Tight Bounds for Testing k-Linearity

Eric Blais[*] Daniel Kane[†]

August 8, 2012

Abstract

The function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is *k-linear* if it returns the sum (over $\mathbb{F}_2$) of exactly k coordinates of its input. We introduce strong lower bounds on the query complexity for testing whether a function is k-linear. We show that for any $k \leq \frac{n}{2}$, at least $k - o(k)$ queries are required to test k-linearity, and we show that when $k \approx \frac{n}{2}$, this lower bound is nearly tight since $\frac{4}{3}k + o(k)$ queries are sufficient to test k-linearity. We also show that non-adaptive testers require $2k - O(1)$ queries to test k-linearity.

We obtain our results by reducing the k-linearity testing problem to a purely geometric problem on the boolean hypercube. That geometric problem is then solved with Fourier analysis and the manipulation of Krawtchouk polynomials.

1 Introduction

What global properties of functions can we test with only a partial, local view of an unknown object? Property testing, a model introduced by Rubinfeld and Sudan [20], formalizes this question. In this model, a *property* of functions $\mathbb{F}_2^n \to \mathbb{F}_2$ is simply a subset of these functions. A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is *ϵ-far* from a property $\mathcal{P}$ if for every $g \in \mathcal{P}$, the inequality $f(x) \neq g(x)$ holds for at least an ϵ fraction of the inputs $x \in \mathbb{F}_2^n$. A *q-query ϵ-tester* for $\mathcal{P}$ is a randomized algorithm that queries a function f on at most q inputs and distinguishes with probability at least $\frac{2}{3}$ between the cases where $f \in \mathcal{P}$ and where f is ϵ-far from $\mathcal{P}$. The aim of property testing is to identify the minimum number of queries required to test various properties. For more details on property testing, we recommend the recent surveys [17, 18, 19] and the collection [13].

Linearity testing is one of the earliest success stories in property testing. The function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is *linear* if it is of the form $f(x) = \sum_{i \in S} x_i$ for some set $S \subseteq [n]$, where the sum is taken over $\mathbb{F}_2$. Equivalently, f is linear if every pair $x, y \in \mathbb{F}_2^n$ satisfies the identity $f(x) + f(y) = f(x+y)$. Blum, Luby, and Rubinfeld [7] showed that, remarkably, linearity can be ϵ-tested with only $O(1/\epsilon)$ queries. The exact query complexity of this problem has since been studied extensively [2, 3, 1, 15] and is well understood.

In this work, we study a closely related property: k-linearity. The function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is *k-linear* if it is of the form $f(x) = \sum_{i \in S} x_i$ for some set $S \subseteq [n]$ of size $|S| = k$. The k-linearity property plays a fundamental role in testing properties of boolean functions. Notably, the query complexity of the k-linearity testing problem provides a lower bound for the query complexity for testing juntas [11], testing low Fourier degree [9], testing computability by small-depth decision trees [9], and testing a number of other basic properties of boolean functions.

Our goal is to determine the *exact* query complexity of the k-linearity testing problem. As an initial observation, we note that for any $0 \leq k \leq n$, the query complexity for the k-linearity and $(n-k)$-linearity testing problems are identical. (See Proposition **??** in the appendix.) This observation lets us concentrate on the range $0 \leq k \leq \frac{n}{2}$ from now on; all our results also apply to the range $\frac{n}{2} < k \leq n$ by applying this identity.

[*] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. eblais@cs.cmu.edu

[†] Department of Mathematics, Stanford University, Stanford, CA 94305. dankane@math.stanford.edu

1.1 Previous Work

The connection between property testing and learning theory, first established by Goldreich, Goldwasser, and Ron [14], yields a simple and non-adaptive k-linearity tester with query complexity $n + O(1/\epsilon)$. For $i = 1, 2 \ldots, n$, define $e_i \in \mathbb{F}_2^n$ to be the vector with value 1 in the ith coordinate and value 0 elsewhere. The tester queries the function on the inputs $e_1, e_2, \ldots, e_n \in \mathbb{F}_2^n$. If $f(e_i) = 1$ for exactly k indices $i \in [n]$, then f is consistent with exactly one k-linear function h. We can query the function f on $O(1/\epsilon)$ additional inputs chosen uniformly and independently at random from $\mathbb{F}_2^n$ to verify that the rest of the function f is also consistent with h. This test always accepts k-linear functions, while the functions that are ϵ-far from k-linear functions fail at least one of the two steps of the test with high probability. We call this algorithm the *learning tester* for k-linearity.

Fischer, Kindler, Ron, Safra, and Samorodnitsky [11] introduced an algorithm for testing k-linearity with roughly $O(k^2)$ queries. They also showed that for $k = o(\sqrt{n})$, non-adaptive testers—that is, testers that must fix all their queries before observing the value of the function on any of those queries—require roughly $\Omega(\sqrt{k})$ queries to test k-linearity. This implies a lower bound of $\Omega(\log k)$ queries for general (i.e., possibly adaptive) k-linearity testers for the same range of values of k.

The upper bound on the query complexity for testing k-linearity was improved implicitly by the introduction of a new algorithm for testing k-juntas—that is, testing whether a function depends on at most k variables—with only $O(k \log k)$ queries [4]. By combining this junta tester with the BLR linearity test [7], one can test k-linearity with roughly $O(k \log k)$ queries.

The first lower bound for testing k-linearity that applied to all values of k was discovered by Blais and O'Donnell [6], who, as a special case of a more general theorem on testing function isomorphism, showed that non-adaptive testers need at least $\Omega(\log k)$ queries to test k-linearity.

A much stronger bound was obtained by Goldreich [12], who showed that $\Omega(k)$ queries are required to test k-linearity non-adaptively, and that general testers require at least $\Omega(\sqrt{k})$ queries for the same task. He conjectured that this last bound could be strengthened to show that $\Omega(k)$ queries are required to test k-linearity for all $1 \leq k \leq \frac{n}{2}$.[1] Goldreich's conjecture was recently verified by Blais, Brody, and Matulef [5], who proved the desired lower bound by establishing a new connection between communication complexity and property testing.

1.2 Our Results

Continuing on the line of work described above, we pose the following question: can we obtain *exact* bounds on the query complexity of the k-linearity testing problem? The results presented in this paper make significant progress on this question. Our main results are new lower bounds for general as well as for non-adaptive testing algorithms.

Theorem 1.1. *Fix $1 \leq k \leq \frac{n}{2}$. At least $k - O(k^{2/3})$ queries are required to test k-linearity.*

Theorem 1.2. *Fix $1 \leq k \leq \frac{n}{2}$. Non-adaptive testers for k-linearity need at least $2k - O(1)$ queries.*

A particularly interesting case for k-linearity testing is when $k = \frac{n}{2}$. The learning tester for $\frac{n}{2}$-linearity requires n queries, so the lower bound in Theorem 1.2 shows that no non-adaptive tester can reduce this query complexity by more than an additive constant. It is reasonable to ask whether Theorem 1.1 can be strengthened to obtain the same conclusion for adaptive testers as well. It cannot: our next result shows that there is an adaptive $\frac{n}{2}$-linearity tester that makes much fewer than n queries.

Theorem 1.3. *It is possible to test $\frac{n}{2}$-linearity with $\frac{2}{3}n + O(\sqrt{n})$ queries.*

This theorem is a special case of a more general upper bound on the query complexity for testing k-linearity for values of k that are close to $\frac{n}{2}$. The details and the proof of this more general upper bound are presented in Appendix **??**.

[1] Goldreich's results and conjecture are stated in terms of the slightly different problem of testing $\leq k$-linearity—the property of being a function that returns the sum over $\mathbb{F}_2^n$ of *at most* k variables. The $\leq k$-linearity and k-linearity problems are largely equivalent; see [5, 12] for more details.

The lower bounds in Theorems 1.1 and 1.2, as well as all previous lower bounds on the query complexity for testing $\frac{n}{2}$-linearity, proceed by establishing a lower bound on the number of queries required to distinguish $\frac{n}{2}$-linear and $(\frac{n}{2}+2)$-linear functions. Our final result shows that for this promise problem our lower bound is optimal up to the lower order error term.

Theorem 1.4. *We can distinguish $\frac{n}{2}$-linear and $(\frac{n}{2}+2)$-linear functions with $\lceil \frac{n}{2} \rceil + 1$ queries. More generally, for $\ell \geq 1$, let b be the smallest positive integer for which 2^b does not divide ℓ. It is possible to distinguish $\frac{n}{2}$-linear and $(\frac{n}{2}+2\ell)$-linear functions with $\frac{2}{3}(1-2^{-2b})n + o(n)$ queries.*

1.3 Implications

The k-linearity testing problem plays a fundamental role in the study of property testing on boolean functions. In particular, lower bounds on the query complexity of this problem imply lower bounds for the query complexity of a number of other property testing problems. Our lower bounds carry over directly to all these other problems. As a result, Theorem 1.1 sharpens several previous results. In this section, we only provide a short description of these results; the details are found in Appendix **??**.

Corollary 1.5. *Fix $1 \leq k \leq \frac{n}{2}$. At least $k - O(k^{2/3})$ queries are required to test (1) k-juntas, (2) k-sparse $\mathbb{F}_2$-polynomials, (3) functions of Fourier degree at most k, (4) functions computable by depth-k decision trees, and (5) isomorphism to the function $f : x \mapsto x_1 + \cdots + x_k$.*

A property of linear functions is called *symmetric* if it is invariant under relabeling of its variables. A symmetric property $\mathcal{P}$ of linear functions is completely characterized by the function $h_{\mathcal{P}} : \{0, 1, \ldots, n\} \to \{0, 1\}$ where $h_{\mathcal{P}}(k) = 1$ iff k-linear functions are included in $\mathcal{P}$. Define $\Gamma_{\mathcal{P}}$ to be the minimum value of $\ell \in \{0, 1, \ldots, \lfloor \frac{n}{2} \rfloor\}$ for which every value of k in the range $\ell \leq k \leq n - \ell$ satisfies $h_{\mathcal{P}}(k) = h_{\mathcal{P}}(k+2)$. This measure is closely related to the Paturi complexity of symmetric functions [16]. It also provides a lower bound on the query complexity for testing $\mathcal{P}$.

