi[cQI_\pi = 0^n] \;=\; \frac{\#\mathrm{pairs}(w) \cdot \#\mathrm{pairs}(n-w)}{\#\mathrm{pairs}(n)} \;=\; \frac{\binom{n/2}{w/2}}{\binom{n}{w}} \tag{16}$$

Indeed, if w is odd, then this probability equals zero. Using Eq. (16), we get

$$\begin{aligned}
\Pr_\pi[\exists c \neq 0^t \text{ s.t } cQI_\pi = 0^n] \;&\leq\; \sum_{c \neq 0^t} \Pr_\pi[cQI_\pi = 0^n] && (17)\\
&\leq\; \sum_{w \in [m] \cap \{2i : i \in \mathbb{N}\}} \;\sum_{c\,:\,\mathrm{wt}(cQ) \in \{w, n-w\}} \frac{\binom{n/2}{w/2}}{\binom{n}{w}} && (18)\\
&<\; 3 \sum_{k \in [t] \cap \{2i : i \in \mathbb{N}\}} \left(\binom{t}{k-1} + \binom{t}{k} \right) \cdot \frac{\binom{n/2}{k/2}}{\binom{n}{k}} && (19)
\end{aligned}$$

where the last inequality optimizes the contribution of the various c's according to the weight of cQ, while using Eq. (15). Next, using $\binom{n/2}{k/2}^2 = o(\binom{n}{k})$, we upper-bound Eq. (19) by

$$\begin{aligned}
\sum_{k=2}^{t} \binom{t}{k} \cdot \binom{n}{k}^{-1/2} \;&<\; \sum_{k=2}^{t} (3t/k)^k \cdot (k/n)^{k/2} && (20)\\
&=\; \sum_{k=2}^{t} (9t^2/nk)^{k/2} && (21)
\end{aligned}$$

Finally, using $t < \sqrt{n}/6$, we upper-bound Eq. (21) by $2 \cdot (9t^2/n)$, and the lemma follows. ■

Proof of Theorem 2.7: Again, we consider two classes of linear functions, denoted GOOD and BAD, such that GOOD $\subset W_{0.5}$, whereas BAD $\cap\, W_{0.5} = \emptyset$, which implies that every function in BAD is 0.5-far from $W_{0.5}$. This time, however, the partition of $[n]$ to blocks is not fixed but is rather random.

That is, for $m = n/2$, we consider a uniformly chosen matching of $[n]$ into m ordered pairs, and denote the j^{th} pair in π by $\pi(j) = (\pi_1(j), \pi_2(j))$. For every such π and $j_0 \in [m]$, we let $g_{\pi, j_0, \sigma_1 \cdots \sigma_m}$ denote the linear function f_v such that $v = (v_1, ..., v_n)$ satisfies (1) $v_{\pi_1(j_0)} = v_{\pi_2(j_0)} = 0$ and (2) for every $j \in [n] \setminus \{j_0\}$ it holds that $v_{\pi_1(j)} = 1 - v_{\pi_2(j)} = \sigma_j$. The function $b_{\pi, j_0, \sigma_1 \cdots \sigma_m}$ is defined similarly, except that condition (1) is replaced by $v_{\pi_1(j_0)} = v_{\pi_2(j_0)} = 1$. Now, GOOD consists of all the functions $g_{\pi, j_0, \sigma_1 \cdots \sigma_m}$, whereas BAD consists of all the functions $b_{\pi, j_0, \sigma_1 \cdots \sigma_m}$.

The foregoing description corresponds to the description of the distribution of $(n-2)$-bit long strings of weight $m - 1 = (n-2)/2$ provided in the proof of Lemma 2.8. Indeed, the distributions described there correspond to setting the coordinates $\pi_1(j_0)$ and $\pi_2(j_0)$ to zero, which indeed fits the definition of $g_{\pi, j_0, \sigma}$. Here, however, it will be more convenient to consider the subclasses GOOD$_{i_1, i_2}$ and BAD$_{i_1, i_2}$ defined by conditioning the distribution over all (π, j_0, σ)-indexed functions on $\pi(j_0) = (i_1, i_2)$. We thus consider the following generic randomized process:

1. Select $i_1 \neq i_2$ uniformly in $[n]$.
2. Uniformly select $j_0 \in [m]$ and an m-way partition into ordered pairs, π, such that $\pi(j_0) = (i_1, i_2)$. Uniformly select $\sigma \in \{0,1\}^m$. Output $g_{\pi, j_0, \sigma}$ (resp., $b_{\pi, j_0, \sigma}$).

Indeed, depending on the case used in the last step (i.e., outputting $g_{\pi,j_0,\sigma}$ or $b_{\pi,j_0,\sigma}$), this process outputs a function uniformly distributed in either GOOD or BAD. It will be instructive to think of this selection as consisting of two steps: First, a pair (i_1,i_2) is selected, and next we select a function uniformly in GOOD_{i_1,i_2} (resp., BAD_{i_1,i_2}).

We consider the sequence of queries in the order they were issued, and evaluate the situation after each query. For each prefix of the sequence of queries, $\overline{q} = (q^{(1)}, ..., q^{(t)})$, and every fixed pair (i_1,i_2) selected as above, *we say that the pair (i_1,i_2) is* special w.r.t $\overline{q}$ *if $q^{(1)}, ..., q^{(t)}$ spans a vector of weight in $\{0,1,2,n-2,n-1,n\}$ with the exceptional positions belonging to $\{i_1,i_2\}$*. That is, if (i_1,i_2) is special w.r.t $\overline{q}$ then $q^{(1)}, ..., q^{(t)}$ span a vector q that satisfies the following condition: *there exists a $\tau \in \{0,1\}$ such that for every $i \in [n] \setminus \{i_1,i_2\}$ it holds that $q_i = \tau$.*

We may assume that the vectors in $\overline{q}$ are linearly independent, because all functions considered are linear and so their values at any linear combination of the $q^{(j)}$'s is determined by the corresponding answers. Likewise, we may assume that the vectors in $\overline{q}$ do not span 1^n, since all functions that we consider evaluate to $(m-1) \bmod 2$ at 1^n. Thus, if (i_1,i_2) is special w.r.t $\overline{q}$, then it is the case that $\overline{q}$ spans a vector q such that $\text{wt}(q) \in \{1,2,n-2,n-1\}$ (i.e., $q_i = \tau$ for every $i \in [n] \setminus \{i_1,i_2\}$ and $q_i = 1-\tau$ for some $i \in \{i_1,i_2\}$). We upper-bound the number, M, of special pairs (w.r.t $\overline{q} = (q^{(1)}, ..., q^{(t)})$) as follows. We consider a graph in which these pairs are vertices and edges connect pairs that have non-empty intersection. Then, each vertex has degree at most $4n$, and hence the graph contains an independent set of size $M/4n$. Considering the vectors corresponding to these pairs (i.e., or each pair (i_1,i_2) we consider a vector q such that $q_i = \tau$ for every $i \in [n] \setminus \{i_1,i_2\}$ and $q_i = 1-\tau$ for some $i \in \{i_1,i_2\}$), we obtain at least $M/8n$ independent vectors (i.e., vectors that use the same value τ and correspond to disjoint pairs). Thus, $M/8n \le t$, and it follows that the number of special pairs is at most $8tn$.

Fixing a pair (i_1,i_2) and letting Q^+ denote the t-by-n matrix obtained by using the $q^{(i)}$'s as rows in a matrix, we let Q denote the t-by-$(n-2)$ matrix obtained from Q^+ when omitting the columns i_1 and i_2. Note that *if (i_1,i_2) is not special w.r.t $\overline{q}$, then Q is full rank and its rows do not span 1^{n-2}*, because $cQ = \tau^{n-2}$ (for $c \ne 0^t$) implies that (i_1,i_2) is special w.r.t $\overline{q}$. Thus, in this case, the conditions of Lemma 2.8 hold (except that here the number of columns is $n-2$ rather than n).

