Electronic Colloquium on Computational Complexity, Report No. 199 (2010)

Symmetric LDPC codes are not necessarily locally testable

Eli Ben-Sasson* Ghid Maatouk† Amir Shpilka‡ Madhu Sudan§

December 14, 2010

Abstract

Locally testable codes, i.e., codes where membership in the code is testable with a constant number of queries, have played a central role in complexity theory. It is well known that a code must be a "low-density parity check" (LDPC) code for it to be locally testable, but few LDPC codes are known to the locally testable, and even fewer classes of LDPC codes are known *not* to be locally testable. Indeed, most previous examples of codes that are not locally testable were also not LDPC. The only exception was in the work of Ben-Sasson et al. [2005] who showed that random LDPC codes are not locally testable. Random codes lack "structure" and in particular "symmetries" motivating the possibility that "symmetric LDPC" codes are locally testable, a question raised in the work of Alon et al. [2005]. If true such a result would capture many of the basic ingredients of known locally testable codes.

In this work we rule out such a possibility by giving a highly symmetric ("2-transitive") family of LDPC codes that are not testable with a constant number of queries. We do so by continuing the exploration of "affine-invariant codes" — codes where the coordinates of the words are associated with a finite field, and the code is invariant under affine transformations of the field. New to our study is the use of fields that have many subfields, and showing that such a setting allows sufficient richness to provide new obstacles to local testability, even in the presence of structure and symmetry.

Keywords: Property testing, Invariance, Error-correcting codes

*Faculty of Computer Science, Technion — Israel Institute of Technology, Haifa, Israel, eli@cs.technion.ac.il. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 240258 and from the US-Israel Binational Science Foundation under grant number 2006104.

†School of Computer and Communications Sciences, EPFL, Switzerland, ghid.maatouk@epfl.ch. Part of this work was conducted while the author was an intern at Microsoft Research. It is supported in part by Grant 228021-ECCSciEng of the European Research Council.

‡Faculty of Computer Science, Technion — Israel Institute of Technology, Haifa, Israel and Microsoft Research, Cambridge, MA, shpilka@cs.technion.ac.il. This research was partially supported by the Israel Science Foundation (grant number 339/10).

§Microsoft Research New England, Cambridge, Massachusetts, USA, madhu@mit.edu.

ISSN 1433-8092

1 Introduction

An error-correcting code $C \subseteq \mathbb{F}_2^N$ is said to be locally testable if membership in the code is verifiable with a constant number of queries (independent of N). (Many of the terms introduced in this section will be defined more formally in later sections.) The existence of locally testable codes (LTCs), especially with good rate and distance, is a surprising phenomenon. Their discovery (and utility so far) is closely tied to the discovery of probabilistically checkable proofs, and the performance of the two have also been closely related. (The first systematic study of such codes occurred in [Goldreich and Sudan, 2006], though such codes were defined informally as far back as in [Babai et al., 1991], and significantly investigated in recent years.)

While many, simple and complex, constructions of locally testable codes are known, less seems to be known about what makes codes locally testable. There are few general "necessary" or "sufficient" conditions known. It is known that very sparse linear codes [Kaufman and Litsyn, 2005, Kaufman and Sudan, 2007] and a certain subfamily of "symmetric" codes [Kaufman and Sudan, 2008, Kaufman and Lovett, 2010] are locally testable. At the same time, there are few "counterexamples" to several tempting conjectures about what makes a code testable. Our result in this work considers two very natural "conditions" for sufficiency, and rules out their *conjunction* as being a sufficient condition for local testability. To describe these two conditions we need to explain the basic context.

Linear Codes and Duals: Throughout this work, we are interested in binary *linear* codes. Specifically, letting $\mathbb{F}_2$ denote the field of two elements, a code $C \subseteq \mathbb{F}_2^N$ is said to be linear if it is a subspace of $\mathbb{F}_2^N$. A natural method to test linear codes is via their "duals". For $x, y \in \mathbb{F}_2^N$, let x_i, y_i denote the ith coordinates of x and y respectively, and let $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$ denote the standard inner product of x and y. Then, the *dual* of C, denoted $C^\perp$, is the collection of vectors $\{y \in \mathbb{F}_2^N | \langle x, y \rangle = 0\}$. (Note that $C^\perp$ is also a linear code, with $(C^\perp)^\perp = C$.)

Every $y \in C^\perp$ denotes a potential "test" for membership in C and a vector w passes the y-test if $\langle w, y \rangle = 0$. Since such a test only needs to query w on the support of y (the set $\{i \in \{1, \ldots, N\} | y_i \neq 0\}$), this can be a "local" test if y has small support. Furthermore, as shown by Ben-Sasson et al. [2005], a tester for membership in C can be converted into one that picks $y \in C^\perp$ according to some distribution and applies the y-test (without qualitative change in parameters, and in particular, without any increase in the query complexity).

Low-Density Parity Check (LDPC) Codes: The connection between tests for membership in C and the dual of C, leads to a natural necessary condition for a code C to have a k local tester (a tester making at most k queries). Specifically, $C^\perp$ must have low "weight" codewords, i.e., codewords of support size at most k. Furthermore, if the test rejects every non-codeword with positive probability, then it must be the case that $C^\perp$ is *spanned* linearly by codewords of weight at most k. Codes that have the property that their duals are spanned by their low-weight codewords are the very popular and well-studied family of LDPC codes.[1]

A natural question about LDPC codes is whether every LDPC code is also locally testable. This question was answered negatively by Ben-Sasson, Harsha, and Raskhodnikova [2005] who showed that random LDPC codes (with appropriate parameters) are not locally testable.

[1] The usual description of LDPC codes is via a low density "parity check" matrix H such that $xH = 0$ for every $x \in C$. To relate this to our description one should take the columns of H to be a maximal-size set of linearly independent elements of $C^\perp$ of weight at most k.

Codes with Symmetries: One informal interpretation of the negative result of [Ben-Sasson et al., 2005] is that random codes offer very little "structure" and testability ought to be hard in the absence of "structure".

One way to formalize such a statement could be via the symmetries of the code. Given a code $C \subseteq \mathbb{F}_2^N$ and a permutation $\pi : \{1, \ldots, N\} \to \{1, \ldots, N\}$, let $C \circ \pi$ denote the set $\{x \circ \pi = \langle x_{\pi(1)}, \ldots, x_{\pi(N)} \rangle | x \in C\}$. Given a code $C \subseteq \mathbb{F}_2^N$, let $G(C)$ be its automorphism group, i.e., $G(C)$ is the set of permutations $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that C is invariant under π, i.e., $C = C \circ \pi$. For a random code $C \subseteq \mathbb{F}_2^n$ (of appropriately large size) the automorphism group is likely to be trivial, i.e., contain only the identity permutation. Perhaps codes with non-trivial automorphism groups are testable?

This question was raised explicitly by Alon et al. [2005] who asked if every code C with a "2-transitive" automorphism group, *and* a low-weight codeword in the dual (the above-mentioned necessary condition), is locally-testable. ($G(C)$ acting on $\{1, \ldots, N\}$ is t-transitive if for every two sequences $i_1, \ldots, i_t$ and $j_1, \ldots, j_t$ of distinct elements of $\{1, \ldots, N\}$, there is a permutation $\pi \in G(C)$ with $\pi(i_\ell) = j_\ell$ for every $\ell \in \{1, \ldots, t\}$.) 2-transitivity implies a high level of redundancy in the low-weight codewords of $C^\perp$. Indeed the presence of even a single word of weight at most k in $C^\perp$ implies the presence of $\Omega(N^2)$ such words (by the transitivity, since every permutation preserves the weight of words) and gives a code that is locally decodable and locally correctable. Such a highly redundant set of low-weight words in the dual seems to suggest that they should span the dual, and furthermore suffice for testing C (under some appropriately chosen distribution). Unfortunately, even the first of these hopes turn out to be not true, as shown by Grigorescu et al. [2008]. They give an example of a family of 2-transitive codes which have a low-weight codeword in the dual, but are not spanned by low-weight codewords (and so are not LDPC codes).

Our Results: This work is motivated by a less ambitious interpretation of the question raised by Alon et al.: Specifically we consider the question as to whether symmetries in an LDPC code lead to local testability. Boosting the hope for such a question is the fact that any 2-transitive LDPC code is a locally decodable (even self-correctible) code. Thus to conjecture testability is tempting. However, we give a negative answer to this question. Specifically we show the following theorem:

Theorem 1.1. *There exists an infinite family of 2-transitive LDPC codes which is not testable with a constant number of queries. Specifically, there exists a constant c and an infinite family of LDPC codes $\{C_N \subseteq \mathbb{F}_2^N\}$ where each $C_N^\perp$ is spanned by its codewords of weight at most c, such that for every k, for sufficiently large N, C_N is not k-locally testable.*

Prior to our work, the only result showing some LDPC codes are not locally testable, were those of [Ben-Sasson et al., 2005], who show random (and thus far from symmetric) codes are not locally testable. Thus, our results give the first "structured" family of LDPC codes that are not locally testable.

We also note that in the broader context of symmetries in property testing (beyond linear codes), our example gives a more symmetric property than previously known ones that is not testable. The only previous work, due to Goldreich and Kaufman [2010], gives a 1-transitive (non-linear) property that is not locally testable.

Affine-Invariance: We derive our counterexample codes by looking into the class of "affine-invariant" codes. This is a well-studied family of codes in the literature on coding theory. Its

study in the context of local testability was initiated by Kaufman and Sudan [2008] and the codes considered in the above mentioned work of Grigorescu et al. [2008] come from this family of codes.

The coordinates of a binary affine-invariant code $C \subseteq \mathbb{F}_2^N$ are associated with a finite field $\mathbb{F}_{2^n}$ (so $N = 2^n$), and the code is invariant under every permutation $\pi_{\alpha,\beta} : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$, $\pi_{\alpha,\beta}(x) = \alpha x + \beta$, where $\alpha \in \mathbb{F}_{2^n} \setminus \{0\}$ and $\beta \in \mathbb{F}_{2^n}$. (Note that in the work of Kaufman and Sudan [2008] they also consider seemingly broader classes, where the coordinates form a vector space over some field, but every such code will also be affine-invariant in our sense.)