Corollary 1.6. *Let $\mathcal{P}$ be a symmetric property of linear functions. Then at least $\Gamma_{\mathcal{P}} - O(\Gamma_{\mathcal{P}}^{2/3})$ queries are required to test $\mathcal{P}$.*

1.4 Discussion of our Results

Rare are the questions in theoretical computer science for which we can obtain exact (as opposed to asymptotic) answers. The results in this paper shows that the query complexity of the k-linearity testing problem is one of those special questions. Yet, while the fundamental nature of the k-linearity testing problem causes the determination of its exact query complexity to be of intrinsic interest, two other reasons form the main motivation for the research described in this article.

First, one main reason for studying the k-linearity testing problem is to gain a better understanding of the structure of linear functions. All the previous works on this problem yielded new insights into this structure. However, the insights into the structure of linear functions have yet to be exhausted by the current line of research. Indeed, as we will discuss below, our research uncovered new connections between the problem of testing k-linearity and the geometry of the boolean hypercube.

Second, the asymptotic bounds on query complexity hide some important questions. For example, consider the following rephrasing of our main question: what is the *difference* between the query complexities of the best $\frac{n}{2}$-linearity tester and the (naïve) learning tester? An asymptotic lower bound on the query complexity of $\frac{n}{2}$-linearity testers is too weak to shed any light on this question. In stark contrast, Theorem 1.2 shows that if we restrict our attention to non-adaptive testers, the difference is at most *constant*. Furthermore, Theorem 1.3 shows that for adaptive testers the difference is *linear* in n.

1.5 Our Techniques

We reduce the problem of testing k-linear functions to a purely geometric problem on the Hamming cube. Namely, we obtain our testing lower bound by showing that affine subspaces of large dimension intersect roughly the same fraction of the middle layers of the cube. More precisely, let $W_k \subseteq \mathbb{F}_2^n$ denote the set of vectors $x \in \mathbb{F}_2^n$ of Hamming weight k. Our main technical contribution is the following result.

Lemma 1.7. *There is a constant $c > 0$ such that for any affine subspace $V \subseteq \{0,1\}^n$ of codimension $d \leq \frac{n}{2} - cn^{2/3}$,*

$$\left| \frac{|V \cap W_{\frac{n}{2}-1}|}{|W_{\frac{n}{2}-1}|} - \frac{|V \cap W_{\frac{n}{2}+1}|}{|W_{\frac{n}{2}+1}|} \right| \leq \tfrac{1}{36} 2^{-d}.$$

We prove the lemma with Fourier analysis and with the manipulation of Krawtchouk polynomials.

The proof of our lower bound for non-adaptive testers proceeds via a similar reduction to a geometric problem on the Hamming cube. See Section 4 for the details.

2 Preliminaries

2.1 Fourier Analysis

For a finite dimensional vector space V over $\mathbb{F}_2$, the *inner product* of two functions $f, g : V \to \mathbb{R}$ is $\langle f, g \rangle = \mathbf{E}_{x \in V}[f(x) \cdot g(x)]$, where the expectation is over the uniform distribution on V. The L_2 norm of f is $\|f\|_2 := \sqrt{\langle f, f \rangle}$. A *character* of V is a group homomorphism $\chi : V \to \{1, -1\}^*$. Equivalently a character is a function $\chi : V \to \{1, -1\}$ so that for any $x, y \in V$, $\chi(x + y) = \chi(x)\chi(y)$. Define $\hat{V}$ to be the set of characters of V.

For a function $f : V \to \mathbb{R}$, the *Fourier transform* of f is the function $\hat{f} : \hat{V} \to \mathbb{R}$ given by $\hat{f}(\chi) := \langle f, \chi \rangle$. The *Fourier decomposition* of f is $f(x) = \sum_{\chi \in \hat{V}} \hat{f}(\chi)\chi(x)$. A fundamental property of the Fourier transform is that it preserves the squared L_2 norm.

Fact 2.1 (Parseval's Identity)**.** *For any $f : V \to \mathbb{R}$, $\|f\|_2^2 = \sum_{\chi \in \hat{V}} \hat{f}(\chi)^2$.*

The *pushforward* of the function $f : V \to \mathbb{R}$ by the linear function $g : V \to W$ is defined by $(g_*(f))(x) := \frac{1}{|V|} \sum_{y \in g^{-1}(x)} [f(y)]$.

Fact 2.2. *For any linear function $g : V \to W$ and any function $f : V \to \mathbb{R}$, $\widehat{g_*(f)}(\chi) = \frac{1}{|W|} \hat{f}(\chi \circ g)$.*

2.2 Krawtchouk Polynomials

For $n > 0$ and $k = 0, 1, \ldots, n$, the (binary) *Krawtchouk polynomial* $K_k^n : \{0, 1, \ldots, n\} \to \mathbb{Z}$ is defined by

$$K_k^n(m) = \sum_{j=0}^{k} (-1)^j \binom{m}{j} \binom{n-m}{k-j}.$$

The generating function representation of the Krawtchouk polynomial $K_k^n(m)$ is $K_k^n(m) = [x^k]\,(1-x)^m(1+x)^{n-m}$. Krawtchouk polynomials satisfy a number of useful properties. In particular, we use the following identities in our proofs.

Fact 2.3. *Fix $n > 0$. Then*

i. *For every $2 \leq k \leq n$, $K_k^n(m) - K_{k-2}^n(m) = K_k^{n+2}(m+1)$.*

ii. $\sum_{k=0}^{n} K_k^n(m)^2 = (-1)^m K_n^{2n}(2m)$.

iii. *For every $0 \leq d \leq \frac{n}{2}$, $\sum_{j=0}^{d} \binom{d}{j} (-1)^j K_{\frac{n}{2}}^n(2j+2) = 2^{2d} K_{\frac{n}{2}-d}^{n-2d}(2)$.*

iv. *$K_n^{2n}(2m+1) = 0$ and $(-1)^m K_n^{2n}(2m)$ is positive and decreasing in* $\min\{m, n-m\}$.

Fact 2.4. *Fix $n > 0$ and $-\frac{n}{2} \leq k \leq \frac{n}{2}$. Then*

$$K^n_{\frac{n}{2}+k}(m) = \frac{2^{n-1} i^m}{\pi} \int_0^{2\pi} \sin^m \theta \cos^{n-m} \theta e^{i2k\theta} \, \mathrm{d}\theta.$$

Krawtchouk polynomials are widely used in coding theory [22] and appear in our proofs because of their close connection with the Fourier coefficients of the (Hamming weight indicator) function $I_{W_k} : \mathbb{F}_2^n \to \{0,1\}$ defined by $I_{W_k}(x) = \mathbf{1}_{|x|=k}$. With the Hamming weight of the vector $\alpha = (\alpha_1, \ldots, \alpha_n) \in \{0,1\}^n$ defined as $|\alpha| = \sum_{i=1}^n \alpha_i$, the connection is formulated as follows.

Fact 2.5. *Fix $0 \leq k \leq n$, and $\alpha \in \{0,1\}^n$. Then $\widehat{I}_{W_k}(\alpha) = 2^{-n} K^n_k(|\alpha|)$.*

For completeness, we include the proofs of Facts 2.3–2.5 in Appendix **??**.

2.3 Property Testing

The proof of Theorem 1.1 uses the following standard property testing lemma.

Lemma 2.6. *Let $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ be any two distributions over functions $\mathbb{F}_2^n \to \mathbb{F}_2$. If for every set $X \subseteq \mathbb{F}_2^n$ of size $|X| = q$ and any vector $r \in \mathbb{F}_2^q$ we have that $\left|\Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X) = r] - \Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X) = r]\right| < \frac{1}{36} 2^{-q}$, then any algorithm that distinguishes functions drawn from $\mathcal{D}_{\text{yes}}$ from those drawn from $\mathcal{D}_{\text{no}}$ with probability at least $\frac{2}{3}$ makes at least $q + 1$ queries.*

Lemma 2.6 follows from Yao's Minimax Principle [23]. The proof of this result can be found in [10, 8] and, for the reader's convenience, in Appendix **??**.

3 Proof of the General Lower Bound

Proof of Theorem 1.1. We first prove the special case where $k = \frac{n}{2} - 1$. There is a natural bijection between linear functions $\mathbb{F}_2^n \to \mathbb{F}_2$ and vectors in $\mathbb{F}_2^n$: associate $f(x) = \sum_{i \in S} x_i$ with the vector $\alpha \in \mathbb{F}_2^n$ whose coordinates satisfy $\alpha_i = 1$ iff $i \in S$. Note that $f(x) = \alpha \cdot x$.

For $0 \leq \ell \leq n$, let $W_\ell \subseteq \mathbb{F}_2^n$ denote the set of elements of Hamming weight ℓ. Fix any set $X \subseteq \mathbb{F}_2^n$ of $q < \frac{n}{2} - O(n^{2/3})$ queries and any response vector $r \in \mathbb{F}_2^q$. The set of linear functions that return the response vector r to the queries in X corresponds in our bijection to an affine subspace $V \subseteq \mathbb{F}_2^n$ of codimension q. This is because for each $x \in X$, the requirement that $f(x) = r_i$ imposes an affine linear relation on f. By Lemma 1.7, this subspace satisfies the inequality

$$\left| \frac{|V \cap W_{\frac{n}{2}-1}|}{|W_{\frac{n}{2}-1}|} - \frac{|V \cap W_{\frac{n}{2}+1}|}{|W_{\frac{n}{2}+1}|} \right| \leq \tfrac{1}{36} 2^{-q}. \tag{1}$$

Define $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ to be the uniform distributions over $(\frac{n}{2} - 1)$-linear and $(\frac{n}{2} + 1)$-linear functions, respectively. By our bijection, $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ correspond to the uniform distributions over $W_{\frac{n}{2}-1}$ and $W_{\frac{n}{2}+1}$. As a result, the probability that a function drawn from $\mathcal{D}_{\text{yes}}$ or from $\mathcal{D}_{\text{no}}$ returns the response r to the set of queries X is

$$\Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X) = r] = \frac{|V \cap W_{\frac{n}{2}-1}|}{|W_{\frac{n}{2}-1}|} \quad \text{and} \quad \Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X) = r] = \frac{|V \cap W_{\frac{n}{2}+1}|}{|W_{\frac{n}{2}+1}|}.$$

So (1) and Lemma 2.6 imply that at least $\frac{n}{2} - O(n^{2/3})$ queries are required to distinguish $(\frac{n}{2} - 1)$-linear and $(\frac{n}{2} + 1)$-linear functions. All $(\frac{n}{2} + 1)$-linear functions are $\frac{1}{2}$-far from $(\frac{n}{2} - 1)$-linear functions, so this completes the proof of the theorem for $k = \frac{n}{2} - 1$.