Our analysis proceeds in iterations corresponding to the queries made by the adaptive tester. For every t, we denote the corresponding t-by-n matrix of queries by $Q^{(t)}$, and denote the corresponding set of non-special pairs of indices by $P^{(t)}$. Starting with $t=1$, we invoke Lemma 2.8 on the matrices Q obtained from $Q^{(1)} = q^{(1)}$ by dropping each $(i_1,i_2) \in P^{(1)}$, where $q^{(1)}$ is oblivious of everything (since it is the first query issued by the tester). We obtain corresponding sets $G^{(1)}_{i_1,i_2}$ that satisfy the two conditions of the lemma, which means that for every $(i_1,i_2) \in P^{(1)}$ conditioned on $\omega \in G^{(1)}_{i_1,i_2}$ the answer seen by the tester is uniformly distributed (regardless of whether the answer is obtained from a random function in GOOD_{i_1,i_2} or in BAD_{i_1,i_2}). We stress that, for any $(i_1,i_2) \in P^{(1)}$, the second query of the tester will be distributed identically, when considering the executions that correspond to a uniformly selected $\omega \in G^{(1)}_{i_1,i_2}$. Focusing only on these executions, we let $q^{(2)}$ describe the distribution of the second query, which is oblivious of $(i_1,i_2) \in P^{(1)}$, and consider the corresponding set $P^{(2)}$. (Indeed, $q^{(2)}$ as well as $P^{(2)}$ are random variables, but we shall treat them as if they were fixed, while noting that their distribution is independent of $(i_1,i_2) \in P^{(1)}$.)[6]

Likewise, in the t^{th} iteration, we invoke Lemma 2.8 on the matrices Q obtained from $Q^{(t)}$ by dropping each $(i_1,i_2) \in P^{(t)}$, and obtain sets $G^{(t)}_{i_1,i_2} \subseteq G^{(t-1)}_{i_1,i_2}$. The fact that the sets $G^{(t)}_{i_1,i_2}$ satisfy the two conditions of the lemma means that, for every $(i_1,i_2) \in P^{(t)}$, conditioned on $\omega \in G^{(t)}_{i_1,i_2}$ the

[6]Actually, also $q^{(1)}$ and $P^{(1)}$ are random variables, but their independence of (i_1,i_2) introduced later is trivial.

answer seen by the tester is uniformly distributed (regardless of whether the answer is obtained from a random function in GOOD_{i_1,i_2} or in BAD_{i_1,i_2}). So again, for any $(i_1,i_2) \in P^{(t)}$, the next query (i.e., $t+1^{\text{st}}$ query) of the tester will be distributed identically, when considering the executions that correspond to a uniformly selected $\omega \in G^{(t)}_{i_1,i_2}$.

This foregoing process makes sense as long as the sets $P^{(t)}$ and $G^{(t)}_{i_1,i_2}$ are not empty. Actually, we wish the sets $P^{(t)}$ and $G^{(t)}_{i_1,i_2}$ to be relatively large (i.e., have high density with respect to $[n] \times [n]$ and Ω, respectively), so that the probability mass of the executions that we consider is large. Recalling the upper bound on the number of special pairs, we have $|P^{(t)}| = (1 - o(1)) \cdot n^2$ as long as $t = o(n)$, whereas Lemma 2.8 guarantees that $|G^{(t)}_{i_1,i_2}| \geq (1 - (18t^2/n)) \cdot |\Omega|$. Thus, for $t = \sqrt{n}/9$, with probability at least $(1 - o(1)) \cdot 7/9 > 2/3$, a random pair (i_1,i_2) is in $P^{(t)}$ and $\omega \in G^{(t)}_{i_1,i_2}$. In this case, the answers to the t adaptively chosen queries are distributed identically regardless of whether the answers are from a random function in GOOD or from a random function in BAD. Thus, the statistical gap between random functions in GOOD and in BAD that can be observed by t adaptive queries is smaller than $1/3$, and the theorem follows. ■

On the tightness of the analysis. As we show next (in Proposition 2.9), Lemma 2.8 provides an accurate picture of the deviation (from the uniform distribution) of the answers to *individual queries* (i.e., the case of $t = 1$). Thus, improvements are possible only with respect to the handling of $t > 1$, where the hope is to reduce the deviation upper bound from its current value of $O(t^2/n)$ to a possible value of $O(t/n)$.

Proposition 2.9 *Suppose that v is uniformly distributed among all n-bit binary vectors having weight $m = n/2$. Then, for any $q \in \{0,1\}^n \setminus \{0^n, 1^n\}$, the value of $\langle q, v\rangle$ equals 1 with probability*

$$\frac{1}{2} + \chi_2(\text{wt}(q)) \cdot (1 - 2\chi_4(\text{wt}(q))) \cdot \frac{\binom{n/2}{\text{wt}(q)/2}}{\binom{n}{\text{wt}(q)}} \tag{22}$$

where $\chi_i(w) \stackrel{\text{def}}{=} 1$ if $w \equiv 0 \pmod{i}$ and $\chi_i(w) \stackrel{\text{def}}{=} 0$ otherwise.

Note that for odd $w = \text{wt}(q)$ the value of Eq. (22) equals $1/2$ (since $\chi_2(w) = 0$), whereas for even w the value of Eq. (22) deviates from $1/2$ (since $\chi_2(w) = 1$ and $1 - 2\chi_4(\text{wt}(q) = \pm 1)$. Specifically, for $w \equiv 2 \pmod{4}$ the value of Eq. (22) is strictly larger than $1/2$ (since $\chi_2(w) \cdot (1 - 2\chi_4(w)) = 1$).[7] Recall that $\frac{\binom{n/2}{w/2}}{\binom{n}{w}}$ is $\Theta(w^{-1/2}) \cdot \binom{n}{w}^{-1/2}$ (and always smaller than $\binom{n}{w}^{-1/2}$).

Proof: We use the same random process used in the proof of Lemma 2.8. Referring to the m-way partition π (selected in the first step), we call π good if it matches some 1-entry of q with a 0-entry of q (i.e., if there exists $j \in [m]$ such that $\{q_{\pi_1(j)}, q_{\pi_2(j)}\} = \{0,1\}$). Note that every π is good if $\text{wt}(q)$ is odd, and that if π is good then for a random v (selected in the second step) the value $\langle q, v\rangle$ is uniformly distributed (because the case in which $v_{\pi_1(j)} = 1 \neq v_{\pi_2(j)}$ is matched with the case in which $v_{\pi_1(j)} = 0 \neq v_{\pi_2(j)}$, where j satisfies $\{q_{\pi_1(j)}, q_{\pi_2(j)}\} = \{0,1\}$). On the other hand, if π is not good, then the value $\langle q, v\rangle$ equals $(\text{wt}(q)/2) \bmod 2$ (because for every $j \in [m]$ it holds that $q_{\pi_1(j)} = q_{\pi_2(j)}$ whereas $v_{\pi_1(j)} \neq v_{\pi_2(j)}$).[8] Thus, it remains to compute the probability that π is not good, which was essentially done in the proof of Lemma 2.8 (cf., Eq. (16)). Recall that

[7]Likewise, for $w \equiv 0 \pmod{4}$ the value of Eq. (22) is strictly smaller than $1/2$ (since $\chi_2(w) \cdot (1 - 2\chi_4(w)) = -1$).
[8]Indeed, it follows that $\sum_i q_i v_i = \sum_j q_{\pi_1(j)} = \text{wt}(q)/2$.

letting $w = \mathrm{wt}(q)$, and denoting by #pairs(x) the number of partitions of x elements to pairs, the probability that π is not good equals

$$\frac{\#\mathrm{pairs}(w) \cdot \#\mathrm{pairs}(n-w)}{\#\mathrm{pairs}(n)} = \frac{\binom{n/2}{w/2}}{\binom{n}{w}} . \tag{23}$$

The claim follows. ■

An alternative proof of a linear lower bound for non-adaptive testers. Building on Proposition 2.9, one can derive an alternative proof of Proposition 2.6. The key new component is the following Proposition 2.10, which seems of independent interest.

Proposition 2.10 *Let $t < n/2$ and let Q be a t-by-n full rank matrix such that its rows do not span the vector 1^n. Suppose that v is uniformly distributed among all n-bit binary vectors having weight $m = n/2$. Then, the variation distance between Qv and the uniform distribution over t-bit strings is at most t/n.*

Considering the random process presented in the proof of Theorem 2.7 (which starts by selecting a random pair (i_1, i_2)), and defining special pairs as in that proof, we establish Proposition 2.6 by considering the case that (i_1, i_2) is not special, and then invoking Proposition 2.10 on the residual matrix. Thus, it is left to prove the latter.