Affine-invariant codes give explicit families of highly symmetric codes, and provide enough variety to both give "broad" positive results, and counterexamples to even "broader" conjectures. For example, Kaufman and Sudan consider codes with the "single-orbit" property, namely that the dual is spanned by the "orbit" of a single low-weight codeword. (The "orbit" of a word $w \in \mathbb{F}_2^N$ is simply all words $w \circ \pi$, where π is an affine permutation.) They show that *every* code with the single orbit property is locally testable, by the natural test.

Indeed their work motivated the following hope: that every affine-invariant LDPC code may have the single-orbit property and thus be testable. (Of course, the family from Theorem 2.6, does not have such a single-orbit property.) Previous works attempting to rule out testability for affine-invariant codes have only been able to set forth some conditions under which the codes do not have low-weight codewords in the dual at all [Ben-Sasson and Sudan, 2010], or these do not span the dual [Grigorescu et al., 2008]. Thus to analyze codes whose duals are spanned by their low-weight words, but do not have the single-orbit property, leads to new technical challenges, and, we hope, to new insights about the class of affine-invariant codes.

High-level view of our codes: The main idea behind our construction itself uses some of the rich theory already developed for affine-invariant codes. Our basic idea is to construct several affine-invariant LDPC codes $C_1, \ldots, C_\ell \subseteq \mathbb{F}_2^N$, where $N = 2^n$ and consider $C = \cap_{i \in \{1,\ldots,\ell\}} C_i$. Since the intersection preserves both affine-invariance and the LDPC property, C ends up also being an affine-invariant LDPC code. The key is to select C_i's so that the intersection is not locally testable, or at least does not have the single-orbit feature, even though C_i's individually do. (Note that all previously known affine-invariant LDPC codes did satisfy the single-orbit feature.) Below we attempt to describe how we select the C_i's.

Our first task is to ensure that the C_i's are LDPC codes. Unfortunately the "explicit" ones we know (based on low-degree multivariate polynomials) turn out not easy to analyze. So we turn to a less explicit, but more general result due to Grigorescu et al. [2009] (see also [Kaufman and Lovett, 2010]) which claims that any "sparse" affine-invariant code is a single-orbit code (hence an LDPC code). "Sparseness" simply fixes the number of codewords to be some polynomial in N. Unfortunately taking C_i to be a sparse code is futile since C, being a subset of C_i will be even more sparse and the above mentioned results would imply that C also has the single-orbit feature. This is where we turn to a special class of integers n. We only consider n that are a product of small primes. We pick $p_1, \ldots, p_\ell$ to be distinct primes with $p_i = \Theta(\log n)$ and $\ell = O(\log n / \log\log n)$ so that $n = p_1 \cdot p_2 \cdots p_\ell$. We take $\tilde{C}_i$ to be some sparse code contained in $\mathbb{F}_2^{2^{p_i}}$ where we view the coordinates of this code to be $\mathbb{F}_{2^{p_i}} \subseteq \mathbb{F}_{2^n}$. Since $\tilde{C}_i$ is sparse, it is a single-orbit code over the smaller domain. We then propose a certain "lifting" of this code to the code C_i over the domain $\mathbb{F}_{2^n}$ which preserves the single-orbit property while making the code non-sparse. The resulting code $C = \cap_i C_i$ now at least has some hope of being non-sparse and even of not being testable.

Proving that a code is not testable is non-trivial. We do so by considering the "canonical tester" proposed by Ben-Sasson et al. [2005] which tests a vector w by picking a low-weight codeword $y \in \mathcal{C}^\perp$ and accepts w if and only if $\langle y, w \rangle = 0$. Our main technical result involves showing that for a careful choice of $\tilde{C}_i$'s a codeword of weight k tends to accept some words far from C with probability at least $1 - k/\ell$ (and so the testing complexity is $\Omega(\ell) = \Omega(\log \log N / \log \log \log N)$. To explain the actual choice of the $\tilde{C}_i$'s we need to explain a fair bit of the (known) structural results about affine-invariant codes, so we won't do it now. But it suffices to say that the choice, understanding the "lifts" of these codes, and finally proving the limitation of local tests on C are the main contributions of this work.

Organization: In Section 2 we present some basic definitions and a formal statement of our main result (Theorem 2.6 which is a more formal version of Theorem 1.1). In Section 3 we present some standard background related to affine-invariant codes. In Section 4 we present our new class of codes and prove our main theorem modulo some technical results. We start with a simple proof that our codes are indeed LDPC codes, in Section 5. In Section 6 we show that our codes do not possess a sufficient condition for local testability. The machinery allows us to prove in Section 7 that our codes are not locally testable.

2 Formal definitions and statement of results

Notation and basic coding theory definitions: Let $[n]$ denote the set $\{1, \ldots, n\}$. The letters $\mathbb{F}, \mathbb{K}, \mathbb{L}$ will invariably denote finite fields and $\mathbb{F}_q$ is the field of size q. For functions $f : X \to Y$ and $g : Y \to Z$ (for arbitrary sets X, Y, Z), we let $g \circ f : X \to Z$ denote their composition, i.e., $(g \circ f)(x) = g(f(x))$. We use the standard notation for error correcting codes as defined in, e.g., MacWilliams and Sloane [1978]. For finite alphabet Σ and integers n, k, d, a $(n, k, d)_\Sigma$ code $\mathcal{C}$ is a subset of Σ^n of size $|\mathcal{C}| \geq |\Sigma|^k$ such that the minimal Hamming distance between distinct codewords $w, u \in \mathcal{C}$ is at least d. For the special case that Σ is a finite field $\mathbb{F}$ (as will hold throughout this paper), an $[n, k, d]_\mathbb{F}$-code $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}^n$ such that every nonzero codeword $f \in \mathcal{C} \setminus \{0\}$ has weight at least d. We shall view a codeword $f \in \mathcal{C}$ as a function $f : [n] \to \Sigma$ and define its support to be the set $\mathrm{supp}(f) = \{i \in [n] \mid f(i) \neq 0\}$ and its weight to be the size of its support. The dual code of $\mathcal{C}$ is $\mathcal{C}^\perp = \{u \in \mathbb{F}^n \mid u \perp \mathcal{C}\}$ where $u \perp V$ if and only if $\langle u, v \rangle = 0$ for all $v \in V$. Let $\mathcal{C}^\perp_{\leq q}$ denote the set of dual words that have weight at most q. If $(\mathcal{C}^\perp_{\leq q})^\perp = \mathcal{C}$ we say that $\mathcal{C}$ is a *q-low-density-parity-check code* (q-LDPC code, for brevity) because it is characterized by a parity check matrix in which each row has small weight (or low density).

Definition 2.1 (Locally testable code (LTC))**.** *For integer q and constants $\delta, s, c \in [0, 1]$, a* (q, δ, s, c)-tester *for a $(n, k, d)_\Sigma$ code $\mathcal{C}$ is a randomized Turing machine T with oracle access to a purported codeword $f : [n] \to \Sigma$ that satisfies:*

Operation *T makes at most q queries to f and outputs either accept or reject. Let T^f denote the output of T on oracle f and notice T^f is a random variable because T is a random machine.*

Completeness *If $f \in \mathcal{C}$ then* $\Pr[T^f = \texttt{accept}] \geq c$.

Soundness *If the Hamming distance of f from $\mathcal{C}$ is at least δn (in which case we say f is δ-far from $\mathcal{C}$) then* $\Pr[T^f = \texttt{reject}] \geq s$.

The probability stated above is with respect to the randomness used by T. The code $\mathcal{C}$ is said to be (q, δ, s, c)-locally testable if there exists a (q, δ, s, c)-tester for it. q is the query complexity*, δ the* distance *(or proximity) parameter, s is the* soundness *and c is the* completeness *of the tester and its associated code. When $c = 1$ (as will be the case in this paper) we say the tester and associated code have* perfect completeness *and omit the completeness parameter altogether.*

Remark 2.2 (On soundness and completeness)**.** *Taking soundness and completeness parameters $s + c \leq 1$ leads to trivial results — all codes are testable with such parameters by a tester that accepts all words with probability c hence rejects all words with probability $1 - c \geq s$. So we shall always assume $s + c > 1$.*

The following class of canonical testers is particularly useful for proving lower bounds on linear LTCs, because the analysis of a canonical tester can be carried out using tools from linear algebra. In what follows, for $f \in \mathbb{F}^n$ and $I \subset [n]$ let $f|_I$ be the projection of f to the set of coordinates I and for $\mathcal{C} \subset \mathbb{F}^n$ let $\mathcal{C}|_I = \{f|_I \mid f \in \mathcal{C}\}$.

Definition 2.3 (Linear and canonical testers for a linear code)**.** *A* linear q-tester *for a $[n, k, d]_{\mathbb{F}}$-linear code $\mathcal{C}$ is specified by a distribution μ over subsets $\{I \subset [n] \mid |I| \leq q\}$. Such a tester operates by sampling I according to μ and accepting if and only if $f|_I \in \mathcal{C}|_I$.*

A canonical q-tester *for $\mathcal{C}$ is specified by a distribution μ over $\mathcal{C}^{\perp}_{\leq q}$. It operates by sampling u according to μ and accepting f if and only if $\langle u, f \rangle = 0$, where $\langle u, f \rangle = \sum_{i=1}^{n} u(i)f(i)$.*

Ben-Sasson et al. [2005] showed that a tester for a linear code can be assumed to be a linear tester without any loss in parameters. And going from a linear tester to a canonical one results in a loss in soundness by a factor that depends only on the field size. (See, e.g., [Ben-Sasson et al., 2009, Section 2] for a discussion of the linear-to-canonical transition.) We summarize this by the following claim.

Claim 2.4. *If $\mathcal{C}$ is an $[n, k, d]_{\mathbb{F}}$-code that has a (q, δ, s, c)-tester then $\mathcal{C}$ has a $(q, \delta, (s + c - 1))(1 - \frac{1}{|\mathbb{F}|}), 1)$-canonical tester.*

2.1 Affine invariant low density parity check (LDPC) codes

We now turn to define affine-invariant codes, focusing on LDPC affine-invariant codes. Before getting to the definitions we make a shift in our coding-related notation to be in line with the notation used for describing such codes. In particular, we shall associate the set of coordinates $[n]$ with the elements of a finite field $\mathbb{K}$ (with $|\mathbb{K}| = n$) and view words in $\mathbb{F}^n$ as functions mapping $\mathbb{K}$ to $\mathbb{F}$. Letting $\{\mathbb{K} \to \mathbb{F}\}$ denote the set of functions from $\mathbb{K}$ to $\mathbb{F}$, a code $\mathcal{C}$ is now viewed as a linear subspace of $\{\mathbb{K} \to \mathbb{F}\}$. We define $\text{Affine}_{\mathbb{K}}$ to denote the set of affine transformations $T : \mathbb{K} \to \mathbb{K}$.