For other values of k, we apply a simple padding argument. When $k < \frac{n}{2} - 1$, modify $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ to be uniform distributions over k-linear and $(k + 2)$-linear functions, respectively, under the restriction that all coordinates in the sum taken from the set $[2k + 2]$. This modification with $k = \frac{n}{2} - 2$ shows that $\frac{n}{2} - O(n^{2/3})$ queries are required to distinguish $(\frac{n}{2} - 2)$- and $\frac{n}{2}$-linear functions; this implies the lower bound in the theorem for the case $k = \frac{n}{2}$. □

Proof of Lemma 1.7. For any set $A \subseteq \mathbb{F}_2^n$, define $I_A : \mathbb{F}_2^n \to \{0,1\}$ to be the indicator function for A. For a given function $f : \mathbb{F}_2^n \to \{0,1\}$, let us write $\mathbf{E}[f]$ as shorthand for $\mathbf{E}_x[f(x)]$ where the expectation is over the uniform distribution of $x \in \mathbb{F}_2^n$. Similarly, for two functions f, g, we write $\mathbf{E}[f \cdot g]$ as short-hand for $\mathbf{E}_x[f(x) \cdot g(x)]$.

For any subsets $A, B \subseteq \mathbb{F}_2^n$, $|A \cap B| = 2^n \cdot \mathbf{E}[I_A \cdot I_B]$. Since $|W_{\frac{n}{2}-1}| = |W_{\frac{n}{2}+1}| = \binom{n}{\frac{n}{2}-1}$,

$$\left| \frac{|V \cap W_{\frac{n}{2}-1}|}{|W_{\frac{n}{2}-1}|} - \frac{|V \cap W_{\frac{n}{2}+1}|}{|W_{\frac{n}{2}+1}|} \right| = \frac{2^n}{\binom{n}{\frac{n}{2}-1}} \cdot \mathbf{E}\big[I_V \cdot (I_{W_{\frac{n}{2}-1}} - I_{W_{\frac{n}{2}+1}})\big].$$

The subspace V can be defined by a set $S \subseteq [n]$ of size $|S| = d$ and an affine-linear function $f : \{0,1\}^{n-d} \to \{0,1\}^d$, where $x \in V$ iff $x_S = f(x_{\bar{S}})$. Define I_m^S and $I_m^{\bar{S}}$ to be indicator functions for $|x_S| = m$ and $|x_{\bar{S}}| = m$, respectively. Then

$$\mathbf{E}[I_V \cdot (I_{W_{\frac{n}{2}-1}} - I_{W_{\frac{n}{2}+1}})] = \sum_{m=0}^{d} \mathbf{E}\left[I_V \cdot I_m^S \cdot (I^{\bar{S}}_{\frac{n}{2}-m-1} - I^{\bar{S}}_{\frac{n}{2}-m+1})\right].$$

Let $U \subseteq \{0,1\}^S$ be the image of f. Let $d' = \dim(U)$. Define $h_m : \{0,1\}^S \to [-1,1]$ by setting $h_m(u) = \mathbf{E}_{x \in \{0,1\}^{\bar{S}}}[I_V(x,u) \cdot (I^{\bar{S}}_{\frac{n}{2}-m-1}(x) - I^{\bar{S}}_{\frac{n}{2}-m+1}(x))]$. Note that $h_m = f_*\left(I^{\bar{S}}_{\frac{n}{2}-m-1} - I^{\bar{S}}_{\frac{n}{2}-m+1}\right)$. Notice also that h_m is supported on U. We have

$$\mathbf{E}[I_V \cdot (I_{W_{\frac{n}{2}-1}} - I_{W_{\frac{n}{2}+1}})] = \sum_{m=0}^{d} \mathbf{E}\left[I_m^S \cdot h_m\right] = \sum_{m=0}^{d} \mathbf{E}\left[I_m^S \cdot \mathbf{1}_U \cdot h_m\right]. \tag{2}$$

Two applications of the Cauchy-Schwarz inequality yield

$$\sum_{m=0}^{d} \mathbf{E}\left[I_m^S \cdot \mathbf{1}_U \cdot h_m\right] \le \sum_{m=0}^{d} \|I_m^S \cdot \mathbf{1}_U\|_2 \cdot \|h_m\|_2 \le \sqrt{\sum_{m=0}^{d} \|I_m^S \cdot \mathbf{1}_U\|_2^2} \cdot \sqrt{\sum_{m=0}^{d} \|h_m\|_2^2}. \tag{3}$$

We bound the two terms on the right-hand side. The first term satisfies

$$\sum_{m=0}^{d} \|I_m^S \cdot \mathbf{1}_U\|_2^2 = \sum_m \mathbf{E}_x[I_m^S(x)^2 \cdot \mathbf{1}_U] = \mathbf{E}_x\left[\mathbf{1}_U \sum_m I_m^S(x)^2\right] = 2^{d'-d}, \tag{4}$$

where the last equality follows from the fact that for every $x \in \{0,1\}^n$, there is exactly one m for which $I_m^S(x) = 1$.

We now examine the second term. By Parseval's Identity, we have that $\|h_m\|_2^2 = \sum_{\alpha \in \{0,1\}^S} \hat{h}_m(\chi_\alpha)^2$. Suppose that the image of f has dimension $d' \le d$. Then, since h_m is a pushforward,

$$\hat{h}_m(\chi) = 2^{-d}\left(\widehat{I}^{\bar{S}}_{\frac{n}{2}-m-1}(\chi \circ f) - \widehat{I}^{\bar{S}}_{\frac{n}{2}-m+1}(\chi \circ f)\right).$$

The characters $\chi \circ f$ depend only on the restriction of χ to $f(\{0,1\}^{\bar{S}})$. Thus these characters all lie in some subspace $W \subseteq \widehat{\{0,1\}}^{\bar{S}}$ of dimension d', with each character appearing $2^{d-d'}$ times. Thus, we have that

$$\|h_m\|_2^2 = 2^{-d-d'} \sum_{\chi \in W} \left(\widehat{I}^{\bar{S}}_{\frac{n}{2}-m-1}(\chi) - \widehat{I}^{\bar{S}}_{\frac{n}{2}-m+1}(\chi)\right)^2.$$

For any set $\chi \subseteq \bar{S}$, we can apply Facts 2.5 and 2.3(i) to obtain

$$\widehat{I}^{\bar{S}}_{\frac{n}{2}-m+1}(\chi) - \widehat{I}^{\bar{S}}_{\frac{n}{2}-m-1}(\chi) = 2^{-(n-d)} K^{n-d+2}_{\frac{n}{2}-m+1}(|\chi| + 1).$$

Therefore, $\sum_{m=0}^{d} \|h_m\|_2^2 = 2^{-2n+d-d'} \sum_m \sum_{\chi \in W} K^{n-d+2}_{\frac{n}{2}-m+1}(|\chi|+1)^2$ and by Fact 2.3(ii),

$$\sum_{m=0}^{d} \|h_m\|_2^2 \leq 2^{-2n+d-d'} \sum_{\chi \in W} (-1)^{|\chi|+1} K^{2(n-d+1)}_{n-d+1}(2|\chi|+2). \tag{5}$$

There exist some d' coordinates such that the projection of W onto those coordinates is surjective. Therefore the number of elements of W with weight at most ℓ is at most $\sum_{j=1}^{\ell} \binom{d'}{j}$. We also have a similar bound on the number of elements of W of size at least $n-d-\ell$. Therefore, since by Fact 2.3(iv) the summand in (5) is decreasing in $\min(|\chi|, n-d-|\chi|)$, we have

$$\sum_{m=0}^{d} \|h_m\|_2^2 \leq 2^{-2n+d-d'+1} \sum_{j=0}^{d'} \binom{d'}{j} (-1)^{j+1} K^{2(n-d+1)}_{n-d+1}(2j+2).$$

By Fact 2.3(iii), the sum on the right-hand side evaluates to $-K^{2(n-d-d'+1)}_{n-d-d'+1}(2)$. We can then apply the generating function representation of Krawtchouk polynomials to obtain

$$\begin{aligned}
\sum_{m=0}^{d} \|h_m\|_2^2 &\leq -2^{-2n+d+d'+1}[x^{n-d-d'+1}](1-x)^2(1+x)^{2(n-d-d')} \\
&= 2^{-2n+d+d'+2}\left(\binom{2(n-d-d')}{n-d-d'} - \binom{2(n-d-d')}{n-d-d'-1}\right) \\
&= 2^{-d-d'}\Theta(n-d-d')^{-3/2} = 2^{-d-d'}O\left((n-2d)^{-3/2}\right).
\end{aligned}$$

Thus we have that

$$\begin{aligned}
\mathbf{E}[I_V \cdot (I_{W_{\frac{n}{2}+1}} - I_{W_{\frac{n}{2}-1}})] &\leq \sqrt{2^{d'-d}}\sqrt{2^{-d-d'}O\left((n-2d)^{-3/2}\right)} \\
&= 2^{-d}O\left((n-2d)^{-3/4}\right).
\end{aligned}$$

When $d = \frac{n}{2} - cn^{2/3}$ for some large enough constant $c > 0$, we therefore have $\mathbf{E}[I_V \cdot (I_{W_{\frac{n}{2}+1}} - I_{W_{\frac{n}{2}-1}})] < \frac{1}{36}\binom{n}{\frac{n}{2}-1}2^{-n-d}$ and the lemma follows. □

4 Non-Adaptive Lower Bound

The strategy for the proof of Theorem 1.2 is similar to that of the proof of the general lower bound in the last section. Once again, we reduce the problem to a geometric problem on the Hamming cube. The main difference is that in this case we prove the following lemma.