Proof: By using a variant of the XOR Lemma (i.e., Item 2 of Lemma A.4), we upper-bound the variation distance by the square root of the sum of the square of the corresponding biases. That is, we use the upper-bound

$$\sum_{\alpha \in \{0,1\}^t} \left| \mathrm{Pr}_v[Qv = \alpha] - 2^{-t} \right| \leq \frac{1}{2} \cdot \sqrt{\sum_{c \in \{0,1\}^t \setminus \{0^t\}} |\mathrm{Pr}_v[cQv = 1] - \mathrm{Pr}_v[cQv = 0]|^2} \tag{24}$$

$$= \frac{1}{2} \cdot \sqrt{\sum_{c \in \{0,1\}^t \setminus \{0^t\}} \left(\frac{\binom{n/2}{\mathrm{wt}(cQ)/2}}{\binom{n}{\mathrm{wt}(cQ)}} \right)^2} \tag{25}$$

where the equality is due to Proposition 2.9. Using the same reasoning as in the justification of Eq. (19) (in the the proof of Lemma 2.8), we upper bound Eq. (25) by

$$\frac{1}{2} \cdot \sqrt{3 \sum_{k \in [t] \cap \{2i : i \in \mathbb{N}\}} \left(\binom{t}{k-1} + \binom{t}{k} \right) \cdot \left(\frac{\binom{n/2}{k/2}}{\binom{n}{k}} \right)^2} < \sqrt{\sum_{k=2}^{t} \binom{t}{k} \cdot \left(\frac{\binom{n/2}{k/2}}{\binom{n}{k}} \right)^2} \tag{26}$$

$$< \sqrt{\frac{1}{2} \cdot \sum_{k=2}^{t} \frac{\binom{t}{k}}{\binom{n}{k}}} \tag{27}$$

where the last inequality uses $\binom{n/2}{k/2}^2 = o(\binom{n}{k})$. Hence, we obtain an upper bound of t/n, and the claim follows. ■

3 Hardness of Testing a Subclass of Width 3 OBDDs

We shall consider the class of linear functions over GF(3), consisting of all such functions that have binary coefficients. That is, for every $v \in \{0,1\}^n$, we consider the function $f_v : \mathrm{GF}(3)^n \to \mathrm{GF}(3)$ defined by $f_v(x) = \sum_{i=1}^n v_i x_i$, where the arithmetic is modulo 3. Let $\mathcal{BL}_3 = \{f_v : v \in \{0,1\}^n\}$.

Conjecture 3.1 *Testing $\mathcal{BL}_3$ requires $\Omega(n)$ queries, even when allowing adaptive testers of two-sided error.*

Theorem 3.2 (Theorem 5, restated): *Testing $\mathcal{BL}_3$ requires $\Omega(\sqrt{n})$ queries, even when allowing adaptive testers of two-sided error.*

Proof: We consider the class $\mathrm{BAD} = \{b_{j_0,v} : j_0 \in [n], v \in \{0,1\}^n\}$ such that $b_{j_0,v}(x) \stackrel{\rm def}{=} f_v(x) + x_{j_0}$. Note that all functions in BAD are linear and that exactly half of BAD is not in $\mathcal{BL}_3$ (since $b_{j_0,v} \in \mathcal{BL}_3$ if and only if $v_{j_0} = 0$). Hence, with probability 1/2, a uniformly selected function in BAD is 2/3-far from $\mathcal{BL}_3$. Our goal is to prove that distinguishing a uniformly selected function in $\mathcal{BL}_3$ from a uniformly selected function in BAD requires $\Omega(\sqrt{n})$ queries.

Recall that an element in either sets is selected by specifying an index $j_0 \in [n]$ and an n-bit string. Fixing any sequence of queries $\overline{q} = (q^{(1)}, ..., q^{(t)})$, we shall show that if this sequence has a certain feature with respect to j_0, then the answers are distributed almost identically in the two distributions. This feature is defined next, where w is an integer (i.e., we shall use $w = \sqrt{n}$).

Definition 3.2.1 *An index $j \in [n]$ is called w-*special *with respect to a sequence of queries $\overline{q} = (q^{(1)}, ..., q^{(t)})$ if there exists a linear combination of these queries that yields an n-bit string q such that $j \in \mathrm{supp}(q)$ and $|\mathrm{supp}(q)| \le w$, where $\mathrm{supp}(q) \stackrel{\rm def}{=} \{i : q_i \neq 0\}$.*

It will be convenient to use matrix notation in our analysis. Presenting $\overline{q}$ as a matrix, denoted Q, such that the $i^{\rm th}$ row of Q equals $q^{(i)}$, the foregoing condition asserts that there exists a t-vector c such that $\mathrm{supp}(cQ)$ contains j as well as at most $w-1$ other indices. Thus, we get:

Claim 3.2.2 *For any sequence of t queries, $\overline{q}$, there exists at most $w \cdot t$ indices that are w-special with respect to $\overline{q}$.*

Proof: Let S denote the set of w-special indices with respect to $\overline{q}$. For every $j \in S$, there exists a t-vector $c^{(j)}$ such that $\mathrm{supp}(c^{(j)}Q)$ contains j as well as at most $w-1$ other elements of S. Using a greedy strategy, we can obtain a set I of at least $|S|/w$ elements of S such that for every $j \in I$ it holds that $\mathrm{supp}(c^{(j)}Q) \cap I = \{j\}$. Thus, the rank of Q is lower bounded by $|S|/w$, and the claim follows. □

Claim 3.2.3 *Suppose that j_0 is not w-special with respect to $\overline{q} = (q^{(1)}, ..., q^{(t)})$. Then, for every $\alpha \in \{0,1,2\}^t$, when $v = (v_1, ..., v_n)$ is selected uniformly in $\{0,1\}^n$, it holds that*

$$\Pr_v[(f_v(q^{(1)}), ..., f_v(q^{(t)})) = \alpha] \;=\; \Pr_v[(b_{j_0,v}(q^{(1)}), ..., b_{j_0,v}(q^{(t)})) = \alpha] \pm 2^{-(w-1)}. \tag{28}$$

Proof: For every $\alpha \in \{0,1,2\}^n$, we denote by $D_{j_0,\overline{q}}(\alpha)$ the difference between the two probabilities in Eq. (28); that is,

$$D_{j_0,\overline{q}}(\alpha) \stackrel{\rm def}{=} \Pr_v[(f_v(q^{(1)}), ..., f_v(q^{(t)})) = \alpha] \;-\; \Pr_v[(b_{j_0,v}(q^{(1)}), ..., b_{j_0,v}(q^{(t)})) = \alpha]. \tag{29}$$

Our aim is to prove that the max-norm of $D_{j_0,\overline{q}}(\cdot)$ is at most $2^{-(w-1)}$. By using the relation between bases (cf. Lemma A.5 (Part 2)),[9] it suffices to show that for every $c \in \{0,1,2\}^t$ it holds that

$$\sum_{\tau\in\{0,1,2\}} \left| \sum_{\alpha\in S_{c,\tau}} D_{j_0,\overline{q}}(\alpha) \right| \leq 2^{-(w-1)}, \tag{30}$$

where $S_{c,\tau} \stackrel{\rm def}{=} \{\alpha \in \{0,1\}^t : \sum_{i=1}^{t} c_i\alpha_i = \tau\}$ denotes the set of all t-bit vectors that have $3k+\tau$ non-zero entries (for some k). The l.h.s of Eq. (30) equals

$$\sum_{\tau\in\{0,1,2\}} \left| \Pr_v\left[\sum_{i=1}^{t} c_i f_v(q^{(i)}) = \tau\right] \; - \; \Pr_v\left[\sum_{i=1}^{t} c_i b_{j_0,v}(q^{(i)}) = \tau\right] \right|. \tag{31}$$