Definition 2.5 (Affine-invariant codes)**.** *A linear code $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ is said to be* affine-invariant *if $\mathbb{K}$ extends $\mathbb{F}$ (so $n = |\mathbb{K}| = |\mathbb{F}|^t$ for some integer t) and $\mathcal{C}$ is "invariant" under the action of the affine semi-group. Specifically, for every $f \in \mathcal{C}$ and every affine-transformation $T \in \text{Affine}_{\mathbb{K}}$ we have $(f \circ T) \in \mathcal{C}$ where $(f \circ T)(\alpha) = f(T(\alpha))$ for all $\alpha \in \mathbb{K}$.*

We are ready to state our main result whose proof appears in Section 4.

Theorem 2.6 (Affine-invariant LDPC codes are not necessarily locally testable)**.** *For every prime p there exist constants $\delta, \gamma > 0$, a positive integer k and an infinite family of positive integers $\mathcal{N}$ such that for every $n \in \mathcal{N}$ the following holds:*

Code *There is an affine-invariant code $\mathcal{C}^{(n)} \subseteq \{\mathbb{F}_{p^n} \to \mathbb{F}_p\}$. I.e., $\mathcal{C}^{(n)}$ is a code of block length $N = p^n$ over $\mathbb{F}_p$.*

LDPC *$\mathcal{C}^{(n)}$ is a k-LDPC code.*

Non-testable *$\mathcal{C}^{(n)}$ is not $o(\log n/\log\log n)$-locally testable. Specifically, for every $s, c \in (0,1]$ satisfying $s+c>1$, every $(q(n), \delta, s, c)$-tester for $\mathcal{C}^{(n)}$ satisfies $q(n) \geq \gamma(s+c-1))\log n/\log\log n$.*

Remark 2.7. *Note that as a function of the block length $N = p^n$, the locality lower bound is $\Omega(\log\log N/\log\log\log N)$.*

3 Basic Background

To describe our codes, we need to reintroduce some basic notions used in previous works on testing affine-invariant properties, specifically, the notion of a "single-orbit characterization", the "degree-set" of an affine-invariant family, and the "sparsity" of a family. Readers familiar with the works of [Kaufman and Sudan, 2008, 2007, Grigorescu et al., 2008, 2009, Ben-Sasson and Sudan, 2010, Kaufman and Lovett, 2010] can skip most of this section after familiarizing themselves with our notation. We restrict our attention to functions mapping $\mathbb{K} = \mathbb{F}_{p^n}$ to $\mathbb{F} = \mathbb{F}_p$.

3.1 Single-orbit Characterization

A basic concept in testing is the notion of a *constraint*, namely a sequence of points $\alpha_1, \ldots, \alpha_k \in \mathbb{K}$ and a subset $S \subset \mathbb{F}^k$. A function f satisfies the constraint if $\langle f(\alpha_1), \ldots, f(\alpha_k)\rangle \in S$. We refer to k as the *locality* of the constraint. In this work we need to work only with *basic constraints* where the set S is given by a single linear constraint on the values (i.e., S is a co-dimension one subspace of $\mathbb{F}^k$). Thus a basic constraint is given by a pair $(\alpha, \lambda) \in \mathbb{K}^k \times \mathbb{F}^k$, where f satisfies the constraint if $\sum_{i=1}^k \lambda_i f(\alpha_i) = 0$. We suppress the term basic (all constraints considered in this work are basic) and simply refer to (α, λ) as a k-constraint. Note that such a constraint is equivalent to a dual codeword of weight k.

A code $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ satisfies a constraint if every $f \in \mathcal{C}$ satisfies it. A collection of constraints $\left\{(\alpha^{(j)}, \lambda^{(j)})\right\}_{j\in[m]}$ *characterizes* a code $\mathcal{C}$ if $\mathcal{C}$ is exactly the set of functions that satisfy every constraint $(\alpha^{(j)}, \lambda^{(j)})$, $j \in [m]$. Note that being a k-LDPC code is equivalent to being characterized by a collection of k-constraints.

The notion of interest to us is a single-orbit characterization, which arises when the characterization is simply the permutations of a single constraint under affine transformations of the domain, as formalized below.

Definition 3.1 (Single-orbit characterization)**.** *A k-constraint (α, λ) is said to be a* k-single orbit characterization*, or simply a* k-s-o-c*, of an affine-invariant code $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ if the following holds: $f \in \mathcal{C}$ if and only if f satisfies the constraints $(T \circ \alpha, \lambda)$ for every $T \in \mathrm{Affine}_{\mathbb{K}}$, where $T \circ \alpha = \langle T(\alpha_1), \ldots, T(\alpha_k)\rangle$. If $\mathcal{C}$ has a k-s-o-c we say $\mathcal{C}$ is* k-single-orbit*.*

The following result of Kaufman and Sudan [2008] shows that all affine-invariant codes that have a k-s-o-c are necessarily k-locally testable.

Theorem 3.2 (k-single-orbit codes are locally testable, [Kaufman and Sudan, 2008, Theorem 2.10])**.** *There exists $s > 0$ such that for every positive integer k, prime p, field $\mathbb{K}$ extending $\mathbb{F}_p$, and every $\delta > 0$ the following holds: If $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}_p\}$ is a k-single-orbit affine-invariant code then $\mathcal{C}$ is a $(k, \delta, s\delta/k^2, 1)$-locally testable code.*

All previously known affine-invariant k-locally testable codes were actually k-single-orbit. Since all previously known affine-invariant LDPCs were also locally testable, it follows that all previously studied LDPCs were actually single-orbit codes. Not surprisingly, single-orbit codes also form the starting points of our construction. Our goal is to come up with codes and transformations which preserve affine-invariance and the LDPC feature, while losing the single-orbit and local testability properties.

3.2 Degree sets of affine-invariant codes

To pick our basic (single-orbit) affine-invariant codes, we use a certain representation in terms of the "degrees" of the polynomials that represent codewords — a property studied previously in Kaufman and Sudan [2008] and subsequent works. We review the main notions below.

For a function $f : \mathbb{K} \to \mathbb{F}_p$ we let $f(X) \in \mathbb{K}[X]$ denote the unique univariate polynomial over $\mathbb{K}$ of degree at most $|\mathbb{K}| - 1$ that computes f. In what follows let $\mathrm{supp}_{\deg}\left(\sum_{i\geq 0} f_i X^i\right) = \{d \mid f_d \neq 0\}$ denote the set of powers of $f(X)$ with nonzero coefficients. We use the Trace map from $\mathbb{K}$ to $\mathbb{F}_p$ defined by $\mathrm{Trace}_{\mathbb{K}\to\mathbb{F}_p}(X) = \sum_{i=0}^{[\mathbb{K}:\mathbb{F}_p]-1} X^{p^i}$. (In the future we shall omit $\mathbb{K}$ and $\mathbb{F}_p$ when they are unambiguously defined.)

Definition 3.3 (Degree sets)**.** *Given a code $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ we let $\mathrm{Deg}(\mathcal{C}) = \cup_{f\in\mathcal{C}} \mathrm{supp}_{\deg}(f)$ be its* degree set*. Conversely, given a set $D \subseteq \{0, \ldots, |\mathbb{K}| - 1\}$, let $\mathrm{Code}(D) = \{\mathrm{Trace} \circ f | f \in \mathbb{K}[X], \mathrm{supp}_{\deg}(f) \subseteq D\}$, denote the code of D.*

For general codes, the notions above do not carry much relevance, however for affine-invariant codes they do. To summarize the effect we need to study the members of degree sets in their base-p representation. The following definitions become important to us.

Definition 3.4 (Shadow- and orbit- closed degree sets)**.** *For prime p and integer d let $[d]_p$ denote the base-p representation of d, i.e., $[d]_p = \langle d_0, d_1, \ldots\rangle$ such that $d = \sum_{i\geq 0} d_i p^i$. The* p-weight *of d is $\mathrm{wt}_p(d) = \sum_{i\geq 0} d_i$. Let the* p-shadow *of d be the set of integers whose base-p representation is, coordinate-wise, not greater than the base-p representation of d,*

$$\mathrm{shadow}_p(d) = \left\{ \sum_{i\geq 0} e_i p^i \;\middle|\; \forall i, e_i \leq d_i \right\}.$$

We write $e \leq_p d$ to denote $e \in \mathrm{shadow}_p(d)$ and $e <_p d$ denotes $e \in \mathrm{shadow}_p(d) \setminus \{d\}$.

A set of integers D is said to be p-shadow closed *if for all $d \in D$ we have $\mathrm{shadow}_p(d) \subseteq D$. When $\mathbb{K}$ is a degree-n extension of $\mathbb{F}_p$ we let $\mathrm{orbit}_{\mathbb{K}}(D) = \left\{d \cdot p^i \bmod p^n - 1 \;\middle|\; d \in D, i \in \{0, \ldots, n-1\}\right\}$ denote the* orbit *of D in $\mathbb{K}$. A set D is said to be* $\mathbb{K}$-orbit closed *if $\mathrm{orbit}_{\mathbb{K}}(D) = D$.*

The connection between shadow- and orbit-closed degree sets and affine-invariant codes is given by the following result which is implicit in many different works. For completeness we give its proof in Section 8.

Lemma 3.5 (Closed degree sets specify affine-invariant codes)**.** *Let $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ be affine-invariant. Then $\mathrm{Deg}(\mathcal{C})$ is shadow-closed and orbit-closed and $\mathcal{C} = \mathrm{Code}(\mathrm{Deg}(\mathcal{C}))$. Conversely, for every shadow-closed and orbit-closed family D, the code $\mathrm{Code}(D)$ is affine-invariant and satisfies $D = \mathrm{Deg}(\mathrm{Code}(D))$.*

Our codes will be constructed by choosing the degree set carefully, and then analyzing algebraic conditions that explain when they have single-orbit characterizations. The following lemma, proved in Section 9, expresses the notions of being a constraint and a single-orbit characterization, in terms of degree sets.