Lemma 4.1. *There is a constant $d_0 > 0$ such that for any linear subspace $V \subseteq \{0,1\}^n$ of codimension $d \leq n - d_0$,*

$$\sum_{x \in \{0,1\}^n/V} \left(\frac{|(V+x) \cap W_{\frac{n}{2}-1}|}{|W_{\frac{n}{2}-1}|} - \frac{|(V+x) \cap W_{\frac{n}{2}+1}|}{|W_{\frac{n}{2}+1}|} \right)^2 \leq \tfrac{1}{3}2^{-d}.$$

Proof. As in the last section, define $I_A : \{0,1\}^n \to \{0,1\}$ to be the indicator function for the set $A \subseteq \{0,1\}^n$. To prove Lemma 4.1, we want to show that

$$\sum_{x \in \{0,1\}^n/V} \left(\frac{\mathbf{E}[I_{V+x} \cdot I_{W_{\frac{n}{2}-1}}]}{\mathbf{E}[I_{W_{\frac{n}{2}-1}}]} - \frac{\mathbf{E}[I_{V+x} \cdot I_{W_{\frac{n}{2}+1}}]}{\mathbf{E}[I_{W_{\frac{n}{2}+1}}]} \right)^2 \leq \tfrac{1}{3}2^{-d}.$$

Let $D_{\frac{n}{2}} = I_{W_{\frac{n}{2}-1}} - I_{W_{\frac{n}{2}+1}}$, and note that $\mathbf{E}[I_{W_{\frac{n}{2}-1}}] = \mathbf{E}[I_{W_{\frac{n}{2}+1}}] = \binom{n}{\frac{n}{2}-1}/2^n$. Then the above inequality is equivalent to

$$\sum_{x \in \{0,1\}^n/V} \mathbf{E}[I_{V+x} \cdot D_{\frac{n}{2}}]^2 \leq \tfrac{1}{3} 2^{-d} \cdot \left(\frac{\binom{n}{\frac{n}{2}-1}}{2^n} \right)^2 .$$

Let $\pi : \{0,1\}^n \to \{0,1\}^n/V$ be the projection map. Notice that $\mathbf{E}[I_{V+x} \cdot D_{\frac{n}{2}}] = \pi_* D_{\frac{n}{2}}(x)$. By Parseval's Theorem and Fact 2.2,

$$\mathbf{E}_{x \in \{0,1\}^n/V}[\pi_* D_{\frac{n}{2}}(x)^2] = |\pi_* D_{\frac{n}{2}}|_2^2 = \sum_{\chi \in \widehat{\{0,1\}^n/V}} \widehat{\pi_* D_{\frac{n}{2}}}(\chi) = 2^{d-n} \sum_{\chi \in V^\perp} \widehat{D_{\frac{n}{2}}}(\chi).$$

Where above $V^\perp$ is the set of pullbacks of $\widehat{\{0,1\}^n/V}$ to $\widehat{\{0,1\}^n}$, which is the space of characters of $\{0,1\}^n$ that are trivial on V.

By Fact 2.5, $\widehat{D}_{\frac{n}{2}}(\chi) = 2^{-n} K^{n+2}_{\frac{n}{2}+1}(|\chi|+1)$. By Fact 2.3(iv), the absolute value of this is 0 for $|\chi|$ even and otherwise decreasing in $\min(|\chi|, n-|\chi|)$. Since there are at most $2\sum_{j=0}^{\ell} \binom{d}{j}$ $\chi \in V^\perp$ with $\min(|\chi|, n-|\chi|) < \ell$, the above sum is less than it would be if there were 2 $\chi \in V^\perp$ with $|\chi| = 0$, $2\left(\binom{d}{2} + \binom{d}{1}\right)$ with $|\chi| = 2$, $2\left(\binom{d}{4} + \binom{d}{3}\right)$ with $|\chi| = 4$, and so on. Hence

$$\sum_{x \in \{0,1\}^n/V} \mathbf{E}[I_{V+x} \cdot D_{\frac{n}{2}}]^2 \leq 2^{2-2n-d} \sum_{m=0}^{d} \binom{d}{m} K^{n+2}_{\frac{n}{2}+1}(m+1)^2.$$

By Fact 2.4, we can expand the sum on the right-hand side of the inequality into a double integral. Namely,

$$\begin{aligned} &\sum_{m=0}^{d} \binom{d}{m} K^{n+2}_{\frac{n}{2}+1}(m+1)^2 \\ &\quad = \frac{2^{2n}}{\pi^2} \iint \sum_{m=0}^{d} \binom{d}{m} (-1)^m \sin^{m+1}\theta \sin^{m+1}\phi \cos^{n-m+1}\theta \cos^{n-m+1}\phi \, \mathrm{d}\theta \, \mathrm{d}\phi. \end{aligned}$$

As we show in Proposition **??**, we can manipulate the trigonometric functions and apply the Cauchy-Schwarz inequality to obtain

$$\sum_{m=0}^{d} \binom{d}{m} K^{n+2}_{\frac{n}{2}+1}(m+1)^2 \leq O\big(2^{2n} d^{-\frac{1}{2}} (n-d+1)^{-\frac{3}{2}}\big). \tag{6}$$

Using this bound, we obtain

$$\sum_{x \in \{0,1\}^n/V} \mathbf{E}[I_{V+x} \cdot D_{\frac{n}{2}}]^2 \leq O\left(2^{-d} d^{-\frac{1}{2}} (n-d+1)^{-\frac{3}{2}}\right).$$

Note that $2^{-d} \cdot \left(\frac{\binom{n}{\frac{n}{2}-1}}{2^n} \right)^2 = \Theta(2^{-d} n^{-1/2})$. If $d < n/2$, $\sum_{x \in \{0,1\}^n/V} \mathbf{E}[I_{V+x} \cdot D_{\frac{n}{2}}]^2$ is $O(2^{-d} n^{-3/2})$, which is too small. Otherwise it is $O(2^{-d} n^{-1/2} (n-d+1)^{-3/2})$, which is too small as long as $n-d$ is bigger than a sufficiently large constant. □

5 Upper Bounds

We provide a sketch of the proofs of Theorems 1.3 and 1.4 in this section.

Let us begin by describing the algorithm for distinguishing $\frac{n}{2}$-linear and $(\frac{n}{2}+2)$-linear functions. The starting point for this algorithm is an elementary observation: $\frac{n}{2} \not\equiv \frac{n}{2} + 2 \pmod 4$. For a set $S \subseteq [n]$, let

$x_S \in \mathbb{F}_2^n$ be the vector with value 1 at each coordinate in S and 0 in the remaining coordinates. Query $f(x_{\{1,2\}}), f(x_{\{3,4\}}), \ldots, f(x_{\{n-1,n\}})$. Let m denote the number of queries that returned 1. Define the set $T = \{2i : f(x_{\{2i-1,2i\}}) = 0\}$. Query $f(x_T)$; if $f(x_T) = 1$, increment m by 2. When f is k-linear, we have $m \equiv k \pmod 4$ and this algorithm completes the proof of the first claim in Theorem 1.4.

The algorithm that proves the more general claim in Theorem 1.4 is obtained by applying the same approach recursively. When $b > 0$ is the minimum integer for which $2^b \nmid \ell$ and f is k-linear, we can determine the value of k modulo 2^b in b rounds and thereby distinguish between the cases where $k = \frac{n}{2}$ and $k = \frac{n}{2} + 2\ell$.

Finally, to complete the proof of Theorem 1.3, we essentially combine the Blum–Luby–Rubinfeld (BLR) linearity test [7] with the algorithm described above. The BLR test rejects functions that are far from linear; after that, the problem of testing k-linearity is essentially equivalent to that of distinguishing k-linear from functions that are k'-linear for some $k' \neq k$.

References

[1] Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos Kiwi, and Madhu Sudan. Linearity testing in characteristic two. *IEEE Trans. on Information Theory*, 42(6):1781 –1795, 1996.

[2] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proc. of the 25th Symposium on Theory of Computing*, pages 294–304, 1993.

[3] Mihir Bellare and Madhu Sudan. Improved non-approximability results. In *Proc. of the 26th Symposium on Theory of Computing*, pages 184–193, 1994.

[4] Eric Blais. Testing juntas nearly optimally. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 2009.

[5] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. In *Proc. of the 26th Conference on Computational Complexity*, 2011.

[6] Eric Blais and Ryan O'Donnell. Lower bounds for testing function isomorphism. In *Proc. of the 25th Conference on Computational Complexity*, pages 235–246, 2010.

[7] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47:549–595, 1993.

[8] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Nearly tight bounds for testing function isomorphism. In *Proc. 22nd Symposium on Discrete Algorithms*, pages 1683–1702, 2011.

[9] Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. In *Proc. 48th Symposium on Foundations of Computer Science*, pages 549–558, 2007.

[10] Eldar Fischer. The art of uninformed decisions. *Bulletin of the EATCS*, 75:97–126, 2001.

[11] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *J. Comput. Syst. Sci.*, 68(4):753–787, 2004.

[12] Oded Goldreich. On testing computability by small width OBDDs. In *Proc. of the 13th international conference on Approximation, and 14 the International conference on Randomization, and combinatorial optimization: algorithms and techniques*, APPROX/RANDOM'10, pages 574–587, 2010.

[13] Oded Goldreich, editor. *Property Testing: Current Research and Surveys*, volume 6390 of *LNCS*. Springer, 2010.

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

[15] Tali Kaufman, Simon Litsyn, and Ning Xie. Breaking the ϵ-soundness bound of the linearity test over GF(2). *SIAM J. on Computing*, 39:1988–2003, 2010.

[16] Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions (preliminary version). In *Proc. STOC '92*, pages 468–474, 1992.

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

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

[19] Ronitt Rubinfeld and Asaf Shapira. Sublinear time algorithms. Technical Report TR11-013, ECCC, 2011.