Using the linearity of both functions, and moving to matrix notation, each term in Eq. (31) equals

$$\Pr_v[f_v(cQ) = \tau] \; - \; \Pr_v[b_{j_0,v}(cQ) = \tau], \tag{32}$$

which equals $\Pr_v[cQv = \tau] - \Pr_v[cQ(v + u^{j_0}) = \tau]$, where $u^{j_0} = 0^{j_0-1}10^{n-j_0}$ is the $j_0^{\rm th}$ unit vector. Thus, Eq. (31) equals

$$\sum_{\tau\in\{0,1,2\}} \left| \Pr_v\left[cQv = \tau\right] \; - \; \Pr_v[cQv + cQu^{j_0} = \tau] \right|. \tag{33}$$

To upper-bound Eq. (33), we consider two cases (regarding the value of cQu^{j_0}). If $cQu^{j_0} = 0$, then Eq. (33) equals zero. On the other hand, if $cQu^{j_0} \neq 0$, then $\mathrm{supp}(cQ)$ contains j_0, and it follows that $|\mathrm{supp}(cQ)| > w$ (because otherwise j_0 would have been w-special w.r.t $\overline{q}$). But in this case, it follows that $\sum_{\tau\in\{0,1,2\}} |\Pr_v[cQv = \tau] - \frac{1}{3}| < 2^{-w}$ (see Eq. (43)) and the same holds for $\Pr_v[cQv = \tau - cQu^{j_0}]$. Thus, Eq. (33) is upper-bounded by $2 \cdot 2^{-w}$, and the claim follows. □

Armed with Claims 3.2.2 and 3.2.3, we prove the theorem by considering the sequence of queries in the order they were issued. Setting $w = \sqrt{n}$, we evaluate the situation after each additional query. Using Claim 3.2.3, we note that as long as j_0 is not special with respect to the queries made, the answers are almost oblivious of whether the function is uniformly selected in BAD or in $\mathcal{BL}_3$ in the sense that the probabilistic deviation on each possible sequence of answers (i.e., α) is at most $2^{-(w-1)}$. Recalling that the functions in $\mathcal{BL}_3$ are oblivious of j_0, it follows that the answers obtained from a random function in BAD are also almost oblivious of j_0 (as long as j_0 is not special with respect to the queries made). Noting that the answers determine the next query, we infer that this query is also almost oblivious of the currently non-special value of j_0, and so the probability that j_0 is special with respect to the augmented sequence of queries can be bounded using Claim 3.2.2. Details follow.

We may assume, (as usual and) without loss of generality, that the tester is deterministic, and so the query sequence is determined adaptively by the previous answers. Thus, we consider the 2^{t-1} possible t-query sequences that arise from each possible sequence of t answers. For each such sequence, we first dispose of the case that j_0 is special with respect to it, which by Claim 3.2.2 happens with probability at most tw/n. Assuming that j_0 is not special with respect to that sequence, we conclude (by Claim 3.2.3) that the corresponding sequence of answers occurs with

[9] Specifically, letting ω denote the third root of unity, it suffices to upper-bound $|\sum_{\tau\in\mathrm{GF}(3)} \omega^\tau \sum_{\alpha\in S_{c,\tau}} D_{j_0,\overline{q}}(\alpha)|$, where $S_{c,\tau} = \{\alpha : \sum_i c_i\alpha_i = \tau\}$. Instead, we upper-bound each of the three terms of the outer summation (and use $|\omega| = 1$).

about the same probability in both distributions. Over all, the statistical distance between the observed answers is at most $(tw/n) + 2^{t-1} \cdot 2^{-(w-1)}$, and the theorem follows.[10] ■

4 Hardness of Testing the Class of Width 4 Realizable Functions

In this section we establish Theorems 1 and 6.

Conjecture 4.1 *Testing the class of functions that are implementable by width-4 OBDDs requires $\Omega(n)$ queries, even when allowing adaptive testers of two-sided error.*

Theorem 4.2 (Theorem 1, restated): *Testing the class of functions that are implementable by width-4 OBDDs requires $\Omega(\sqrt{n})$ queries, even when allowing adaptive testers of two-sided error.*

Proof: We consider Boolean functions of $4n$-bit long strings, which are quadratic polynomials over GF(2). Specifically, these functions are linear combinations of n quadratic expressions, where each quadratic expression refers to a distinct block of four variables. A generic block, containing the variables x_1, x_2, x_3, x_4, will contribute a linear combination of x_1x_3 and x_2x_4, where the combination $x_1x_3 + x_2x_4$ is considered bad because the expression $x_0 + x_1x_3 + x_2x_4$ cannot be computed by a width-4 OBDDs. Specifically, letting $f_0(x_1, x_2, x_3, x_4) = 0$, $f_1(x_1, x_2, x_3, x_4) = x_1x_3$, and $f_2(x_1, x_2, x_3, x_4) = x_2x_4$, we will consider the class GOOD that consists of functions of the form $g_{\sigma_1,...,\sigma_n}$ such that

$$g_{\sigma_1,...,\sigma_n}(x_1, ..., x_{4n}) = \sum_{j\in[n]} f_{\sigma_j}(x_{4(j-1)+1}, ..., x_{4(j-1)+4}), \tag{34}$$

where $\sigma_1, ..., \sigma_n \in \{0, 1, 2\}$. Note that each such function can be computed by a width-4 OBDD, which uses one "bit" to store the accumulated sum and another "bit" to compute the value of the current block. In contrast, the class BAD consists of functions of the form $b_{j_0,\sigma_1,...,\sigma_n}$ such that

$$\begin{aligned} b_{j_0,\sigma_1,...,\sigma_n}(x_1, ..., x_{4n}) &= \sum_{j\in[n]\setminus\{j_0\}} f_{\sigma_j}(x_{4(j-1)+1}, ..., x_{4(j-1)+4}) \\ &\quad + x_{4(j_0-1)+1}x_{4(j_0-1)+3} + x_{4(j_0-1)+2}x_{4(j_0-1)+4} \end{aligned} \tag{35}$$

[10] Formally, let $X = X(v)$ be a random variable representing the sequence of answers that the tester sees when querying a uniformly distributed function in $\mathcal{BL}_3$ (i.e., the function f_v, where v is uniformly distributed in $\{0,1\}^n$). Likewise, let $Y_j = Y_j(v)$ be a random variable representing the sequence of answers that the tester sees when querying $b_{j,v}$, where v is uniformly distributed in $\{0,1\}^n$. Then, we are interested in

$$\begin{aligned} \Delta &\stackrel{\text{def}}{=} \frac{1}{2} \cdot \sum_{\alpha\in\{0,1,2\}^t} \left| \Pr[X=\alpha] - \frac{1}{n} \cdot \sum_{j\in[n]} \Pr[Y_j=\alpha] \right| \\ &\leq \frac{1}{2n} \cdot \sum_{\alpha\in\{0,1,2\}^t} \sum_{j\in[n]} \left|\Pr[X=\alpha] - \Pr[Y_j=\alpha]\right|. \end{aligned}$$

Letting S_α denote the set of indices that are w-special with respect to the query sequence induced by the answer sequence α, we may upper-bound Δ by considering seperately terms that correspond to $j \in S_\alpha$ and terms that correspond to $j \in [n] \setminus S_\alpha$. Furthermore, the terms that correspond to $j \in S_\alpha$ are partitioned according to the index of the query that causes j to be w-special, and so each such j contributes at most two units to the total sum (since the event of becoming special depends only on the first $i-1$ answers whereas the distribution of these answers is almost identical in the two cases).

Since, except when $\sigma_1 \cdots \sigma_{j_0-1} = 0^{j_0-1}$, the j_0^{th} block can not be computed by a width-4 OBDD (while maintaining the accumulated sum), it follows that such functions are 1/16-far from width-4 (see Lemma A.6, which is a simple XOR Lemma for OBDDs, which is also an over-kill).