Lemma 3.6 (Degrees vs. Constraints and Characterizations)**.** *Let $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ be an affine-invariant code with degree set $D = \mathrm{Deg}(\mathcal{C})$. Let $(\alpha, \lambda) \in \mathbb{K}^k \times \mathbb{F}^k$ be a k-constraint. We have the following.*

1. *$\mathcal{C}$ satisfies (α, λ) if and only if $\sum_{i=1}^{k} \lambda_i \alpha_i^d = 0$ for every $d \in D$.*
2. *(α, λ) is a k-s-o-c of $\mathcal{C}$ if and only if we have: $d \in D(\mathcal{C}) \Leftrightarrow \forall e \leq_p d$, $\sum_{i=1}^{k} \lambda_i \alpha_i^e = 0$.*

3.3 Sparsity

Finally, we introduce the notion of the "sparsity" of an affine-invariant code. For our purposes, it is best to define it in terms of its degree set.

For a field $\mathbb{L}$ extending $\mathbb{F}$, we say that an affine-invariant code $\mathcal{C} \subseteq \{\mathbb{L} \to \mathbb{F}\}$ has *sparsity* s if there exists a set[2] S with $|S| \leq s$ such that $\mathcal{C} = \mathrm{Code}(S)$. It is easy to show that an s-sparse code has at most $|\mathbb{L}|^s$ codewords. Such codes are interesting in that they are single-orbit characterized, as shown in [Grigorescu et al., 2009, Theorem 4] for the case when $\mathbb{F} = \mathbb{F}_2$ and $\mathbb{L}$ is a prime-degree extension of $\mathbb{F}$, and by [Kaufman and Lovett, 2010, Theorem 1.7] for the general case.

Theorem 3.7 (Sparse affine-invariant codes have good distance and are single-orbit)**.** *For every prime p and constant s there exists a constant $\delta > 0$ and integer k such that the following holds. Let $\mathbb{L}$ be the degree m extension of $\mathbb{F}_p$. Let $\mathcal{C} \subseteq \{\mathbb{L} \to \mathbb{F}_p\}$ be an affine-invariant code of sparsity at most s. Then*

1. *$\mathcal{C}$ is a code of relative distance at least δ.*
2. *$\mathcal{C}$ is characterized by a single constraint of size at most k.*

[2]Note that one does not have to use sets that are orbit-closed for this definition. Indeed $\mathrm{Code}(S)$ equals $\mathrm{Code}(\mathrm{orbit}_{\mathbb{L}}(S))$ due to the nature of the Trace function but using S directly may lead to much smaller sparsity.

4 The Construction

In this section we describe the construction of our affine-invariant LDPC codes which are not locally testable. At the end of the section we state the main technical result regarding these codes (Theorem 4.7) and conclude Theorem 2.6 which is the main result of this paper. The proof of Theorem 4.7 is deferred to later sections.

Recall from Section 1 that we consider $n = p_1 \cdot p_2 \cdots p_\ell$ where the p_i's are distinct primes with each $p_i \approx \log n$ and $\ell = \Theta(\log n / \log\log n)$. This yields a field $\mathbb{K}$ with many subfields in it. The plan is to construct several codes $\tilde{\mathcal{C}}_i$, one corresponding to each $i \in [\ell]$ and then "lift" them into codes $\mathcal{C}_i$ and letting $\mathcal{C} = \cap_i \mathcal{C}_i$. We now go into the details of $\tilde{\mathcal{C}}_i$ and the "lifting".

Let $\mathbb{L}_i = \mathbb{F}_{p^{p_i}}$. We pick $\tilde{\mathcal{C}}_i$ to be an s-sparse, affine-invariant code mapping $\mathbb{L}_i$ to $\mathbb{F}$. By Theorem 3.7 such a code is k-single-orbit, so we cannot hope it is not testable. We then define a lifting operation which lifts this code to a code $\mathcal{C}_i \subseteq \{\mathbb{K} \to \mathbb{F}\}$. The lifting loses sparsity (as it must for our final result) but does not lose the k-single-orbit property (which also it needs to lose). (We only wish to preserve the LDPC property; the preservation of the single-orbit property is collateral damage.) But the single-orbit property is not necessarily preserved when we take the intersections of the $\mathcal{C}_i$, and indeed forms the basis of our hope that $\cap_i \mathcal{C}_i$ is not single-orbit or locally-testable. In later sections we prove that these features are not preserved confirming our hope, but for now we define the "lifting" operation and then describe our codes.

Formally, we define lifting in terms of what it does to the degree sets of affine-invariant codes. But to get some intuition, the idea, given a sequence of nested fields $\mathbb{F} \subset \mathbb{L} \subset \mathbb{K}$, is to take a constraint (α, λ) and then to just view it as a constraint on codes mapping $\mathbb{K}$ to $\mathbb{F}$. If (α, λ) is a k-s-o-c of some code $\tilde{\mathcal{C}} \subseteq \{\mathbb{L} \to \mathbb{F}\}$ when viewed as a constraint on codes mapping $\mathbb{L}$ to $\mathbb{F}$, and a k-s-o-c of $\mathcal{C}$ when viewed as a constraint on codes mapping $\mathbb{K}$ to $\mathbb{F}$, then we define $\mathcal{C}$ to be the "lift" of $\tilde{\mathcal{C}}$.

Definition 4.1 (Lifted code)**.** *Let $\mathbb{K} \supsetneq \mathbb{L} \supsetneq \mathbb{F}_p$ be finite fields. For $D \subseteq \{0, \ldots, |\mathbb{L}| - 1\}$ we define the* lift *of D from $\mathbb{L}$ to $\mathbb{K}$ to be the set of integers*

$$\mathrm{lift}_{\mathbb{L} \nearrow \mathbb{K}}(D) = \left\{ d' \in \{0, \ldots, |\mathbb{K}| - 1\} \;\middle|\; \left(\mathrm{shadow}_p(d') \bmod (|\mathbb{L}| - 1)\right) \subseteq D \right\}.$$

For an affine-invariant code $\mathcal{C} \subseteq \{\mathbb{L} \to \mathbb{F}_p\}$ with degree set D, let $\mathrm{lift}_{\mathbb{L} \nearrow \mathbb{K}}(\mathcal{C})$ be the affine-invariant code specified by degree set $\mathrm{lift}_{\mathbb{L} \nearrow \mathbb{K}}(D)$,

$$\mathrm{lift}_{\mathbb{L} \nearrow \mathbb{K}}(\mathcal{C}) = \{f : \mathbb{K} \to \mathbb{F}_p \mid \mathrm{supp}_{\deg}(f) \subseteq \mathrm{lift}_{\mathbb{L} \nearrow \mathbb{K}}(D)\} = \mathrm{Code}(\mathrm{lift}_{\mathbb{L} \nearrow \mathbb{K}}(D).$$

Remark 4.2 (Lifted affine-invariant code is affine-invariant)**.** *If $\mathcal{C}$ is an affine-invariant code, characterized by degree set D, notice that lifting $\mathcal{C}$ to a larger field results in an affine-invariant code. This is because the set $\mathrm{lift}_{\mathbb{L} \nearrow \mathbb{K}}(D)$ is, by definition, p-shadow-closed (assuming D is p-shadow-closed) and inspection reveals it is orbit closed because $|\mathbb{L}|$ divides $|\mathbb{K}|$.*

Next we define our family of non-LTCs.

Definition 4.3 (Main Construction)**.** *Fix a prime field $\mathbb{F}_p$. Given distinct primes $p_1, \ldots, p_\ell$ let $\mathbb{L}_i$ be the degree p_i extension of $\mathbb{F}_p$ and let $\mathbb{K}$ be the degree $n = \prod_{i=1}^{\ell} p_i$ extension of $\mathbb{F}_p$. Let $D_i = \mathrm{orbit}_{\mathbb{L}_i}(\{1, 2, 1 + p\})$ and let $\tilde{\mathcal{C}}_i$ be the affine-invariant code with degree set D_i, i.e.,*

$$\tilde{\mathcal{C}}_i = \{\mathrm{Trace}_{\mathbb{L}_i \to \mathbb{F}}(f) \mid f \in \mathbb{L}_i[X], \mathrm{supp}_{\deg}(f) \subseteq D_i\} = \mathrm{Code}(D_i).$$

Let $\mathcal{C}_i = \mathrm{lift}_{\mathbb{L}_i \nearrow \mathbb{K}}(\tilde{\mathcal{C}}_i)$. *Finally we let* $\mathcal{C} = \mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell) \subseteq \{\mathbb{K} \to \mathbb{F}_p\}$ *be the affine-invariant code defined as*

$$\mathcal{C} = \bigcap_{i=1}^{\ell} \mathcal{C}_i.$$

In other words, $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell) = \mathrm{Code}(D)$ *where* $D = D(\mathbb{F}_p; p_1, \ldots, p_\ell) = \bigcap_{i=1}^{\ell} \mathrm{lift}_{\mathbb{L}_i \nearrow \mathbb{K}}(D_i)$.

Remark 4.4 (Basic Properties)**.** *Every affine-invariant code of dimension greater than one contains every degree of p-weight one in its degree set, and so does* $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ *(since it must contain some degree of weight one, and then the* Trace *operator adds every degree of weight one into the degree set). This obvious fact will be used often later.*

Remark 4.5 (Sparsity of lifted code)**.** *Notice each "base-code"* $\tilde{C}_i \subseteq \{\mathbb{L}_i \to \mathbb{F}_p\}$ *has sparsity at most* 3*. (It can be verified to be isomorphic to a subcode of Reed-Muller code of degree* 2 *over* $\mathbb{F}_p$*, cf. [Ben-Sasson and Sudan, 2010, Lemma 3.7].) However, the lifted code* $\mathrm{lift}_{\mathbb{L}_i \nearrow \mathbb{K}}(\mathcal{C}_i)$ *has super-constant sparsity. To see this notice that* $\left\{ p + p^{jp_i} \;\middle|\; j = 0, \ldots, \frac{n}{p_i} - 1 \right\} \subseteq \mathrm{lift}_{\mathbb{L}_i \nearrow \mathbb{K}}(D_i)$ *and each element of this set has a distinct orbit in* $\mathbb{K}$*, which implies that the sparsity of* $\mathrm{lift}_{\mathbb{L}_i \nearrow \mathbb{K}}(D_i)$ *is at least* $[\mathbb{K} : \mathbb{L}_i] = n/p_i$.