[20] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[21] Gábor Szegő. *Orthogonal Polynomials*, volume 23 of *Colloquium Publications*. AMS, fourth edition, 1975.

[22] Jacobus H. V. Van Lint. *Introduction to Coding Theory*, volume 86 of *Graduate Texts in Mathematics*. Springer, third edition, 1999.

[23] Andrew C. Yao. Probabilistic computations: towards a unified measure of complexity. In *Proc. 18th Sym. on Foundations of Comput. Sci.*, pages 222–227, 1977.

A Krawtchouk Polynomials

We include the proofs for the facts related to Krawtchouk polynomials that we introduced in Section 2. All these facts follow from elementary manipulations of the generating function representation of Krawtchouk polynomials. We include these proofs for the convenience of the reader; for a more complete reference on Krawtchouk polynomials, see [21, 22].

Fact 2.3 (Restated)**.** *Fix $n > 0$. Then*

i. For every $2 \le k \le n$, $K_k^n(m) - K_{k-2}^n(m) = K_k^{n+2}(m+1)$.

ii. $\sum_{k=0}^n K_k^n(m)^2 = (-1)^m K_n^{2n}(2m)$.

iii. For every $0 \le d \le \frac{n}{2}$, $\displaystyle\sum_{j=0}^{d} \binom{d}{j} (-1)^j K_{\frac{n}{2}}^n(2j+2) = 2^{2d} K_{\frac{n}{2}-d}^{n-2d}(2)$.

iv. $K_n^{2n}(2m+1) = 0$ and $(-1)^m K_n^{2n}(2m)$ is positive and decreasing in $\min\{m, n-m\}$.

Proof. We prove each statement individually.

i. The first statement follows directly from the generating function representation of Krawtchouk polynomials.

$$\begin{aligned} K_k^n(m) - K_{k-2}^n(m) &= \left([x^k]\ (1-x)^m(1+x)^{n-m}\right) - \left([x^{k-2}]\ (1-x)^m(1+x)^{n-m}\right) \\ &= [x^k]\ (1-x)^m(1+x)^{n-m}(1-x^2) \\ &= [x^k]\ (1-x)^{m+1}(1+x)^{n-m+1} = K_k^{n+2}(m+1). \end{aligned}$$

ii. By some more elementary manipulation of generating functions, we have

$$\begin{aligned} K_k^n(m) &= [x^k]\ (1-x)^m(1+x)^{n-m} \\ &= [x^{-k}]\ (1-\tfrac{1}{x})^m(1+\tfrac{1}{x})^{n-m} \\ &= [x^{n-k}]\ (x-1)^m(x+1)^{n-m} = (-1)^m K_{n-k}^n(m). \end{aligned}$$

Therefore,

$$\sum_{k=0}^{n} K_k^n(m)^2 = (-1)^m \sum_{k=0}^{n} K_k^n(m) K_{n-k}^n(m).$$

The Cauchy product of two sequences $\{a_0, a_1, \ldots\}$ and $\{b_0, b_1, \ldots\}$ is

$$\Big(\sum_{n\geq 0} a_n\Big)\Big(\sum_{n\geq 0} b_n\Big) = \sum_{n\geq 0}\Big(\sum_{k=0}^{n} a_k b_{n-k}\Big).$$

Let $a_k = b_k = [x^k]\ (1-x)^m(1+x)^{n-m}$. Then $(\sum_{n\geq 0} a_n) = (1-x)^m(1+x)^{n-m}$ and

$$\sum_{k=0}^{n} K_k^n(m) K_{n-k}^n(m) = [x^n]\ (1-x)^{2m}(1+x)^{2(n-m)} = K_n^{2n}(2m).$$

iii. Considering generating functions and applying the binomial theorem, we get

$$\begin{aligned} \sum_{j=0}^{d} \binom{d}{j}(-1)^j K_n^{2n}(2j+2) &= [x^n]\ \sum_{j=0}^{d}\binom{d}{j}(-1)^j(1-x)^{2j+2}(1+x)^{2n-2j-2} \\ &= [x^n]\ (1-x)^2(1+x)^{2n-2d-2}\sum_{j=0}^{d}\binom{d}{j}\big(-(1-x)^2\big)^j\big((1+x)^2\big)^{d-j} \\ &= [x^n]\ (1-x)^2(1+x)^{2n-2d-2}(4x)^d = 2^{2d}K_{n-d}^{2(n-d)}(2). \end{aligned}$$

iv. By the last statement, $K_n^{2n}(2m+1)$ is pure imaginary. Since it is also real, it must be 0. The last statement also yields

$$\begin{aligned} (-1)^m K_n^{2n}(2m) &= \frac{2^{2n-1}}{\pi}\int_0^{2\pi} \sin^{2m}(\theta)\cos^{2n-2m}(\theta)d\theta \\ &= \frac{2^{2n-2}}{\pi}\int_0^{2\pi} \sin^{2m}(\theta)\cos^{2n-2m} + \cos(\theta)^{2m}\sin(\theta)^{2n-2m}(\theta)d\theta. \end{aligned}$$

By AM-GM, for fixed n, the integrand is a decreasing function of $\min\{m, n-m\}$.

□

Fact A.1. *Fix $n > 0$ and $-\frac{n}{2} \leq k \leq \frac{n}{2}$. Then*

$$K_{\frac{n}{2}+k}^n(m) = \frac{2^{n-1}}{\pi} i^m \int_0^{2\pi} \sin^m\theta\cos^{n-m}\theta e^{i2k\theta}\,\mathrm{d}\theta.$$

Proof. By elementary manipulation of generating functions, we obtain

$$\begin{aligned} K_{\frac{n}{2}+k}^n(m) &= [x^{\frac{n}{2}+k}]\ (1-x)^m(1+x)^{n-m} \\ &= [x^k]\ (\tfrac{1}{\sqrt{x}}-\sqrt{x})^m(\tfrac{1}{\sqrt{x}}+\sqrt{x})^{n-m} \\ &= [x^{-2k}]\ (x-\tfrac{1}{x})^m(x+\tfrac{1}{x})^{n-m}. \end{aligned}$$

Applying Cauchy's integral formula to this expression, we get

$$K^n_{\frac{n}{2}+k}(m) \quad = \quad \frac{1}{2\pi}\int_0^{2\pi} (e^{i\theta}-e^{-i\theta})^m (e^{i\theta}+e^{-i\theta})^{n-m} e^{i2k\theta}\,\mathrm{d}\theta.$$

From the trigonometric identities $\sin\theta = \frac{e^{i\theta}-e^{-i\theta}}{2i}$ and $\cos\theta = \frac{e^{i\theta}+e^{-i\theta}}{2}$, we get

$$K^n_{\frac{n}{2}+k}(m) = \frac{2^n}{2\pi} i^m \int_0^{2\pi} \sin^m\theta \cos^{n-m}\theta e^{i2k\theta}\,\mathrm{d}\theta.$$

□

Fact 2.5 (Restated)**.** *Fix $0 \le k \le n$ and $\alpha \in \{0,1\}^n$. Then*

$$\widehat{I}_{W_k}(\alpha) = 2^{-n} K_k(|\alpha|).$$

Proof. The Fourier coefficient of I_{W_k} at α is

$$\begin{aligned} \widehat{I}_{W_k}(\alpha) &= 2^{-n} \sum_{x\in\{0,1\}^n : |x|=k} (-1)^{\alpha\cdot x} \\ &= 2^{-n} \sum_{j=0}^{k} (-1)^j \binom{|\alpha|}{j}\binom{n-|\alpha|}{k-j} \\ &= 2^{-n} K^n_k(|\alpha|). \end{aligned}$$

□

B Property Testing Lemmas

We complete the proofs of Lemma 2.6 and a similar lemma for proving lower bounds on the query complexity for non-adaptive testers. We also provide the proof of the claim in the introduction that the k-linearity and $(n-k)$-linearity testing problems have the same query complexity.

Lemma 2.6. (Restated) *Let $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ be any two distributions over functions $\{0,1\}^n \to \{0,1\}$. If for every set $X \subseteq \{0,1\}^n$ of size $|X| = q$ and any vector $r \in \{0,1\}^q$ we have that*

$$\left| \Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X)=r] - \Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X)=r] \right| < \tfrac{1}{36}\, 2^{-q},$$

then any algorithm that distinguishes functions drawn from $\mathcal{D}_{\text{yes}}$ from those drawn from $\mathcal{D}_{\text{no}}$ with probability at least $\frac{2}{3}$ makes at least $q+1$ queries.

Proof. Define $\mathcal{D}$ to be the distribution obtained by drawing a function from $\mathcal{D}_{\text{yes}}$ or from $\mathcal{D}_{\text{no}}$, each with probability 1/2. By Yao's Minimax Principle[23], to prove the lemma it suffices to show that any deterministic testing algorithm needs at least $q+1$ queries to distinguish functions drawn from $\mathcal{D}_{\text{yes}}$ or from $\mathcal{D}_{\text{no}}$ with probability at least 2/3.

A deterministic testing algorithm can be described by a decision tree with a query $x \in \{0,1\}$ at each internal node and a decision to accept or reject at every leaf. Each boolean function f defines a path through the tree according to the value of $f(x)$ at each internal node.

Consider a testing algorithm that makes at most q queries. Then it has depth at most q and at most 2^q leaves. Let us call a leaf ℓ *negligible* if the probability that a function $f \sim \mathcal{D}$ defines a path that terminates at ℓ is at most $\frac{1}{12}2^{-q}$. The total probability that $f \sim \mathcal{D}$ defines a path to a negligible leaf is at most $\frac{1}{12}$.