Our goal is to prove that a random function in GOOD is hard to distinguish from a random function in BAD, where "random" does not necessarily refer to the uniform distribution over the corresponding set (but rather any two distributions will do). Specifically, we consider a distribution over GOOD, in which each σ_i is set to 0 with probability 1/2 and is uniformly distributed in $\{1, 2\}$ otherwise. (This random selection process determines a function $g_{\sigma_1,...,\sigma_n} \in$ GOOD.) We consider a related distribution over GOOD$\cup$BAD, where $\sigma_1, ..., \sigma_n$ are selected as above, the index j_0 is selected uniformly in $[n]$, and the function being determined is $g_{\sigma_1,...,\sigma_n} + a_{j_0}$, where $a_{j_0}(x_1, ..., x_{4n}) = x_{4(j_0-1)+1}x_{4(j_0-1)+3} + x_{4(j_0-1)+2}x_{4(j_0-1)+4}$. Note that the resulting function is in BAD if and only if both $\sigma_1 \cdots \sigma_{j_0-1} \neq 0^{j_0-1}$ and $\sigma_{j_0} = 0$, which means that it is in BAD with probability $\frac{1}{2} - o(1)$.

Our analysis reduces to analyzing related families of linear functions defined over variables $y_1, ..., y_{2n}$ such that $y_{2(j-1)+1} = x_{4(j-1)+1}x_{4(j-1)+3}$ and $y_{2(j-1)+2} = x_{4(j-1)+2}x_{4(j-1)+4}$. Specifically, we first show that distinguishing the foregoing two distributions (of quadratic functions) leads to distinguishing the two corresponding distributions of linear functions, where in both the latter distributions $\sigma_1, ..., \sigma_n$ and j_0 are selected as above (i.e., j_0 is distributed uniformly in $[n]$ and each σ_i is set to 0 with probability 1/2 and is uniformly distributed in $\{1, 2\}$ otherwise). Letting $f'_0(y_1, y_2) = 0$, $f'_1(y_1, y_2) = y_1$, and $f'_2(y_1, y_2) = y_2$, the linear functions in these two distributions are:

$$
\begin{aligned}
g'_{\sigma_1,...,\sigma_n}(y_1, ..., y_{2n}) &= \sum_{j\in[n]} f'_{\sigma_j}(y_{2(j-1)+1}, y_{2(j-1)+2}) && (36)\\
b'_{j_0,\sigma_1,...,\sigma_n}(y_1, ..., y_{2n}) &= g'_{\sigma_1,...,\sigma_n}(y_1, ..., y_{2n}) + y_{2(j_0-1)+1} + y_{2(j_0-1)+2}\,. && (37)
\end{aligned}
$$

The reduction between these distinguishing problems is quite straightforward: Given a distinguisher D for the original distinguishing problem (i.e., regarding quadratic functions), we obtain a distinguisher D' for the distinguishing problem regarding linear functions. The new distinguisher (i.e., D') invokes D and serves each query $q = (q_1, ..., q_{4n})$ that it issues (to its quadratic oracle) by forwarding the query $q' = (q'_1, ..., q'_{2n})$ to the actual (linear function) oracle, where $q'_{2(j-1)+1} = q_{4(j-1)+1}q_{4(j-1)+3}$ and $q'_{2(j-1)+2} = q_{4(j-1)+2}q_{4(j-1)+4}$ for every $j \in [n]$. Thus, when given oracle access to $g'_{\sigma_1,...,\sigma_n}$, we emulate an execution of D with $g_{\sigma_1,...,\sigma_n}$, whereas when given oracle access to $b'_{j_0,\sigma_1,...,\sigma_n}$, we emulate an execution of D with $g_{\sigma_1,...,\sigma_n} + a_{j_0}$.

We now turn to prove that distinguishing the two aforementioned distributions on linear functions requires $\Omega(\sqrt{n})$ queries. Our proof follows the structure of the proof of Theorem 3.2. Specifically, in analogy to Definition 3.2.1, we say that $j \in [n]$ is w-special with respect to a sequence of queries $\overline{q}$ if there exists a linear combination of these queries that yields a $2n$-bit string q such that $\{2j-1, 2j\} \cap \text{supp}(q) \neq \emptyset$ and $|\text{supp}(q)| \leq w$. Analogously to Claim 3.2.2, the number of w-special indices with respect to a sequence of t queries is bounded by $w \cdot t$. Next, analogously to Claim 3.2.3 we upper-bound the deviation of the answers whenever j_0 is not w-special with respect to the sequence of queries.

Claim 4.2.1 *Suppose that j_0 is not w-special with respect to $\overline{q} = (q^{(1)}, ..., q^{(t)}) \in (\{0,1\}^{2n})^t$. Then, for every $\alpha \in \{0,1\}^t$, when $\sigma = (\sigma_1, ..., \sigma_n)$ is selected as above, it holds that*

$$
\Pr[(g'_\sigma(q^{(1)}), ..., g'_\sigma(q^{(t)})) = \alpha] = \Pr[(b'_{j_0,\sigma}(q^{(1)}), ..., b_{j_0,\sigma}(q^{(t)})) = \alpha] \pm 2^{-\Omega(w)}\,. \qquad (38)
$$

Proof: Like in the proof of Claim 3.2.3, it suffices to show that, for every $c \in \{0,1\}^t$,

$$
\left|\Pr_\sigma\left[g'_\sigma(cQ) = 1\right] - \Pr_\sigma\left[b'_{j_0,\sigma}(cQ) = 1\right]\right| \leq 2^{-\Omega(w)}, \qquad (39)
$$

where Q is the matrix with the $q^{(i)}$'s as rows. Let $q = cQ$ and recall that $b'_{j_0,\sigma}(q) = g'_\sigma(q) + q_{2j_0-1} + q_{2j_0}$. We consider two cases. If $q_{2j_0-1} = q_{2j_0} = 0$, then the l.h.s of Eq. (39) equals zero. Otherwise (i.e., $\{2j_0 - 1, 2j_0\} \cap \mathrm{supp}(q) \neq \emptyset$), since j_0 is not w-special, it holds that $|\mathrm{supp}(q) \setminus \{2j_0 - 1, 2j_0\}| \geq w - 1$. Hence, there exists at least $(w-1)/2$ indices j in $[n] \setminus \{j_0\}$ such that $(q_{2j-1}, q_{2j}) \neq (0,0)$, which means that for each such j the value of $f'_{\sigma_j}(q_{2(j-1)+1}, q_{2(j-1)+2})$ is not fixed when σ_j is random as above. Specifically, for each such j (i.e., j such that $(q_{2j-1}, q_{2j}) \neq (0,0)$), it holds that

$$\mathrm{Pr}_{\sigma_j}\left[f'_{\sigma_j}(q_{2(j-1)+1}, q_{2(j-1)+2}) = 1\right] = \begin{cases} \frac{1}{4} & \text{if } q_{2(j-1)+1} + q_{2(j-1)+2} = 1 \\ \frac{1}{2} & \text{otherwise (i.e., } q_{2(j-1)+1} = q_{2(j-1)+2} = 1) \end{cases} \tag{40}$$

and these events, which refer to different j's, are independent. Recalling Eq. (36)&(37), we conclude that each of the two probabilities in the l.h.s of Eq. (39) is $\frac{1}{2} \pm 2^{-\Omega(w)}$, and the claim follows. □

The rest of the analysis mimics the proof of Theorem 3.2. ■

Establishing Theorem 6. In the course of the proof of Theorem 4.2 we actually established a lower bound on the complexity of testing the set of linear functions defined in Eq. (36). Letting $g''_\sigma(z_1, ..., z_{3n})$ equal $g'_\sigma(z_1, z_2, z_4, z_5, ..., z_{3n-2}, z_{3n-1})$ we obtain a set of linear functions in which there are no consecutive influential variables. Theorem 6 follows by observing that testing the former property is reduced to testing the latter property.

Acknowledgments

Part of this work is based on joint research with Dana Ron, who refused to co-author it.

References

[1] N. Alon, O. Goldreich, J. Håstad, R. Peralta. Simple Constructions of Almost k-wise Independent Random Variables. *Journal of Random Structures and Algorithms*, Vol. 3, No. 3, pages 289–304, 1992. Preliminary version in *31st FOCS*, 1990.

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

[3] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, and M. Sudan. Linearity testing in characteristic two. In *36th FOCS*, pages 432–441, 1995.

[4] E. Blais. Testing juntas almost optimally. In *41st STOC*, pages 151–158, 2009.