The following two statements immediately prove our main Theorem 2.6. The first statement, proved in Section 5, follows directly from what is already known about affine-invariant codes but the second one, proved in Section 7, requires some new ideas which are exposed in Sections 6 and 7.

Lemma 4.6 ($\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ is an LDPC code)**.** *For every prime p there exists a positive constant k such that the following holds.* $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ *is an affine-invariant k-LDPC code.*

Theorem 4.7 ($\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ is not locally testable)**.** *For every prime p there exist positive constant* δ *such that the following holds for all* $s, c \in (0, 1]$ *such that* $s + c > 1$ *(cf. Remark 2.2). Given a* (q, δ, s, c)*-tester for* $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ *(as in Definition 4.3) and assuming* $p_1, \ldots, p_\ell \geq 5$*, we have* $q > (s + c - 1)(1 - 1/p)\ell$.

Let us see how these statements imply our main result.

Proof of Theorem 2.6. Let p_i denote the $(i+2)$nd smallest prime (such that $p_1 = 5$). Let $n_\ell = \prod_{i=1}^{\ell} p_i$ and recall that

$$\ell \geq \Omega(\log n_\ell / \log\log n_\ell). \tag{1}$$

This inequality can be derived from the prime number theorem. Consider the family of affine-invariant codes

$$\{\mathcal{C}_{n_\ell} = \mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell) \mid \ell = 1, 2, \ldots\}.$$

Lemma 4.6 proves the first part of Theorem 2.6. For the second part let δ be as in Theorem 4.7. Given a (q_ℓ, s, c)-tester for $\mathcal{C}_{n_\ell}$ Theorem 4.7 shows that

$$q_\ell \geq (s + c - 1)(1 - 1/p)\ell \geq \Omega(\log n_\ell / \log\log n_\ell)$$

where the last inequality uses (1). Letting γ' be the constant hidden inside the asymptotic notation of (1) and setting $\gamma = \gamma'(1 - 1/p)$ completes the proof of Theorem 2.6. □

5 $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ is an LDPC code

In this section we prove Lemma 4.6, the simpler of the two statements needed to prove Theorem 2.6. It will follow directly from Theorem 3.7.

Lemma 5.1 (Lifting a single-orbit code results in a single-orbit code)**.** *If $(\alpha, \lambda) \in \mathbb{L}^k \times \mathbb{F}^k$ is a k-s-o-c of $\mathcal{C} \subseteq \{\mathbb{L} \to \mathbb{F}_p\}$ then (α, λ) is also a k-s-o-c of $\mathrm{lift}_{\mathbb{L}\nearrow\mathbb{K}}(\mathcal{C})$.*

Proof. By Lemma 3.6 it is enough to show that for all $d' \in \{0, \ldots, |K| - 1\}$,

$$d' \in \mathrm{lift}_{\mathbb{L}\nearrow\mathbb{K}}(D) \text{ if and only if } \forall e \leq_p d', \sum_{j=1}^{k} \lambda_j \alpha_j^e = 0. \tag{2}$$

By Definition 4.1 the left hand side of (2) holds if and only if $(\mathrm{shadow}_p(d') \bmod |\mathbb{L}| - 1) \subseteq D$. The exponent e appearing in the right hand side of (2) can be replaced by $e \bmod |\mathbb{L}| - 1$ because $\alpha_i \in \mathbb{L}$. So to prove our lemma it suffices to show

$$\big(\mathrm{shadow}_p(d') \bmod |\mathbb{L}| - 1\big) \subseteq D \quad \Leftrightarrow \quad \forall e \leq_p d', \sum_{j=1}^{k} \lambda_j \alpha_j^{e \bmod (|\mathbb{L}|-1)} = 0 \tag{3}$$

For the forward implication ($\Rightarrow$) notice that every $e \in \mathrm{shadow}_p(d')$ satisfies $(e \bmod (|\mathbb{L}| - 1)) \in D$ so the right hand side of (3) follows from Lemma 3.6. For the reverse implication ($\Leftarrow$) we need the following claim, proved below.

Claim 5.2. *If $e \in (\mathrm{shadow}_p(d') \bmod (|\mathbb{L}| - 1))$ and $f \in \mathrm{shadow}_p(e')$, then there exists $f' \leq_p d'$ such that $f = f' \bmod (|\mathbb{L}| - 1))$.*

This claim shows that the right hand side of (3) holds for $\mathrm{shadow}_p(e \bmod (|\mathbb{L}| - 1))$ and this implies via Lemma 3.6 that $(e \bmod (|\mathbb{L}| - 1)) \in D$. Since this holds for all $e \leq_p d'$ we conclude that the left hand side of (3) holds as well and this completes the proof of the lemma. □

Proof of Claim 5.2. Suppose $[\mathbb{K} : \mathbb{F}_p] = n$. For integer $e < |\mathbb{K}|$ let $S(e) \subseteq \{0, \ldots, n-1\}$ be the unique multiset satisfying $e = \sum_{s \in S(e)} p^s$ and for a multiset $S \subseteq \{0, \ldots, n-1\}$ let $d(S) = \sum_{s \in S} p^s$. Assuming $e' \in \mathrm{shadow}_p(d')$ is an integer such that $e = e' \bmod (|\mathbb{L}| - 1)$ and letting $S(e) = \{s_1, \ldots, s_m\}$ notice $S(e')$ can be partitioned into m multisets $S_1, \ldots, S_m$ such that

$$S(e) = \{S(d(S_i) \bmod (|\mathbb{L}| - 1)) \mid i = 1, \ldots, m\}.$$

Supposing without loss of generality $S(f) = \{s_1, \ldots, s_{m'}\}$ for $m' \leq m$ the integer $f' = d(\cup_{i=1}^{m'} S_i)$ belongs to $\mathrm{shadow}_p(d')$ and has the property that $f' \bmod |\mathbb{L}| - 1$ equals f. This proves the claim. □

We now show that $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ is an LDPC code.

Proof of Lemma 4.6. We have

$$\tilde{\mathcal{C}}_i = \mathrm{Code}(\mathrm{orbit}_{\mathbb{L}_i}(\{1, 2, 1+p\}) = \mathrm{Code}(\{1, 2, 1+p\}) = \mathrm{Code}(\{0, 1, 2, p, 1+p\})$$

and so this code has sparsity at most 5. We can thus apply Part 2 of Theorem 3.7 to conclude $\tilde{\mathcal{C}}_i$ is characterized by a constraint of size k for some constant k independent of p_i. Lemma 5.1 implies that $\mathcal{C}_i = \mathrm{lift}_{\mathbb{L}_i \nearrow \mathbb{K}}(\tilde{\mathcal{C}}_i)$ is characterized by the same constraint of size k. In particular, we get that both $\tilde{\mathcal{C}}_i$ and $\mathcal{C}_i$ are k-LDPC codes. Since the interserction of k-LDPC codes is also k-LDPC codes, we conclude that $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell) = \cap_i \mathcal{C}_i$ is also a k-LDPC code. □

6 $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ is not $(\ell-1)$-single orbit characterizable

This section sets up machinery needed for the proof of Theorem 4.7 which shows that our code $\mathcal{C}$ is not locally testable. In the process we prove the strictly weaker statement (Lemma 6.1) that $\mathcal{C}$ is not single-orbit characterized. Along the way we introduce the concepts and tools needed to prove Theorem 4.7 in the next section.

Lemma 6.1. *Let p be an arbitrary prime and let $p_1, \ldots, p_\ell \geq 5$. Then, if $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ from Definition 4.3 has a k-s-o-c, then $k > \ell$.*

We prove the lemma at the end of this section, after developing the ingredients. The proof consists of two main steps. First, we analyze the degree set $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ of our codes and show some explicit integers that are contained in this set and some integers that are not. Proving the containments (or lack thereof) is easy — but the choice of these integers is done carefully so as to support the second step. In particular, we show that the elements of focus in $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ have a certain recursive description which turns out to be useful in the second step.

The second step is to use the analysis of $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ to prove that $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ does not have a local characterization. For this part, we consider any k-local constraint $(\alpha, \lambda) \in \mathbb{K}^k \times \mathbb{F}_p^k$ satisfied by all codewords in $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$. We use Lemma 3.6 to convert this into an implication that a certain matrix M with k columns has a non-empty kernel. We then define a sequence of ℓ matrices $M_1, \ldots, M_\ell = M$ with nested kernels, i.e.,

$$\ker(M_1) \supseteq \ker(M_2) \supseteq \ldots \supseteq \ker(M_\ell). \tag{4}$$

The crux of our proof is to show that if $\ker(M_i) = \ker(M_{i+1})$ then the constraint somehow "misses" testing membership in $\mathcal{C}_i$, and so it is not a characterization. This step relies on the recursive description of the members of $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ spotlighted in the first. We turn this recursive description into a simple relationship between M_i and M_{i-1} and then, in turn, into a relationship between their kernels.

6.1 Analysis of $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$

Definition 6.2. *For $j \in [\ell]$ let q_j be the integer satisfying*

$$\forall i \in [\ell],\ q_j \bmod p_i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

(The Chinese Remainder Theorem guarantees the existence of q_j.)

For $t \in [\ell]$ let Y_t be the set of integers

$$Y_t = \left\{ p^{\sum_{j=1}^{t} z_j q_j} \;\middle|\; z_j \in \{0,1\} \right\} = \begin{cases} \{1, p^{q_1}\} & t = 1 \\ Y_{t-1} \cup \{p^{q_t} \cdot d \mid d \in Y_{t-1}\} & t = 2, \ldots, \ell \end{cases} \tag{5}$$

Notice that the Y_t's have a nice recursive structure, which will become important in later sections. Our next lemma explains the relationship between the Y_t's and $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$.

Lemma 6.3. *1. For every $e \in Y_\ell$ and $e' \leq_p 1 + e$ we have $e' \in D(\mathbb{F}_p; p_1, \ldots, p_\ell)$.*

2. For all $t \in [\ell]$, we have $1 + p^{2q_t} \notin D(\mathbb{F}_p; p_1, \ldots, p_\ell)$, assuming $p_1, \ldots, p_\ell \geq 5$.