Fix ℓ to be some non-negligible leaf. This leaf corresponds to a set $X \subseteq \{0,1\}^n$ of q queries and a vector $r \in \{0,1\}^q$ of responses; a function f defines a path to the leaf ℓ iff $f(X) = r$. Since ℓ is non-negligible, $\Pr_{f\sim\mathcal{D}}[f(X) = r] > \frac{1}{12}2^{-q}$. So by the hypothesis of the lemma,

$$\left| \Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X)=r] - \Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X)=r] \right| \le \tfrac{1}{36}\, 2^{-q} < \tfrac{1}{3} \Pr_{f\sim\mathcal{D}}[f(X)=r].$$

Then by Bayes' theorem

$$\left|\Pr_{f\sim\mathcal{D}}[f\in\mathcal{P}\mid f(X)=r]-\Pr_{f\sim\mathcal{D}}[f\ \epsilon\text{-far from }\mathcal{P}\mid f(X)=r]\right|$$
$$=\left|\frac{\Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X)=r]-\Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X)=r]}{2\Pr_{f\sim\mathcal{D}}[f(X)=r]}\right|<\frac{1}{6}.$$

Therefore, the probability that the testing algorithm correctly classifies a function $f\sim\mathcal{D}$ that lands at a non-negligible leaf ℓ is less than $\frac{7}{12}$. So even if the algorithm correctly classifies all functions that land in negligible leaves, it still correctly classifies f with probability less than $\frac{11}{12}\cdot\frac{7}{12}+\frac{1}{12}<\frac{2}{3}$, so it is not a valid tester for $\mathcal{P}$. □

Lemma B.1. *Let $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ be any two distributions over functions $\mathbb{F}_2^n\to\mathbb{F}_2$. If for every set $X\subseteq\mathbb{F}_2^n$ of size $|X|=q$ we have that*

$$\sum_{r\in\{0,1\}^q}\left(\Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X)=r]-\Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X)=r]\right)^2<\tfrac{1}{9}2^{-q},$$

then any non-adaptive algorithm that distinguishes functions drawn from $\mathcal{D}_{\text{yes}}$ from those drawn from $\mathcal{D}_{\text{no}}$ with probability at least $\frac{2}{3}$ makes at least $q+1$ queries.

Proof. As in the proof of Lemma 2.6, let $\mathcal{D}$ denote the distribution that obtained by drawing a function from $\mathcal{D}_{\text{yes}}$ or from $\mathcal{D}_{\text{no}}$, each with probability $\frac{1}{2}$. By Yao's Minimax Principle, the proof is completed by showing that any deterministic non-adaptive testing algorithm requires at least $q+1$ queries to distinguish functions drawn from $\mathcal{D}_{\text{yes}}$ or $\mathcal{D}_{\text{no}}$ with probability at least $\frac{2}{3}$.

A deterministic non-adaptive testing algorithm queries all functions on a fixed set X of queries, and must accept or reject strictly based on the values of $f(X)$. When $|X|=q$, the condition in the lemma and the Cauchy-Schwarz inequality imply that

$$\sum_{r\in\{0,1\}^q}\left|\Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X)=r]-\Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X)=r]\right|<\frac{1}{3}. \tag{7}$$

This completes, the proof, since the maximum success probability of the algorithm is

$$\sum_{r\in\{0,1\}^q}\max\left\{\Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X)=r],\Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X)=r]\right\}\le$$
$$\tfrac{1}{2}+\tfrac{1}{2}\sum_{r\in\{0,1\}^q}\left|\Pr_{f\sim\mathcal{D}_{\text{yes}}}[f(X)=r]-\Pr_{f\sim\mathcal{D}_{\text{no}}}[f(X)=r]\right|<\tfrac{2}{3}.$$

□

Proposition B.2. *Fix $0\le k\le n$. For any $0<\epsilon<\frac{1}{2}$, the query complexities for testing k-linearity and $(n-k)$-linearity are identical.*

Proof. Let $f:\mathbb{F}_2^n\to\mathbb{F}_2$ be the function being tested for k-linearity. Let $g:\mathbb{F}_2^n\to\mathbb{F}_2$ be the function defined by setting $g(x)=f(x)+\chi_{[n]}(x)$, where $\chi_{[n]}$ is the parity function over all bits. Then if $f:x\mapsto\sum_{i\in S}x_i$, we have $g:x\mapsto\sum_{i\in[n]\setminus S}x_i$. In particular, if f is k-linear, then g is $(n-k)$-linear. Furthermore, if f is ϵ-far from k-linear, then g is also ϵ-far from $(n-k)$-linear. And, lastly, for any $x\in\mathbb{F}_2^n$, we can obtain the value of $g(x)$ by querying the value of f on a single input, namely, by querying $f(x)$. Therefore, we can use a $(n-k)$-linearity tester to test if f is k-linear without any loss in the query complexity.

The same argument obviously also shows that we can use a k-linearity tester to test $(n-k)$-linearity without any loss in the query complexity; the proposition follows. □

C Proofs of Corollaries 1.5 and 1.6

Definition C.1 (Juntas)**.** The function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a *k-junta* if there is a set $J \subseteq [n]$ of size $|J| \leq k$ such that for each $x, y \in \mathbb{F}_2^n$ that satisfy $x_i = y_i$ for every $i \in J$, the identity $f(x) = f(y)$ holds.

Definition C.2 (Sparse polynomials)**.** The function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ has a unique representation as a multivariate polynomial over the variables $x_1, \ldots, x_n$. If this representation has at most k non-zero coefficients, we say that f is a *k-sparse $\mathbb{F}_2$-polynomial.*

Definition C.3 (Fourier degree)**.** The *Fourier degree* of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is the maximum Hamming weight of any $\alpha \in \mathbb{F}_2$ such that the Fourier coefficient $\hat{f}(\chi_\alpha)$ is non-zero.

Definition C.4 (Decision trees)**.** A *decision tree* is a model of computation that can be represented as a rooted binary tree with each internal node of the tree labeled with an index $i \in [n]$ and the two edges going from a node to its children labeled with $0, 1$. The leaves of the tree are also labeled with $0, 1$. A decision tree *computes* the function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ if for every $x \in \mathbb{F}_2^n$, the path from the root to a leaf followed by taking the edge x_i at each node i leads to a leaf labeled with $f(x)$. The *depth* of a tree is the maximum length of any path from its root to one of its leaves.

Definition C.5 (Function isomorphism)**.** Given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, the *$f$-isomorphism* property includes all functions that are equal to f up to relabeling of the n variables. In other words, g is isomorphic to f if there exists a permutation $\pi \in \mathcal{S}_n$ such that for every $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$, $g(x_1, \ldots, x_n) = f(x_{\pi(1)}, \ldots, x_{\pi(n)})$.

Corollary 1.5. (Restated) *Fix $1 \leq k \leq \frac{n}{2}$. At least $k - O(k^{2/3})$ queries are required to test*

(1) k-juntas,

(2) k-sparse $\mathbb{F}_2$-polynomials,

(3) functions of Fourier degree at most k,

(4) functions computable by depth-k decision trees, and

(5) isomorphism to the function $f : x \mapsto x_1 + \cdots + x_k$.

Proof. Recall that in our proof of Theorem 1.1, we showed that at least $k - O(k^{\frac{2}{3}})$ queries are required to distinguish k-linear and $(k+2)$-linear functions.

As we can easily verify, k-linear functions are k-juntas, they are k-sparse $\mathbb{F}_2$-polynomials, they have Fourier degree at most k, and they can be computed by a (complete) decision tree of depth k. To complete the proof of cases (1)–(4) of the corollary, it suffices to show that $(k+2)$-linear functions are $\frac{1}{2}$-far from those same properties. This is indeed the case, as Fischer et al. [11] showed for the k-junta property and Diakonikolas et al. [9] showed for the other three properties.

Finally, case (5) of the corollary follows immediately from the observation that the set of functions isomorphic to the function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ defined by $f(x) = x_1 + \cdots + x_k$ is exactly the set of k-linear functions. □

Corollary 1.6 (Restated)**.** *Let $\mathcal{P}$ be a symmetric property of linear functions. Then at least $\Gamma_{\mathcal{P}} - O(\Gamma_{\mathcal{P}}^{2/3})$ queries are required to test $\mathcal{P}$.*

Proof. Once again, recall that the proof of Theorem 1.1 shows that at least $k - O(k^{\frac{2}{3}})$ queries are required to distinguish k-linear and $(k+2)$-linear functions. This also implies that the same number of queries are required to distinguish $(n-k)$-linear and $(n-k-2)$-linear functions.

By definition of $\Gamma_{\mathcal{P}}$, at least one of the inequalities $h_{\mathcal{P}}(\Gamma_{\mathcal{P}} - 1) \neq h_{\mathcal{P}}(\Gamma_{\mathcal{P}} + 1)$ or $h_{\mathcal{P}}(n - \Gamma_{\mathcal{P}} - 1) \neq h_{\mathcal{P}}(n - \Gamma_{\mathcal{P}} + 1)$ must hold. In either case, the corollary follows from the lower bounds above. □

D Proof of the Non-Adaptive Lower Bound

Proof of Theorem 1.2. The proof of this theorem is very similar to the proof of Theorem 1.1. Let $k = \frac{n}{2} - 1$. Recall that linear functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be represented as $f : x \mapsto \alpha \cdot x$ for some $\mathbb{F}_2^n$. This representation gives a natural bijection between the set of linear functions and $\mathbb{F}_2^n$. Let $W_\ell \subseteq \mathbb{F}_2^n$ denote the set of elements of Hamming weight ℓ. For any set $X \subseteq \mathbb{F}_2^n$ of $q < n - O(1)$ queries and any response vector $r \in \mathbb{F}_2^q$, the set of linear functions that gives the response r to the queries X corresponds to an affine subspace $V + x \subseteq \mathbb{F}_2^n$ of codimension q. From Lemma 4.1,

$$\sum_{x \in \mathbb{F}_2^n / V} \left(\frac{|(V+x) \cap W_{\frac{n}{2}-1}|}{|W_{\frac{n}{2}-1}|} - \frac{|(V+x) \cap W_{\frac{n}{2}+1}|}{|W_{\frac{n}{2}+1}|} \right)^2 \le \tfrac{1}{3} 2^{-d}. \tag{8}$$

Define $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ to be uniform distributions over $(\frac{n}{2}-1)$-linear and $(\frac{n}{2}+1)$-linear functions, respectively. These distributions correspond by our bijection to the uniform distributions over $W_{\frac{n}{2}-1}$ and $W_{\frac{n}{2}+1}$, and so (**??**) implies that

$$\sum_{r \in \mathbb{F}_2^q} \left(\Pr_{f \sim \mathcal{D}_{\text{yes}}} [f(X) = r] - \Pr_{f \sim \mathcal{D}_{\text{no}}} [f(X) = r] \right)^2 \le \tfrac{1}{3} 2^{-d}. \tag{9}$$

By Lemma **??**, any non-adaptive algorithm that distinguishes $(\frac{n}{2}-1)$-linear from $(\frac{n}{2}+1)$-linear functions must therefore make at least $n - O(1)$ queries. This gives the desired lower bound for testing $(\frac{n}{2}-1)$-linearity. We apply the same padding argument as in the proof of Theorem 1.1 to get the lower bound for the other values of k. □

We conclude the section with the proof of Proposition **??** that was used in the proof of Lemma 4.1.