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

[6] I. Diakonikolas, H. K. Lee, K. Matulef, K. Onak, R. Rubinfeld, R. A. Servedio, and A. Wan. Testing for concise representations. In *48th FOCS*, pages 549–557, 2007.

[7] I. Diakonikolas, H. K. Lee, K. Matulef, , R. A. Servedio, and A. Wan. Efficient testing of sparse GF(2) polynomials. In *35th ICALP*, pages 502–514, 2008.

[8] G. Even. Construction of Small Probabilistic Spaces for Deterministic Simulation. M.Sc. Thesis, Computer Science Dept., Technion – Israel Institute of Technology, Aug. 1991. (In Hebrew, abstract in English).

[9] E. Fischer, G. Kindler, D. Ron, S. Safra, and S. Samorodnitsky. Testing juntas. *JCSS*, Vol. 68 (4), pages 753–787, 2004.

[10] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, July 1998. Extended abstract in *37th FOCS*, 1996.

[11] O. Goldreich, M. Krivelevich, I. Newman, and E. Rozenberg. Hierarchy Theorems for Property Testing. *ECCC*, TR08-097, 2008. Extended abstract in the proceedings of *RANDOM'09*.

[12] O. Goldreich, N. Nisan and A. Wigderson. On Yao's XOR-Lemma. *ECCC*, TR95-050, 1995.

[13] O. Goldreich and D. Ron. On Proximity Oblivious Testing. *ECCC*, TR08-041, 2008. Extended abstract in the proceedings of the *41st STOC*, 2009.

[14] O. Lachish, I. Newman, and A. Shapira. Space Complexity vs. Query Complexity. *Computational Complexity*, Vol. 17, pages 70–93, 2008.

[15] J. Naor and M. Naor. Small-bias Probability Spaces: Efficient Constructions and Applications. *SICOMP*, Vol 22, 1993, pages 838–856. Preliminary version in *22nd STOC*, 1990.

[16] I. Newman. Testing membership in languages that have small width branching programs. *SIAM Journal on Computing*, 31(5):1557–1570, 2002.

[17] M. Parnas, D. Ron, and A. Samorodnitsky. Testing basic boolean formulae. *SIDMA*, Vol. 16 (1), pages 20–46, 2002.

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

[19] D. Ron. Algorithmic and Analysis Techniques in Property Testing. *Foundations and Trends in TCS*, to appear.

[20] D. Ron and G. Tsur. Testing Computability by Width Two OBDDs. In *12th RANDOM*, Springer, LNCS 5687, pages 686–699, 2009.

[21] D. Ron and G. Tsur. Testing Computability by Width Two OBDDs where the Variable Order is Unknown. In *7th CIAC*, to appear.

[22] R. Rubinfeld. On the Robustness of Functional Equations. *SIAM Journal on Computing*, Vol. 28 (6), pages 1972–1997, 1999. Extended abstract in *35th FOCS*, 1994.

[23] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, Vol. 25 (2), pages 252–271, 1996.

Appendix: Technical Background

This appendix contains background material that is known, but may not be easily accessible otherwise. In particular, Section A.1 reproduces Guy Even's upper bound on the bias of random n-bit strings of weight that is a multiple of 3 (cf. [8]); Section A.2 reproduces a known proof of Umesh Vazirani's Information Theoretic XOR Lemma (as well as its generalization to $\mathrm{GF}(p)$ for any prime p); and Section A.3 provides a simple proof of Yao's XOR Lemma for OBDDs (and other related models of computation).

A.1 The bias of the Mod 3 Sample Space

Referring to the uniform distribution over n-bit strings having weight that is a multiple of 3, we present a proof that this distribution has an exponentially vanishing bias, where the bias of a distribution is as defined in Eq. (45).

Proposition A.1 [8]: *Let S be the set of all n-bit strings having a number of 1-entries that is divisible by three. Then, S is an $2^{-O(n)}$-bias sample space.*

Proof: We let $X = X_1 \cdots X_n$ denote a uniformly distributed n-bit string. We first consider the distribution of $\sum_{i=1}^n X_i \bmod 3$. Letting $p_\sigma(n) \stackrel{\rm def}{=} \Pr[\sum_{i=1}^n X_i \equiv \sigma \pmod 3]$, we note that

$$p_\sigma(n) = \frac{1}{2} \cdot p_\sigma(n-1) + \frac{1}{2} \cdot p_{\sigma-1}(n-1) = \frac{1}{2} - \frac{p_{\sigma+1}(n-1)}{2} \qquad (41)$$

and it follows that $|p_\sigma(n) - \frac{1}{3}| = \frac{1}{2} \cdot |p_{\sigma+1}(n-1) - \frac{1}{3}|$. Thus, we get

$$\sum_{\sigma\in\{0,1,2\}} \left|p_\sigma(n) - \frac{1}{3}\right| = \frac{1}{2} \cdot \sum_{\sigma\in\{0,1,2\}} \left|p_\sigma(n-1) - \frac{1}{3}\right| \qquad (42)$$

and similarly for $\max_{\sigma\in\{0,1,2\}}\{|p_\sigma(n-1) - \frac{1}{3}|\}$. Recalling that $p_0(1) = p_1(1) = \frac{1}{2}$, it follows that $p_\sigma(n) = \frac{1}{3} \pm 2^{-n}$. We also mention (for use in the proof of Claim 3.2.3) that

$$\sum_{\sigma\in\{0,1,2\}} \left|p_\sigma(n) - \frac{1}{3}\right| \;=\; \frac{2}{3} \cdot 2^{-(n-1)} \qquad (43)$$

We now turn to analyze the bias of the various XORs. That is, for any fixed non-zero string $q \in \{0,1\}^n$, we consider the probability

$$\Pr\left[\langle q, X\rangle = 0 \;\middle|\; \sum_{i=1}^n X_i \equiv 0 \pmod 3\right] = \frac{\Pr[\langle q, X\rangle = 0 \wedge \sum_{i=1}^n X_i \equiv 0 \pmod 3]}{\Pr[\sum_{i=1}^n X_i \equiv 0 \pmod 3]}$$

We know that the denominator is $\frac{1}{3} \pm 2^{-n}$, and so we focus on the numerator. We distinguish two cases, according to the weight of q, where we assume (w.l.o.g.) that $q = 1^{\mathrm{wt}(q)}0^{n-\mathrm{wt}(q)}$.

Case 1: $w \stackrel{\rm def}{=} \mathrm{wt}(q) \le n/2$. In this case, we have

$$\begin{aligned}
\Pr\left[\langle q, X\rangle = 0 \wedge \sum_{i=1}^n X_i \equiv 0 \pmod 3\right] &= \Pr\left[\sum_{i=1}^w X_i \equiv 0 \pmod 2 \wedge \sum_{i=1}^n X_i \equiv 0 \pmod 3\right] \\
&= \frac{1}{2} \cdot \Pr\left[\sum_{i=1}^n X_i \equiv 0 \pmod 3 \;\middle|\; \sum_{i=1}^w X_i \equiv 0 \pmod 2\right]
\end{aligned}$$

We note that, for any fixing of values to $X_1, ..., X_w$ and every $\sigma \in \{0,1,2\}$, it holds that

$$\Pr\left[\sum_{i=w+1}^{n} X_i \equiv \sigma \pmod 3\right] = p_\sigma(n-w) = \frac{1}{3} \pm 2^{-(n-w)}$$

and using $w \le n/2$ we get that $\Pr[\langle q, X\rangle = 0 \wedge \sum_{i=1}^{n} X_i \equiv 0 \pmod 3] = \frac{1}{6} \pm 2^{-n/2}$.

Case 2: $w \stackrel{\text{def}}{=} \text{wt}(q) \ge n/2$**.** In this case, we use

$$\Pr\left[\langle q, X\rangle = 0 \wedge \sum_{i=1}^{n} X_i \equiv 0 \pmod 3\right] = \Pr\left[\sum_{i=1}^{n} X_i \equiv 0 \pmod 6\right]$$

and observe that $\sum_{i=1}^{n} X_i \equiv 0 \pmod \ell$ represents a random walk on a directed ℓ-cycle where we traverse an edge with probability $1/2$ and otherwise remain in place. It can be easily seen that the corresponding Markov Chain has a second eigenvalue of $1 - \Theta(\ell^{-2})$, and so the probability of reaching any fixed node in an n-step random walk is $\frac{1}{\ell} \pm 2^{-\Omega(n/\ell^2)}$.