Proof. For Part (1), fix $e \in Y_\ell$. Let $z_1, \ldots, z_\ell \in \{0,1\}$ be such that $e = p^{\sum_j z_j q_j}$. Let $d = 1 + e$.

Notice d is of p-weight 2. Hence all integers $d' <_p d$ are of p-weight 1. Since $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ is nonempty and p-shadow-closed it follows that $d' \in D(\mathbb{F}_p; p_1, \ldots, p_\ell)$. So to prove the claim it is enough to show $d \in D(\mathbb{F}_p; p_1, \ldots, p_\ell)$. Recalling $D(\mathbb{F}_p; p_1, \ldots, p_\ell) = \bigcap_{i=1}^{\ell} \mathrm{lift}_{\mathbb{L}_i \nearrow \mathbb{K}}(D_i)$ it is enough to show that $d \in \mathrm{lift}_{\mathbb{L}_i \nearrow \mathbb{K}}(D_i)$ for all $i \in [\ell]$. Since $D_i = \mathrm{orbit}_{\mathbb{L}_i}(\{1, 2, 1+p\})$ this amounts to showing that

$$d \bmod (p^{p_i} - 1) \in D_i \quad \forall i \in [\ell].$$

By construction of q_j we have

$$p^{q_j} \bmod (p^{p_i} - 1) = \begin{cases} p & \text{if } i = j \\ 1 & \text{otherwise,} \end{cases}$$

so that

$$\begin{aligned} d \bmod (p^{p_i} - 1) &= 1 + \prod_j p^{z_j q_j} \bmod (p^{p_i} - 1) \\ &= 1 + p^{z_i q_i} \bmod (p^{p_i} - 1) = \begin{cases} 2 & \text{if } z_i = 0 \\ 1 + p & \text{if } z_i = 1 \end{cases} \end{aligned}$$

In both cases we have $d \bmod (p^{p_i} - 1) \in D_i$. Since this holds for all $i \in [\ell]$ this yields Part (1) of the lemma.

For Part (2), it is enough to prove that $1 + p^{2q_t} \bmod (p^{p_t} - 1) \notin D_t$. (Recall $D_t = \mathrm{orbit}_{\mathbb{L}_t}(\{1, 2, 1+p\})$.) Noting that $p^{q_t} \bmod (p^{p_t} - 1) = p$, we get that

$$1 + p^{2q_t} \bmod (p^{p_t} - 1) = p^2 + 1 \bmod (p^{p_t} - 1) = p^2 + 1.$$

The only way $1 + p^2 \in D_t$ is if $1 + p^2 = p^a(1+p) \bmod (p^{p_t} - 1)$ for some a, but inspection reveals this can not happen if $p_t \geq 5$. (It may happen if $p_t = 3$.) We conclude that $1 + p^{2q_t} \notin D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ as claimed. □

6.2 Analyzing constraints on $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$

We now fix a k-local constraint $(\alpha, \lambda) \in \mathbb{K}^k \times \mathbb{F}^k$ and consider a collection of matrices whose kernels turn out to hold the key to proving Lemma 6.1.

The natural matrix to consider at this stage might be the matrix $\tilde{M}$ whose rows are indexed by $d \in D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ and columns by $i \in [k]$ and where $\tilde{M}_{d,i} = \lambda_i \cdot \alpha_i^d$. The columns of this matrix sum to zero, implying the all ones vector is in its (right) kernel. Unfortunately this matrix does not have a nice enough structure to exploit, so we focus instead on a matrix whose entries are roughly α_i^{d-1} (we consider only d's of the form suggested by Lemma 6.3). In fact we define an entire sequence of matrices that are used to show the recursive structure of the final matrix that we care about.

We start with some generic notation. For any set S of non-negative integers, we let $M[S]$ be the $|S| \times k$ matrix with $M[S]_{i,j} = \lambda_j \cdot \alpha_j^i$ for $i \in S$ and $j \in [k]$. For an $m \times k$ matrix, M, we let $\ker(M)$ denote the set of vectors $\{x \in \mathbb{K}^k | Mx = 0\}$. We let $M^{\uparrow t}$ denote the matrix with $(M^{\uparrow t})_{ij} = (M_{ij})^t$.

Definition 6.4. *For $t \in [\ell]$, let Y_t be as in Definition 6.2. Then $M_t = M[Y_t]$.*

Our interest in M_t's stems from the fact that if (α, λ) is a constraint satisfied by $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$, then M_ℓ must have a non-empty kernel as pointed in the following lemma.

Lemma 6.5. *$\alpha \in \ker(M_t)$ for all $t \in [\ell]$.*

Proof. Fix $e \in Y_t$ and note that $(M_t\alpha)_e = \sum_{i=1}^k \lambda_i \alpha_i^{1+e}$. By Part (1) of Lemma 6.3, we have $1 + e \in D(\mathbb{F}_p; p_1, \ldots, p_\ell)$, and by Part (1) of Lemma 3.6 we have that $\sum_{i=1}^k \lambda_i \alpha_i^d = 0$ for every $d \in D(\mathbb{F}_p; p_1, \ldots, p_\ell)$. We conclude that every coordinate of the vector $M_t\alpha$ is zero, and thus $\alpha \in \ker(M_t)$. □

We now turn to upper bounding the dimension of the kernel of M_t. For this we need the following recursive description of M_t and its kernel.

Proposition 6.6 (Basic properties of $M_1, \ldots, M_\ell$)**.** *For $t = 2, \ldots, \ell$ we have*

$$M_t = \begin{pmatrix} M_{t-1} \\ M_{t-1}^{\uparrow p^{q_t}} \end{pmatrix}. \tag{6}$$

Consequently,

$$\ker(M_t) = \ker(M_{t-1}) \cap \ker\left(M_{t-1}^{\uparrow p^{q_t}}\right). \tag{7}$$

Proof. Follows directly from inspecting the rightmost side of (5) and noticing $\lambda^{p^{q_t}} = \lambda$ because $\lambda \in \mathbb{F}_p^k$. □

We are now ready to use the structure described above to study the kernels of the matrices M_t.

Lemma 6.7 (Kernel decay)**.** *For every $t \in \{2, \ldots, \ell\}$, $\ker(M_{t-1}) \supseteq \ker(M_t)$. Furthermore, if $\ker(M_{t-1}) = \ker(M_t)$ then $\ker(M_{t-1}) = \ker\left(M_{t-1}^{\uparrow p^{mq_t}}\right)$ for all positive integers m.*

Proof. The containment $\ker(M_{t-1}) \supseteq \ker(M_t)$ is immediate from Proposition 6.6.

We prove the second part by induction on $m \geq 1$. The base case ($m = 1$) follows from (7) because the assumption $\ker(M_t) = \ker(M_{t-1})$ implies

$$\ker(M_{t-1}) = \ker\left(M_{t-1}^{\uparrow p^{q_t}}\right). \tag{8}$$

For the induction step use the inductive hypothesis to assume

$$M_{t-1}^{\uparrow p^{mq_t}} v = 0 \quad \Leftrightarrow \quad M_{t-1} v = 0.$$

The operation of raising elements of $\mathbb{K}$ to power p^{q_t} is a (Frobenius) automorphism of $\mathbb{K}$ since $\mathbb{K}$ is an extension of $\mathbb{F}_p$. Raising both sides to this power we get

$$0 = \left(M_{t-1}^{\uparrow p^{mq_t}} v\right)^{\uparrow p^{q_t}} = M_{t-1}^{\uparrow p^{(m+1)q_t}} v^{\uparrow p^{q_t}} \quad \Leftrightarrow \quad 0 = (M_{t-1} v)^{\uparrow p^{q_t}} = M_{t-1}^{\uparrow p^{q_t}} v^{\uparrow p^{q_t}}.$$

Letting $u = v^{\uparrow p^{q_t}}$ and noticing the mapping $v \mapsto u$ is one-to-one (because raising to power p^{q_t} is invertible) we conclude

$$u \in \ker(M_{t-1}^{\uparrow p^{(m+1)q_t}}) \quad \Leftrightarrow \quad u \in \ker(M_{t-1}^{\uparrow p^{q_t}}) \Leftrightarrow u \in \ker(M_{t-1}).$$

The rightmost implication follows from the assumption (8). This completes the proof of the lemma. □

6.3 Proof of Lemma 6.1

We are now ready to prove the main lemma of this section.

Proof of Lemma 6.1. First we claim that for every $t \in \{2, \ldots, \ell\}$, it is the case that $\ker(M_t) \supsetneq \ker(M_{t-1})$. Assume for the sake of contradiction that this does not hold and $\ker(M_{t-1}) = \ker(M_t)$ for some t. Then by Lemma 6.5 we have $\alpha \in \ker(M_{t-1})$. Combining with Lemma 6.7 we further get $\alpha \in \ker\left(M_{t-1}^{\uparrow p^{mq_t}}\right)$ for all $m \geq 1$. Setting $m = 2$ we get $\alpha \in \ker\left((M_{t-1})^{\uparrow p^{2q_t}}\right)$. Now, since $1 = p^0 \in Y_{t-1}$ (cf. Definition 6.2), we get that $(\lambda_1\alpha_1, \ldots, \lambda_k\alpha_k)$ is a row of M_{t-1} and so $(\lambda_1\alpha_1, \ldots, \lambda_k\alpha_k)^{\uparrow p^{2q_t}}$ is a row of $(M_{t-1})^{\uparrow p^{2q_t}}$. The condition $\alpha \in \ker\left((M_{t-1})^{\uparrow p^{2q_t}}\right)$ yields:

$$0 = \sum_{i=1}^{k} \lambda_i^{p^{2q_t}} \cdot \alpha_i^{p^{2q_t}} \cdot \alpha_i = \sum_{i=1}^{k} \sum_{i=1}^{k} \lambda_i \alpha_i^{1+p^{2q_t}}.$$

We claim that this shows that $d = 1 + p^{2q_t}$ belongs to $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ and this contradicts Part (2) of Lemma 6.3. Indeed, the p-weight of d is 2 hence all $e <_p d$ have p-weight 1 so they belong to $D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ (see Remark 4.4). So we have

$$\sum_{i=1}^{k} \lambda_i \alpha_i^{1+p^{2q_t}} = 0, \forall e \leq_p d$$

which, by Part (2) of Lemma 3.6, implies that $d \in D(\mathbb{F}_p; p_1, \ldots, p_\ell)$, contradiction. We conclude that $\ker(M_{t-1}) \supsetneq \ker(M_t)$ as claimed.