Proposition D.1. *Fix $0 \le m \le d \le n$. Then*

$$\sum_{m=0}^{d} \binom{d}{m} K_{\frac{n}{2}+1}^{n+2}(m+1)^2 \le O\big(2^{2n} d^{-\frac{1}{2}} (n-d+1)^{-\frac{3}{2}}\big).$$

Proof. By Fact 2.4 and manipulation of the integrand, we obtain

$$\begin{aligned}
&\sum_{m=0}^{d} \binom{d}{m} K_{\frac{n}{2}+1}^{n+2}(m+1)^2 \\
&\quad = \frac{2^{2n}}{\pi^2} \iint \sum_{m=0}^{d} \binom{d}{m} (-1)^m \sin^{m+1}\theta \sin^{m+1}\phi \cos^{n-m+1}\theta \cos^{n-m+1}\phi \, \mathrm{d}\theta \, \mathrm{d}\phi \\
&\quad = \frac{2^{2n}}{\pi^2} \iint \sin\theta \sin\phi \cos^{n-d+1}\theta \cos^{n-d+1}\phi (\cos\theta\cos\phi - \sin\theta\sin\phi)^d \, \mathrm{d}\theta \, \mathrm{d}\phi \\
&\quad = \frac{2^{2n}}{\pi^2} \iint \sin\theta \sin\phi \cos^{n-d+1}\theta \cos^{n-d+1}\phi \cos^d(\theta - \phi) \, \mathrm{d}\theta \, \mathrm{d}\phi.
\end{aligned}$$

Letting $\psi = \theta - \phi$ this is

$$\frac{2^{2n}}{\pi^2} \iint \sin(\psi + \phi) \sin\phi \cos^{n-d+1}(\psi + \phi) \cos^{n-d+1}\phi \cos^d(\psi) \, \mathrm{d}\phi \, \mathrm{d}\psi.$$

Next we bound the inner integral using Cauchy-Schwarz to obtain the upper bound

$$\frac{2^{2n}}{\pi^2} \int |\cos^d(\psi)| \left(\int \sin^2(\psi + \phi) \cos^{2(n-d+1)}(\psi + \phi) \, \mathrm{d}\phi \right)^{1/2} \left(\int \sin^2\phi \cos^{2(n-d+1)}\phi \, \mathrm{d}\phi \right)^{1/2} \mathrm{d}\psi.$$

Algorithm 1 (k vs. $k+2\ell$)-Linearity Tester

```
Initialize $S_0 = \{1, \dots, n\}$ and $m = 0$.
for $r = 1, \dots, b-1$ do
  Initialize $S_r = \emptyset$.
  if $|S_{r-1}|$ is odd then
    Choose $i \in S_{r-1}$ and update $S_{r-1} = S_{r-1} \setminus \{i\}$.
    Set $m = m + f(\{i\}) \cdot 2^{r-1}$.
  end if
  Let $M$ be a random matching of $S_{r-1}$.
  for each pair $(i,j) \in M$ do
    if $f(\{i,j\}) = 1$ then
      Increment $m = m + 2^{r-1}$.
    else
      Update $S_r = S_r \cup \{i\}$.
    end if
  end for
end for
Output "$f$ is $k$-linear" iff $m + 2^{b-1} f(S_{b-1}) \equiv k \pmod{2^b}$.
```

This is

$$\frac{2^{2n}}{\pi^2}\left(\int |\cos^d(\psi)|\,\mathrm{d}\psi\right)\left(\int \sin^2\phi\cos^{2(n-d+1)}\phi\,\mathrm{d}\phi\right).$$

Now

$$\begin{aligned}\frac{1}{2\pi}\left(\int \sin^2\phi\cos^{2(n-d+1)}\phi\,\mathrm{d}\phi\right) \qquad &= 2^{-2(n-d+1)-1}\left(\binom{2(n-d+1)}{n-d+1} - \binom{2(n-d+1)}{n-d}\right)\\ &= \Theta\left((n-d+1)^{-3/2}\right).\end{aligned}$$

If d is even, we have that

$$\frac{1}{2\pi}\int |\cos^d(\psi)|\,\mathrm{d}\psi = 2^{-d}\binom{d}{d/2} = \Theta\left(d^{-1/2}\right).$$

If d is odd, then $\int |\cos^d(\psi)|\,\mathrm{d}\psi$ is bounded between $\int |\cos^{d+1}(\psi)|\,\mathrm{d}\psi$ and $\int |\cos^{d-1}(\psi)|\,\mathrm{d}\psi$, so in this case also it is $\Theta\left(d^{-1/2}\right)$. □

E Proof of Theorems 1.3 and 1.4

Theorem 1.4 (Restated)**.** *We can distinguish $\frac{n}{2}$- and $(\frac{n}{2}+2)$-linear functions with $\lceil \frac{n}{2} \rceil + 1$ queries. More generally, for $\ell \geq 1$, let b be the smallest positive integer for which 2^b does not divide ℓ. It is possible to distinguish $\frac{n}{2}$- and $(\frac{n}{2}+2\ell)$-linear functions with $\frac{2}{3}(1-2^{-2b})n + o(n)$ queries.*

Proof. As described briefly in Section 5, the general approach for the algorithm that obtains the desired bounds is to count the number of variables included in the parity, modulo 2^b, in order to distinguish k-linear from $(k+2\ell)$-linear functions. The details of a tester that implements this approach is described in Algorithm **??**.

Let's first examine the correctness of the algorithm. To do so, we need to argue that the algorithm correctly counts the number of variables in the parity, modulo 2^b. This is easily verified by noting that every element $i \in S_r$ is the representative of a set of 2^r elements whose corresponding variables are either all included or all excluded from the parity.

To complete the analysis, we must also analyze the query complexity of the algorithm. In the worst-case, Algorithm **??** may query up to $\frac{n}{2^r}$ inputs in round r, for a total of $\frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots \approx n$ queries. But the expected number of queries is much smaller: since we pick our matching at random and f is nearly balanced, the expected number of queries that return 0 is $\frac{1}{2} \pm o(1)$. Therefore, the expected number of elements in S_r is $\frac{n}{2^{2r}} + o(n)$ and the expected number of queries is $\frac{n}{2} + \frac{n}{8} + \frac{n}{32} + \cdots + \frac{n}{2^{2(b-1)}} + o(n) = \frac{2}{3}(1 - 2^{-2b})n + o(n)$. Furthermore, with high probability the number of queries required by the algorithm is within $\pm o(n)$ of this expected value; to complete the proof of the theorem, simply run Algorithm **??** with a query quota so that if the quota is reached, we terminate the algorithm and guess. Setting the quota large enough, this termination occurs only with probability $o(1)$, and so we have a valid (k vs. $k + 2\ell$)-tester. □

Remark 1. When $\ell > \omega(\sqrt{n})$, the result in Theorem 1.4 is not optimal. In fact, in this case it is possible to solve the ($\frac{n}{2} - \ell$ vs. $\frac{n}{2} + \ell$)-parity testing problem with $O(n/\ell^2)$ queries with a simple sampling approach. (Sample $O(n/\ell^2)$ elements $i_1, \ldots, i_s$ uniformly at random from $[n]$, query $f(e_{i_1}), \ldots, f(e_{i_s})$, and guess that f is a $\frac{n}{2} - \ell$-parity function iff at most $\frac{1}{2}$ of the queries returned the value 1. By a Chernoff bound argument, with high probably this approach correctly solves the testing problem.)

Theorem E.1. *Let $n \geq k \geq 0$. Let $k' = n - k$. There is an adaptive k-linearity ϵ-tester that makes*

$$\sum_{i=0}^{\infty} 2^{-i-1}(k^{2^i} + k'^{2^i}) \prod_{j=0}^{i-1}(k^{2^j} + k'^{2^j})^{-1} + O(\sqrt{n} + 1/\epsilon)$$

queries.

Remark 2. Theorem 1.3 is the special case of Theorem **??** where $k = \frac{n}{2}$.

Proof. We first define an algorithm that assumes the input is a linear function. We discuss how to handle non-linear functions at the end of the proof. We define the algorithm recursively. It tests if the parity of a linear function f on $\{0,1\}^S$ (for some set S) is equal to k with failure probability at most p. We assume for sake of simplicity that $k \leq |S|/2$. Were this not the case, we could test for the parity of the pointwise sum of f with the parity function on all of S. The full algorithm is presented in Algorithm **??**.

We have left to verify that Algorithm **??** works in an appropriate runtime. Let

$$h(x, y) = \sum_{i=0}^{\infty} 2^{-i-1}(x^{2^i} + y^{2^i}) \prod_{j=0}^{i-1}(x^{2^j} + y^{2^j})^{-1}.$$

We first note that if f is a k-linear function on $\{0,1\}^S$ for $k \leq |S|/2$, then for a random $i \in S$, then the probability that $f(\{i\}) = 0$ will be at most $1/2$. Therefore, the probability that $\log_2(6/p)$ such i all have this property is at most $p/6$. Hence if $|S|$ is odd, the probability of failure is at most $p/6$ plus the probability of failure of the recursive call, which is at most $5p/6$.