The claim follows. ■

A.2 The Information Theoretic XOR-Lemma

The Information Theoretic XOR-Lemma, commonly attributed to Umesh Vazirani (see also [1, Apdx]), relates two measures of the "randomness" of distributions over n-bit long strings.

- The statistical difference from uniform; namely, the statistical difference (variation difference) between the "target" distribution and the uniform distribution.
- The maximum bias of the xor of certain bit positions; namely, the bias of a 0-1 random variable obtained by taking the exclusive-or of certain bits in the "target" distribution.

The Information Theoretic XOR-Lemma asserts that the statistical difference from uniform is upper-bounded by $\sqrt{2^n}$ times the maximum bias of the XOR's.

Formal setting. Let π be a an arbitrary probability distribution over $\{0,1\}^n$ and let μ denote the uniform distribution over $\{0,1\}^n$ (i.e., $\mu(x) = 2^{-n}$ for every $x \in \{0,1\}^n$). Let $x = x_1 \cdots x_n$ and $N \stackrel{\text{def}}{=} 2^n$. The XOR-Lemma relates two "measures of closeness" of π to μ.

- The statistical difference ("variation difference") between π and μ; namely,

$$\texttt{stat}(\pi) \stackrel{\text{def}}{=} \frac{1}{2} \cdot \sum_x |\pi(x) - \mu(x)| \tag{44}$$

- The "maximum bias" of the exclusive-or of certain bit positions in strings chosen according to the distribution π; namely,

$$\texttt{maxbias}(\pi) \stackrel{\text{def}}{=} \max_{S \ne \emptyset} \{|\pi(\{x : \oplus_{i \in S} x_i = 0\}) - \pi(\{x : \oplus_{i \in S} x_i = 1\})|\} \tag{45}$$

The XOR-Lemma states that $\texttt{stat}(\pi) \le \sqrt{N} \cdot \texttt{maxbias}(\pi)$. Its proof is based on viewing distributions as elements in an N-dimensional vector space and observing that the two measures considered by the lemma are merely two norms taken with respect to two different orthogonal bases. Hence, the XOR-Lemma follows from a (more general and quite straightforward) technical claim that relates norms taken with respect to different orthonormal bases.

The XOR-Lemma and vector spaces. Probability distributions over $\{0,1\}^n$ are functions from $\{0,1\}^n$ to the reals. Such functions form a N-dimensional vector space. The standard basis, denoted K, for this space is the orthonormal basis defined by the "Kroniker functions" (i.e., the Boolean functions $\{k_\alpha : \alpha \in \{0,1\}^n\}$ where $k_\alpha(x) = 1$ if $x = \alpha$). The statistical difference between two distributions equals (half) the norm L_1 of their difference taken in the above K basis. On the other hand, the maxbias of a distribution equals the maximum "Fourier coefficient" of the distribution, which in turn corresponds to the max-norm (norm L_∞) of the distribution taken in a different basis. The basis is defined by the functions $\{b_S : S \subseteq \{1,2,...,n\}\}$, where $b_S(x) = (-1)^{\sum_{i \in S} x_i}$. Note that $b_S(x) = 1$ if the exclusive-or of the bits $\{x_i : i \in S\}$ is 0 and $b_S(x) = -1$ otherwise. The new basis is orthogonal but not orthonormal. We hence consider the normalized basis, denoted F, consisting of the functions $f_S = \frac{1}{\sqrt{N}} \cdot b_S$.

Notation: Let B be an orthonormal basis and r an integer. We denote by $\mathsf{N}_r^B(v)$ the norm L_r of v with respect to the basis B. Namely, $\mathsf{N}_r^B(v) = (\sum_{e \in B} \langle e, v\rangle^r)^{(1/r)}$, where $\langle e, v\rangle$ is the absolute value of the inner product of the vectors e and v. We denote by $\mathsf{N}_\infty^B(v)$ the limit of $\mathsf{N}_r^B(v)$ when $r \to \infty$ (i.e., $\mathsf{N}_\infty^B(v)$ is $\max_{e \in B}\langle e, v\rangle$).

Clearly, $\texttt{stat}(\pi) = \frac{1}{2} \cdot \mathsf{N}_1^K(\pi - \mu)$, whereas $\texttt{maxbias}(\pi) = \sqrt{N} \cdot \mathsf{N}_\infty^F(\pi - \mu)$. Following is a proof of the second equality. Let $\delta(x) = \pi(x) - \mu(x)$. Clearly, $\texttt{maxbias}(\mu) = 0$, and hence $\texttt{maxbias}(\pi) = \texttt{maxbias}(\delta)$. Also $\sum_x \delta(x) = 0$, and so $\sum_x f_\emptyset(x) \cdot \delta(x) = 0$. We get

$$
\begin{aligned}
\texttt{maxbias}(\delta) &= \max_{S \neq \emptyset} \{|\delta(\{x : b_S(x) = 1\}) - \delta(\{x : b_S(x) = -1\})|\} && (46) \\
&= \max_{S \neq \emptyset} \left\{ \left| \sum_x b_S(x) \cdot \delta(x) \right| \right\} && (47) \\
&= \sqrt{N} \cdot \max_S \left\{ \left| \sum_x f_S(x) \cdot \delta(x) \right| \right\} && (48) \\
&= \sqrt{N} \cdot \mathsf{N}_\infty^F(\delta) && (49)
\end{aligned}
$$

The proof of the XOR-Lemma. The XOR-Lemma follows from the following technical claim

Claim A.2 (on bases and norms): *For every two orthogonal bases A and B and every vector v*

$$\mathsf{N}_1^A(v) \le N \cdot \mathsf{N}_\infty^B(v) \tag{50}$$

This technical claim has a three line proof:

1. For every orthogonal basis A, $\mathsf{N}_1^A(v) \le \sqrt{N} \cdot \mathsf{N}_2^A(v)$.
2. For every pair of orthonormal bases A and B, $\mathsf{N}_2^A(v) = \mathsf{N}_2^B(v)$.
3. For every orthogonal basis B, $\mathsf{N}_2^B(v) \le \sqrt{N} \cdot \mathsf{N}_\infty^B(v)$.

Using Claim A.2, we get

Lemma A.3 (The XOR-Lemma): $\texttt{stat}(\pi) \le \frac{1}{2} \cdot \sqrt{N} \cdot \texttt{maxbias}(\pi)$.

Proof: By the above $\texttt{stat}(\pi) = \frac{1}{2} \cdot \mathsf{N}_1^K(\pi - \mu)$ and $\texttt{maxbias}(\pi) = \sqrt{N} \cdot \mathsf{N}_\infty^F(\pi - \mu)$, whereas $\mathsf{N}_1^K(\pi - \mu) \le N \cdot \mathsf{N}_\infty^F(\pi - \mu)$. ■

Variants. Using small variations on the proof of the Claim A.2, we obtain the following.

Lemma A.4 (variants of the XOR-Lemma):

1. $\max_{x\in\{0,1\}^n}\{|\pi(x)-\mu(x)|\} \leq \mathtt{maxbias}(\pi)$.
2. $\mathtt{stat}(\pi) \leq \frac{1}{2}\cdot\sqrt{\sum_{S\neq\emptyset}\mathtt{bias}_S(\pi)^2}$, *where* $\mathtt{bias}_S(\pi)=\sum_x b_S(x)\cdot\pi(x)$.