To complete the proof we bound k. By Lemma 6.5 we have $\dim(\ker(M_\ell)) \geq 1$ because α is nonzero. Since $\dim(\ker(M_{t-1})) \geq \dim(\ker(M_t)) + 1$ for every t, we get that $\dim(M_1) \geq \ell$. But on the other hand, we trivially have $\dim(\ker(M_1)) < k$ because M_1 is nonzero. Thus we get $\ell < k$ as desired. □

7 Non-testability

In this section we prove our main theorem, Theorem 4.7. We follow the strategy for proving lower bounds on query complexity of linear codes suggested by Ben-Sasson et al. [2005], this strategy is summarized by the following proposition (which we describe in our notation). The strategy is defined with respect to canonical testers (cf. Definition 2.3) but given Claim 2.4 it also implies lower bounds for general testers. For μ a distribution over a set S let $s \sim \mu$ denote that s is sampled according to μ.

Proposition 7.1 (Strategy for proving lower bounds on query complexity, [Ben-Sasson et al., 2005])**.** *Let $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ be a linear code and let $\epsilon, \delta > 0$. If there exists a distribution μ supported on $\{\mathbb{K} \to \mathbb{F}\}$ satisfying:*

Distance: *The support of μ is on words that are δ-far from $\mathcal{C}$.*

Undetectability: *For every k-local constraint (α, λ) satisfied by $\mathcal{C}$,*

$$\Pr_{w \sim \mu}\left[\sum_{i=1}^{k} \lambda_i w(\alpha_i) \neq 0\right] < \epsilon. \tag{9}$$

Then any (q, δ, ϵ)-canonical tester for $\mathcal{C}$ satisfies $q > k$.

To follow the strategy we first define the distribution μ of "bad" words for a q-tester, then focus on an arbitrary k-local constraint satisfied by $\mathcal{C}$ and bound the probability of (9) using the machinery from the previous section.

Definition 7.2 (Bad distribution)**.** *Given $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ and $i \in [\ell]$ let $\mathcal{B}_i \subseteq \{\mathbb{K} \to \mathbb{F}_p\}$ be the affine-invariant code $\mathcal{B}_i = \mathrm{Code}(\mathcal{D}_i)$ where*

$$\mathcal{D}_i = \mathrm{orbit}_{\mathbb{K}}\left(\mathrm{shadow}_p\left(\left\{1 + p^{2q_i}\right\}\right)\right).$$

Let μ be the distribution obtained by (i) sampling $i \in [\ell]$ uniformly and then (ii) sampling $f \in \mathcal{B}_i \setminus \mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ uniformly at random.

Claim 7.3 (μ has the distance property)**.** *μ is well-defined, i.e., its support is over a nonempty set. Furthermore, for every prime p there exists a constant $\delta > 0$ such that μ is supported on words that are δ-far from $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$.*

Proof. To see the first part of the claim notice Part (2) of Lemma 6.3 says that $1 + p^{2q_i} \notin D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ and this implies $\mathcal{B}_i \not\subseteq \mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$. So $\mathcal{B}_i \setminus \mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell) \neq \emptyset$ and we see that μ is well-defined.

Moving on to the second part, by construction $\mathcal{B}_i$ is affine-invariant and has sparsity at most $|\text{shadow}_p(\{1 + p^{2q_i}\})| = |\{0, 1, p^{2q_i}, 1 + p^{2q_i}\}| = 4$. The distance of $\mathcal{B}_i$ is thus implied by the first part of Theorem 3.7. In particular, each $w \in \mathcal{B}_i \setminus \mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ is δ-far from $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ where δ is the constant guaranteed by Theorem 3.7. □

Lemma 7.4 (μ is undetectable)**.** *For any k-local constraint $(\alpha = (\alpha_1, \ldots, \alpha_k) \in \mathbb{K}^k, \lambda = (\lambda_1, \ldots, \lambda_k) \in \mathbb{F}_p^k)$ satisfied by $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ we have*

$$\Pr_{f \sim \mu}[f \text{ does not satisfy } (\alpha, \lambda)] < \frac{k-1}{\ell}. \tag{10}$$

Proof. Let $M_t = M[Y_t]$ be as given in Definition 6.4 and recall that $\ker(M_{t-1}) \supseteq \ker(M_t)$ for $t = 2, \ldots, \ell$. Let $T \subset \{2, \ldots, \ell\}$ be the set of indices t satisfying

$$\ker(M_{t-1}) \supsetneq \ker(M_t)$$

and notice $|T| < k-1$ because $\dim(\ker(M_1)) < k$ and $\dim(\ker(M_\ell)) \geq 1$ (the last inequality follows from Part (1) of Lemma 6.3). We claim that for each $t \notin T$ we have $\sum_{j=1}^k \lambda_j \alpha_j^{d_t} = 0$, where $d_t = 1 + p^{2q_t}$. Indeed, by Lemma 6.7, for $t \notin T$ we have $\ker(M_{t-1}) = \ker\left((M_{t-1})^{\uparrow p^{2q_t}}\right)$. As in the proof of Lemma 6.1 this implies that $\alpha \in \ker\left((M_{t-1})^{\uparrow p^{2q_t}}\right)$, or, equivalently $\sum_{i=1}^k \lambda_i \alpha_i^{1+p^{2q_t}} = 0$ as claimed.

Next we claim that for $i \notin T$, the code $\mathcal{B}_i$ satisfies (α, λ). To show this it suffices to show, by Part (1) of Lemma 3.6, that every $d \in \mathcal{D}_i = \text{Deg}(\mathcal{B}_i)$ satisfies $\sum_{j=1}^k \lambda_j \alpha_j^d = 0$. If d is of p-weight one, then $d \in D(\mathbb{F}_p; p_1, \ldots, p_\ell)$ (by Remark 4.4) and since $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ satisfies (α, λ) we have $\sum_{j=1}^k \lambda_j \alpha_j^d = 0$. Else, $d = p^a \cdot d_i$ for some integer a (since these are the only elements in $\mathcal{D}_i$ of weight 2) and in this case also we have $\sum_{j=1}^k \lambda_j \alpha_j^d = \left(\sum_{j=1}^k \lambda_j \alpha_j^{d_i}\right)^{p^a} = 0$. We conclude that every $\mathcal{B}_i$ satisfies the k-constraint (α, λ).

We are almost done. Recall that $f \sim \mu$ is chosen by picking $i \in [\ell]$ uniformly and then picking $f \in \mathcal{B}_i \setminus \mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$. If $i \notin T$, then the constraint (α, λ) is satisfied, and this happens with probability at least $1 - (k-1)/\ell$. The lemma follows. □

We can now complete the proof of Theorem 4.7. Below we use q to denote the locality of tests (as opposed to k).

Proof of Theorem 4.7. Use the strategy given by Proposition 7.1. Given a prime p let $\delta > 0$ be the constant guaranteed by Lemma 7.3. This lemma shows that μ is supported on words that are δ-far from $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$. Fix a q-test for $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$, specified by the constraint (α, λ). Lemma 7.4 shows that the probability that this constraint rejects words sampled from μ is less than q/ℓ. So by Proposition 7.1 any canonical q-tester for $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$ rejects words sampled from μ with probability less than q/ℓ. By Claim 2.4 the existence of a (q, δ, s, c)-tester for $\mathcal{C}(\mathbb{F}_p; p_1, \ldots, p_\ell)$

implies the existence of a canonical q-tester with soundness at least $(s+c-1))(1-1/p)$. We conclude

$$q/\ell > (s+c-1)(1-1/p)$$

and this completes the proof. □

8 Closed degree sets specify affine-invariant codes

In this section we prove Lemma 3.5. The claim of the lemma is implicit in several different works and we basically give the relevant pointers as well as some one line proofs.

Proof. The fact that $\mathrm{Deg}(\mathcal{C})$ is shadow closed is proved in [Ben-Sasson and Sudan, 2010, Lemma 3.3] and the fact that $\mathcal{C} = \mathrm{Code}(\mathrm{Deg}(\mathcal{C}))$ is Lemma 4.2 there.

To see that $\mathrm{Deg}(\mathcal{C})$ is orbit-closed we note that for a function $f(x) = \sum_d f_d x^d$ mapping $\mathbb{K} \to \mathbb{F}_p$ it holds that

$$f(x)^p = f(x) \pmod{x^{|K|} - x}$$

and so $f_{pd} = (f_d)^p$. In particular, $f_d \neq 0$ iff $f_{dp} \neq 0$. Thus, $d \in \mathrm{Deg}(\mathcal{C})$ iff $dp \in \mathrm{Deg}(\mathcal{C})$ and so $\mathrm{Deg}(\mathcal{C})$ is orbit-closed.

In the other direction, recall that

$$\mathrm{Code}(D) = \{\mathrm{Trace} \circ f | f \in \mathbb{K}[X], \mathrm{supp}_{\deg}(f) \subseteq D\} .$$

It follows that $\mathrm{Deg}(\mathrm{Code}(D))$ is contained in $\mathrm{orbit}(D) = D$. Indeed, if $f = \sum_{d \in D} f_d x^d \in \mathrm{Code}(D)$ then

$$\mathrm{Trace}(f) = \sum_{i=0}^{[\mathbb{K}:\mathbb{F}_p]-1} f^{p^i} = \sum_{i=0}^{[\mathbb{K}:\mathbb{F}_p]-1} \sum_{d \in D} (f_d)^{p^i} x^{dp^i}$$

and since D is orbit-closed all the degrees in the RHS are in D and so $\mathrm{supp}_{\deg}(\mathrm{Trace}(f)) \subseteq D$. Containment in the other direction is clear.