We note that each of the pairs (i, j) for which $f(\{i, j\}) = 1$ have total weight of exactly one between them. The other pairs have $f(\{i\}) = f(\{j\})$. Therefore the weight of f on S is equal to k if and only if t plus twice the weight of f on T equals k. We note also that if f were weight k on S that the expected value of t would be $\frac{k(|S|-k)}{2|S|}$ with a variance of $O(|S|)$. Hence for C sufficiently large, we only report False in error due to t being too large or small with probability at most $p/6$. This verifies the correctness of the algorithm. We need to verify that it runs in at most $h(k, |S| - k) + O(\sqrt{|S|})$ queries.

We note that the algorithm makes $O(\log(|S|/p)) + |S|/2$ queries before making a recursive call on t,T. If $x = k$ and $y = |S| - k$, we note that we make recursive calls with new values of x and y given by either x and $y - 1$ or by $\frac{x^2}{2(x+y)} + c$ and $\frac{y^2}{2(x+y)} + c$ with $c = O(\sqrt{(x + y)/p})$. For the first case, we note that for $x \leq y$ that $h(x, y) \leq h(x, y - 1)$. Upon applying the latter recursion, we note that were c equal to 0 that

$$h(x, y) = (x + y)/2 + h\left(\frac{x^2}{2(x + y)}, \frac{y^2}{2(x + y)}\right).$$

We need to show that having a value of c not equal to 0 does not significantly effect the runtime of the recursive call to the algorithm. In particular we show that it changes the runtime by $O(c)$. We do this by

Algorithm 2 Linearity-Tester(k,S,p)

```
if $|S| = 0$ and $k = 0$ then
    Return True
end if
if $|S| = 0$ and $k \neq 0$ then
    Return False
end if
if $|S|$ is odd then
    Choose $\log_2(6/p)$ random $i \in S$
    for Each chosen $i$ do
        query $f(\{i\})$
    end for
    if All of these queries return 1 then
        Return False
    else
        For $i$ so that $f(\{i\}) = 0$, Return Linearity-Tester($k$,$S - \{i\}$,$5p/6$)
    end if
else
    Let $M$ be a random matching of the elements of $S$
    Let $T = \emptyset$, $t = 0$
    for $(i, j) \in M$ do
        if $f(\{i, j\}) = 1$ then
            Set $t = t + 1$
        else
            Set $T = T \cup \{i\}$
        end if
    end for
    if $t \not\equiv k \pmod 2$ then
        Return False
    else if $\left| t - \frac{k(|S|-k)}{2|S|} \right| > C\sqrt{|S|/p}$ for $C$ a sufficiently large constant then
        Return False
    else
        Return Linearity-Tester($(k - t)/2$,$T$,$5p/6$)
    end if
end if
```

showing that the directional derivative of $h(x, y)$ in the $(1, 1)$ direction is $O(1)$. In order to do this we note that $h(x, y) = \sum_{i=0}^{\infty} h_i(x, y)$ where

$$h_i(x, y) = 2^{-i-1}(x^{2^i} + y^{2^i}) \prod_{j=0}^{i-1} (x^{2^j} + y^{2^j})^{-1} = \frac{2^{-i-1}(x^{2^i} + y^{2^i})}{x^{2^i-1} + x^{2^i-2}y + \ldots + y^{2^i-1}}.$$

The derivative of h_i in the $(1, 1)$ direction is

$$\begin{aligned} & \frac{(x^{2^i-1} + y^{2^i-1})(x^{2^i-1} + x^{2^i-2}y + \ldots + y^{2^i-1}) - (x^{2^i} + y^{2^i})(x^{2^i-2} + x^{2^i-3}y + \ldots + y^{2^i-2})}{2(x^{2^i-1} + x^{2^i-2}y + \ldots + y^{2^i-1})^2} \\ = \quad & \frac{x^{2^i-1}y^{2^i-1}}{(x^{2^i-1} + x^{2^i-2}y + \ldots + y^{2^i-1})^2} \leq 2^{-2i}. \end{aligned}$$

Where the last step above is by AM-GM. Thus the directional derivative of $h(x, y)$ is $O(1)$. We prove inductively that for sufficiently large K that our algorithm runs in time at most $h(k, |S| - k) + K\sqrt{|S|/p}$.

From the above discussion it is clear that upon this inductive hypothesis and for sufficiently large C' that our algorithm runs in time at most $h(k, |S|-k)+C'\sqrt{|S|/p}+K\sqrt{(|S|/2)/(2p/3)}$. Hence for K a sufficiently large multiple of C', we can complete our inductive step.

We now complete the proof by extending the algorithm to reject all functions—and not just linear functions—that are far from all k-linear functions. First, we add an extra step where we run the Blum–Luby–Rubinfeld linearity tester $O(1/\epsilon)$ times. This rejects all functions that are ϵ-far from linear.

At this point, we are almost, but not quite, done. We can still have functions that are very close to linear (so that they pass the linearity test), very far from k-linear (so that the overall test should reject), and yet that pass the test in Algorithm 2 with high probability. For example, a function f might be consistent with a k-linear function on all inputs of Hamming weight at most 2 and consistent with some $(k+2)$-linear function on the remaining inputs. Since the algorithm only queries the function on inputs of low Hamming weight, it will erroneously accept f.

To remove this last source of error, we add one last step. In this step, we choose uniformly at random one of the queries $\{i, j\}$ that was made by Algorithm 2. We then choose $x \in \mathbb{F}_2^n$ uniformly at random, and set $y \in \mathbb{F}_2^n$ to be identical to x except for y_i and y_j, who take the opposite values of x_i and x_j, respectively. We then verify that $f(x) = f(y)$ iff $f(\{i, j\}) = 0$. Running this test a constant number of times is sufficient to verify that the function is globally consistent with the answers returned by the queries and, in particular, functions that are close to linear but not are not k-linear and still passed the test in Algorithm 2 will fail this test with high probability. □

We remark that the above testing algorithm tests k-linearity in time $cn + o(n)$ for some constant $c < 1$ except when $k = o(n)$. When k is small, however, there is another simple sampling algorithm that can be used to test k-linearity with $O(k \log(n/k))$ queries.

This algorithm allows us to test for k-linearity in time equal to n times some constant less than 1 unless k is $o(n)$. In this latter case, we have a different type of algorithm. In particular, we show that:

Theorem ??. (Restated) *Let $n \geq k \geq 0$. There is an adaptive k-linearity tester that makes $O(k \log(n/k))$ queries.*

The idea of the algorithm depends largely on the following algorithm. We claim that there is an algorithm that given a linear function f on $\{0,1\}^S$ with $|S| \geq 12k$ spends $O(k)$ time and may return a set $T \subset S$ with $|T| \leq |S|/2$. Furthermore if this algorithm returns a T, there is a probability at least $1-2^{-k}$ that f depends only on the coordinates in T. Also if f is a k-linear function, the algorithm will return a T with at least 50% probability.

This in turn will depend on an even simpler algorithm to test if a linear function f is 0 on $\{0,1\}^S$. This is done simply by evaluating f on a random vector. If f is 0 it will return 0, otherwise it will return 1 with 50% probability. Our range reduction algorithm is as follows:

It is clear that Algorithm **??** runs in time $O(k)$. It is also clear from the last round of queries at the end that the algorithm will return $S-V$ when f is non-trivial on $\{0,1\}^V$ with probability at most 2^{-k}. We have left to show that if f is actually a k-linear function that $|V| > |S|/2$ with probability at least 1/2. We do this by showing that the expected size of $S-V$ is at most $|S|/4$. The result will then follow from the Markov bound.

We first show the following claim. At the beginning on the r^{th} iteration of the while loop, the expected number of elements U of $\mathfrak{S}$, so that f is non-trivial on $\{0,1\}^U$ is at most $4^{-(r-1)}k$. This is clear for $r = 1$. At each round, each such element U will have a probability of 3/4 of being removed from $\mathfrak{S}$. Each of these sets is a union of at most 2^{r-1} of the sets in our original partition. Thus $S-V$ will be a union of n sets from our original partition, and that the expected size of n will be at most $\sum_{r=1}^{\infty} 2^{r-1}4^{-(r-1)}k = 2k$. Hence with probability at least 1/2, $S-V$ is a union of at most $4k$ of these sets. The sets thrown away are at most twice the size of the other sets, and hence in this case, $|S-V| \leq |S|/2$.

In order to get our final k-linearity testing algorithm, we do the following. If $k \leq \sqrt{n}$, we can use an $O(k\log(k)$-query algorithm. Otherwise, we iteratively run our Range-Reduce algorithm $3\log_2(n/k)$ times or until we are left with a set of size less than $12k$. If f were actually a k parity function, then the probability that this algorithm fails more than $2\log_2(n/k)$ times will be at most 10% (at least for n/k sufficiently large).

Algorithm 3 Range-Reduction(k,S)

```
Partition $S$ into $12k$ subsets of nearly equal size. Let $\mathfrak{S}$ be the set of these subsets.
while $|\mathfrak{S}| > 1$ do
  for $U \in \mathfrak{S}$ do
    Query $f$ on two random elements of $\{0,1\}^U$.
    if One of these queries returns 1 then
      Remove $U$ from $\mathfrak{S}$.
    end if
  end for
  Let $M$ be a pairing of all or all but one of the elements of $\mathfrak{S}$
  for $(U, U') \in M$ do
    Remove $U$ and $U'$ from $\mathfrak{S}$ and add $U \cup U'$
  end for
end while
if $\mathfrak{S} = \emptyset$ then
  Return Fail
else
  Let $V$ be the single element of $\mathfrak{S}$
end if
Query $f$ at $k$ random elements of $\{0,1\}^V$
if Any of these queries returns 1 of $V < |S|/2$ then
  Output Fail
else
  Return $S - V$
end if
```

Thus if it fails more than this many times, we declare that f is not k-linear. Otherwise, we are left with a T of size at most $12k$ so that f is 0 on $\{0,1\}^{S-T}$ with probability at least $1 - O(\log(n/k)2^{-k}) \geq 90\%$. Under the assumption that f is actually a parity function on some subset of T, we can then evaluate f on a basis of $\{0,1\}^T$ in $|T| = O(k)$ queries, and thus determine the parity of f. This takes a total of $O(k \log(n/k))$ queries and has a failure probability of at most 20%.