Proof: The first part follows by using $\mathsf{N}^A_\infty(v) \leq \mathsf{N}^A_2(v)$ (instead of $\mathsf{N}^A_1(v) \leq \sqrt{N}\cdot\mathsf{N}^A_2(v)$), and obtaining $\mathsf{N}^K_\infty(\pi-\mu) \leq \sqrt{N}\cdot\mathsf{N}^F_\infty(\pi-\mu)$. The second part follows by using $\mathsf{N}^A_1(v) \leq \sqrt{N}\cdot\mathsf{N}^B_2(v)$ and $\mathsf{N}^F_2(\pi-\mu)=\sqrt{\sum_{S\neq\emptyset}\mathtt{bias}_S(\pi)^2}$. In both parts we also use $\mathtt{bias}_\emptyset(\pi-\mu)=0$. ■

Generalization to GF(p), for any prime p. The entire treatment can be generalized to distributions over $\mathrm{GF}(p)^n$. In this case, we redefine $N \stackrel{\mathrm{def}}{=} p^n$, and $\mathtt{stat}(\pi)$ denote the statistical difference between π and the uniform distribution over $\mathrm{GF}(p)^n$ (cf. Eq. (44)). Letting ω denote the p^{th} root of unity, we generalize Eq. (45) to

$$\mathtt{maxbias}(\pi) \stackrel{\mathrm{def}}{=} \max_{\beta\in\mathrm{GF}(3)^n\setminus\{0\}^n}\left\{\left|\sum_{e\in\mathrm{GF}(p)}\omega^e\cdot\pi\left(\left\{x:\textstyle\sum_{i\in[n]}\beta_i x_i\equiv e \pmod p\right\}\right)\right|\right\} \tag{51}$$

The Fourier basis is generalized analogously: The new basic consists of the functions $\{b_\beta : \beta\in\mathrm{GF}(p)^n\}$, where $b_\beta(x)=\omega^{\sum_{i\in[n]}\beta_i x_i}$. The normalized basis, denoted F, consists of the functions $f_\beta = N^{-1/2}\cdot b_\beta$. Note that, in the case of $p=2$, these definitions coincides with the definitions presented before. By following exactly the same manipulations as in the case of $p=2$, we obtain the following generalization.

Lemma A.5 (The XOR-Lemma, generalized to GF(p)): *Let π be an arbitrary distribution over* $\mathrm{GF}(p)^n$*, and let μ denote the uniform distribution over* $\mathrm{GF}(p)^n$*.*

1. $\mathtt{stat}(\pi) \leq \frac{1}{2}\cdot\sqrt{N}\cdot\mathtt{maxbias}(\pi)$.
2. $\max_{x\in\{0,1\}^n}\{|\pi(x)-\mu(x)|\} \leq \mathtt{maxbias}(\pi)$.
3. $\mathtt{stat}(\pi) \leq \frac{1}{2}\cdot\sqrt{\sum_{S\neq\emptyset}\mathtt{bias}_S(\pi)^2}$, *where* $\mathtt{bias}_S(\pi)=\sum_x b_S(x)\cdot\pi(x)$.

A.3 Yao's XOR Lemma for OBDDs

Loosely speaking, Yao's Lemma asserts that unpredictability is amplified by taking the exclusive-or of values that are individually hard to predict. The lemma holds in various computational models (cf., e.g., [12]), and essentially says that if the predicates f_1 and f_2 cannot be approximated by algorithms of a certain class any better than with success probability $\frac{1+\epsilon_1}{2}$ and $\frac{1+\epsilon_2}{2}$, respectively, then $f(y,z)=f_1(y)\oplus f_2(z)$ cannot be approximated by algorithms of a certain class any better than with success probability $\frac{1+\epsilon_1\epsilon_2}{2}$. In this appendix we provide a simple proof of this result for the case of OBDDs.

Actually, the phrasing of the following lemma avoids reference to any complexity class. It only assumes (unidirectional) on-line access to the input in the sense that the value of $F(y,z)=f_1(y)\oplus f_2(z)$ is predicted by a function of the form $G(y,z)=g_2(g_1(y),z)$, which means that the algorithm first processes y, producing $g_1(y)$, and outputs its final verdict based solely on $g_1(y)$ and

z. Indeed, the reader should consider the case that $|g_1(y)| \ll |y|$. This applies, in particular, to the bounded-width OBDD model. The actual statement is in terms of a reducibility argument. It says that G might as well have the form $\pi(g_1(y)) \oplus g_2(a,z)$, where $\pi : \{0,1\}^* \to \{0,1\}$ and $a \in \{0,1\}^*$ are fixed. This presupposes that computing $\pi \circ g_1$ is not more complex that computing g_1, and that hardwiring constants is for free. Both assumptions holds in the bounded-width OBDD model.

As is usually the case with the XOR Lemma, it is more convenient to work with the ± 1 notation. Thus, $\sigma \in \{0,1\}$ is replaced by $(-1)^\sigma$, and $\oplus$ is replaced by multiplication.

Lemma A.6 *Let $f_1, f_2 : \{0,1\}^* \to \{\pm 1\}$, and $g_1 : \{0,1\}^* \to \{0,1\}^*$, $g_2 : \{0,1\}^* \times \{0,1\}^* \to \{\pm 1\}$. Then, there exists $\pi : \{0,1\}^* \to \{\pm 1\}$ and $a \in \{0,1\}^*$ such that*

$$\begin{aligned} E_{y,z}[(f_1(y)f_2(z))g_2(g_1(y),z)] &\leq E_{y,z}[f_1(y)f_2(z)\pi(g_1(y))g_2(a,z)] \qquad (52)\\ &= E_y[f_1(y)\pi(g_1(y))] \cdot E_z[f_2(z)g_2(a,z)] \qquad (53) \end{aligned}$$

where y and z are arbitrarily distributed in $\{0,1\}^*$, but are independent of one another.

In particular, it follows that if f_1 and f_2 cannot be correlated by a width-w OBDD better than p_1 and p_2, respectively, then $f(y,z) = f_1(y)f_2(z)$ cannot be correlated by this class better than $p_1 p_2$. For our purposes, it suffices to have the (even simpler) special case in which either p_1 or p_2 equals 1.

Proof: The equality is obvious, and so we focus on the inequality. Let $p_1 = \max_\pi\{E_y[f_1(y)\pi(g_1(y))\}$ and $p_2 = \max_{a,s\in\{\pm 1\}}\{s \cdot E_z[f_2(z)g_2(a,z)]\}$.

Define $\rho : \{0,1\}^* \to \mathsf{R}$ such that $\rho(x) \stackrel{\text{def}}{=} E_z[f_2(z)g_2(x,z)]/p_2$. Note that by the definition of p_2 we have $\rho(x) \in [-1,+1]$ for every x (because otherwise $|E_z[f_2(z)g_2(x,z)]| > p_2$). Combining the definition of p_1 and a simple probabilistic fact[11], we have

$$E_y[f_1(y)\rho(g_1(y))] \leq p_1. \qquad (54)$$

Substituting $\rho(g_1(y))$ in Eq. (54), we get

$$E_y[f_1(y)E_z[f_2(z)g_2(g_1(y),z)]/p_2] \leq p_1 \qquad (55)$$

which implies

$$E_{y,z}[f_1(y)f_2(z)g_2(g_1(y),z)] \leq p_1 p_2 \qquad (56)$$

Plugging in the definitions of p_1 and p_2, we get

$$\begin{aligned} E_{y,z}[f_1(y)f_2(z)g_2(g_1(y),z)] &\leq \max_{\pi,a,s}\{s \cdot E_y[f_1(y)\pi(g_1(y))] \cdot E_z[f_2(z)g_2(a,z)]\} \qquad (57)\\ &= \max_{\pi,a}\{E_y[f_1(y)\pi(g_1(y))] \cdot E_z[f_2(z)g_2(a,z)]\} \qquad (58) \end{aligned}$$

and the lemma follows. ■

[11] The fact is that *if for every $\pi : \{0,1\}^* \to \{\pm 1\}$ it holds that $E[Y\pi(Z)] \leq p$, then the same holds for $\pi : \{0,1\}^* \to [-1,+1]$.* The proof follows by the counterpositive. Assuming that $E[Y\pi(Z)] > p$ holds for some $\pi : \{0,1\}^* \to [-1,+1]$, we first define a random process Π such that $\Pi(x) = 1$ with probability $(1+\pi(x))/2$ and $\Pi(x) = -1$ otherwise. Then, $E[Y\Pi(Z)] = E[Y\pi(Z)] > p$, because $E[\Pi(z)] = \pi(z)$, and it follows that there exists a $\pi : \{0,1\}^* \to \{-1,+1\}$ (in the support of Π) that contradicts the hypothesis.