Finally, to see that $\mathrm{Code}(D)$ is affine-invariant when D is shadow-closed and orbit-closed, we observe that for every $f = \sum_{d \in D} f_d x^d$ such that $\mathrm{Trace}(f) \in \mathrm{Code}(D)$, and for every $a \in \mathbb{K}^*$ and $b \in \mathbb{K}$ it holds that

$$\begin{aligned} \mathrm{Trace}(f(ax+b)) &= \sum_{i=0}^{[\mathbb{K}:\mathbb{F}_p]-1} \sum_{d \in D} (f_d)^{p^i} (ax+b)^{dp^i} = \sum_{i=0}^{[\mathbb{K}:\mathbb{F}_p]-1} \sum_{d \in D} (f_d)^{p^i} \sum_{e \leq_p d} x^{ep^i} a^{ep^i} b^{(d-e)p^i} \\ &= \sum_{d \in D} \sum_{e \leq_p d} \sum_{i=0}^{[\mathbb{K}:\mathbb{F}_p]-1} a^{ep^i} b^{(d-e)p^i} (f_d)^{p^i} x^{ep^i} . \end{aligned}$$

Since D is shadow-closed and orbit-closed, each degree ep^i in the sum above is also in D. It follows that $\mathrm{Trace}(f(ax+b))$ is supported on D and therefore is in $\mathrm{Code}(D)$. This proves that $\mathrm{Code}(D)$ is affine-invariant. □

9 Relating degree sets to constraints and characterizations

In this section we prove Lemma 3.6. The lemma is two-fold, and we start by expressing the fact that an affine-invariant code satisfies a constraint in terms of a condition on the degree set of the code.

Lemma 9.1. *Let $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ be an affine-invariant code with degree set $D = \mathrm{Deg}(\mathcal{C})$. Let $(\alpha, \lambda) \in \mathbb{K}^k \times \mathbb{F}^k$ be a k-constraint. Then $\mathcal{C}$ satisfies (α, λ) if and only if $\sum_{i=1}^k \lambda_i \alpha_i^d = 0$ for every $d \in D$.*

The proof of the lemma relies on the following "monomial extraction" result, given by [Kaufman and Sudan, 2008, Lemma 4.2].

Lemma 9.2 (Monomial extraction Kaufman and Sudan [2008])**.** *Let $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ be an affine-invariant code with degree set $D = \mathrm{Deg}(\mathcal{C})$. Then for every d in D, the monomial $f(x) = x^d$ belongs to the code.*

Proof of 9.1. Suppose that $\mathcal{C}$ satisfies (α, λ), so that f is in $\mathcal{C}$ if and only if $\sum_{i=1}^k \lambda_i f(\alpha_i) = 0$. In particular, for any $d \in D$, the monomial $f(x) = x^d$ belongs to $\mathcal{C}$ by Lemma 9.2 and satisfies $\sum_{i=1}^k \lambda_i \alpha_i^d = 0$

Conversely, suppose that (α, λ) is such that $\sum_{i=1}^k \lambda_i \alpha_i^d = 0$ for every $d \in D$. Let $f(x) = \sum_{d \in D} f_d x^d$ be in $\mathcal{C}$. Then

$$\sum_{i=1}^k \lambda_i f(\alpha_i) = \sum_{i=1}^k \lambda_i \sum_{d \in D} f_d \alpha_i^d = \sum_{d \in D} f_d \sum_{i=1}^k \lambda_i \alpha_i^d = 0.$$

□

We now relate the existence of a k-s-o-c of C to a necessary condition on the degree set of $\mathcal{C}$.

Lemma 9.3. *Let $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ be an affine-invariant code with degree set $D = \mathrm{Deg}(\mathcal{C})$. Let $(\alpha, \lambda) \in \mathbb{K}^k \times \mathbb{F}^k$ be a k-constraint. Then the following holds.*
If (α, λ) is a k-s-o-c of $\mathcal{C}$ then we have: $d \in D \Leftrightarrow \forall e \leq_p d$, $\sum_{i=1}^k \lambda_i \alpha_i^e = 0$.

Proof. Let (α, λ) be a k-s-o-c of $\mathcal{C}$, i.e., f is in C if and only if for all $a \in \mathbb{K}^*, b \in \mathbb{K}$,

$$\sum_{i=1}^k \lambda_i f(a\alpha_i + b) = 0.$$

Let d be in D. Then every $e \leq_p d$ is also in D (recall that, by Lemma 3.5, D is p-shadow closed) and is such that $\mathrm{Trace}(\beta x^e) \in \mathcal{C}$ for all $\beta \in \mathbb{K}^*$ (since $\mathcal{C} = \mathrm{Code}(D)$). Thus for all a, b and all β,

$$0 = \sum_{i=1}^k \lambda_i \mathrm{Trace}\left(\beta(\alpha_i a + b)^e\right) = \mathrm{Trace}\left(\beta \sum_{i=1}^k \lambda_i (\alpha_i a + b)^e\right).$$

In particular, Trace $\left(\beta \sum_{i=1}^{k} \lambda_i \alpha_i^e\right) = 0$ for every β. But this is true if and only if $\sum_{i=1}^{k} \lambda_i \alpha_i^e = 0$. Conversely, assume that d is such that $\forall e \leq_p d$, $\sum_{i=1}^{k} \lambda_i \alpha_i^e = 0$. Noting that

$$\sum_{i=1}^{k} \lambda_i \text{Trace}\left((\alpha_i a + b)^d\right) = \text{Trace}\left(\sum_{e \leq_p d} \left(\sum_i \lambda_i \alpha_i^e\right) a^e b^{d-e}\right),$$

we see that $\sum_{i=1}^{k} \lambda_i \text{Trace}\left((\alpha_i a + b)^d\right) = 0$ for every $a \in \mathbb{K}^*$ and $b \in \mathbb{K}$. Hence, $\text{Trace}(x^d) \in \mathcal{C}$ and $d \in \text{Deg}(\mathcal{C}) = D$. □

Finally, we show that the necessary condition in Lemma 9.3 is in fact a sufficient condition for a code to be single-orbit.

Lemma 9.4. *Let $\mathcal{C} \subseteq \{\mathbb{K} \to \mathbb{F}\}$ be an affine-invariant code with degree set $D = \text{Deg}(\mathcal{C})$. Let $(\alpha, \lambda) \in \mathbb{K}^k \times \mathbb{F}^k$ be a k-constraint. Then the following holds.*
If (α, λ) is such that $d \in D \Leftrightarrow \forall e \leq_p d$, $\sum_{i=1}^{k} \lambda_i \alpha_i^e = 0$, then (α, λ) is a k-s-o-c of $\mathcal{C}$.

To prove this lemma, we will need to look at this condition on degree sets of single-orbit codes in yet another way. The following claim will provide us with the tools to view the condition differently.

Claim 9.5. *Let $(\alpha, \lambda) \in \mathbb{K}^k \times \mathbb{F}^k$ and consider an integer $d \pmod{|\mathbb{K}| - 1}$. Then the following are equivalent.*

(i) $\forall e \leq_p d$, $\sum_{i=1}^{k} \lambda_i \alpha_i^e = 0$.

(ii) $\sum_{i=1}^{k} \lambda_i (\alpha_i x + y)^d \equiv 0$ *as a polynomial in x and y.*

Proof. Notice that

$$\begin{aligned}\sum_{i=1}^{k} \lambda_i (\alpha_i x + y)^d &= \sum_{i=1}^{k} \lambda_i \sum_{e \leq_p d} \alpha_i^e x^e y^{d-e} \\ &= \sum_{e \leq_p d} \left(\sum_{i=1}^{k} \lambda_i \alpha_i^e\right) x^e y^{d-e}.\end{aligned}$$

Since the degree is smaller than the field size, this is a formal equality of polynomials (in the variables x and y). Hence,

$$\sum_{i=1}^{k} \lambda_i (\alpha_i x + y)^d \equiv 0 \Leftrightarrow \sum_{e \leq_p d} \left(\sum_{i=1}^{k} \lambda_i \alpha_i^e\right) x^e y^{d-e} \equiv 0 \Leftrightarrow \forall e \leq_p d, \sum_{i=1}^{k} \lambda_i \alpha_i^e = 0.$$

□

Proof of Lemma 9.4. Assume that for all d, $d \in D \Leftrightarrow \forall e \leq_p d$, $\sum_{i=1}^{k} \lambda_i \alpha_i^e = 0$. Equivalently, by Claim 9.5, we have that

$$d \in D \Leftrightarrow \sum_{i=1}^{k} \lambda_i (\alpha_i x + y)^d \equiv 0.$$

Let $f = \sum_{d \in D} f_d x^d$ be in $\mathcal{C}$ and notice that any $d \in \mathrm{supp}_{\deg}(f)$ satisfies $\sum_{i=1}^{k} \lambda_i (\alpha_i a + b)^d = 0$ for any $a \in \mathbb{K}^*$ and $b \in \mathbb{K}$. Thus for all such a, b,

$$\sum_{i=1}^{k} \lambda_i f(\alpha_i a + b) = \sum_{d} f_d \sum_{i=1}^{k} \lambda_i (\alpha_i a + b)^d = 0.$$

Conversely, suppose $f \in \{\mathbb{K} \to \mathbb{F}\}$ is such that $\sum_{i=1}^{k} \lambda_i f(\alpha_i a + b) = 0$ for every $a \in \mathbb{K}^*$ and $b \in \mathbb{K}$. Define the code $\mathcal{C}'$ as the smallest linear affine-invariant family containing f, that is,

$$\mathcal{C}' = \{\sum_{a,b} \gamma_{ab} f(xa + b) \,|\, a \in \mathbb{K}^*, b \in \mathbb{K}, \gamma_{ab} \in \mathbb{F}\}.$$

For every degree $d \in \mathrm{supp}_{\deg}(f)$, d is in $\mathrm{Deg}(\mathcal{C}')$. As $\mathcal{C}'$ is affine-invariant, every $e \leq_p d$ also belongs to $\mathrm{Deg}(\mathcal{C}')$. Hence, for all $e \leq_p d$ and all $\beta \in \mathbb{K}$, $\mathrm{Trace}(\beta x^e) \in \mathcal{C}'$. But notice that any $g \in \mathcal{C}'$ satisfies $\sum_i \lambda_i g(\alpha_i a + b) = 0$ for any $a \in \mathbb{K}^*$ and $b \in \mathbb{K}$. Thus,

$$\sum_i \lambda_i \mathrm{Trace}(\beta(\alpha_i a + b)^e) = \mathrm{Trace}\left(\beta \sum_i \lambda_i (\alpha_i a + b)^e\right) = 0 \quad \forall a,\, b,$$

and in particular $\mathrm{Trace}(\beta \sum_i \lambda_i \alpha_i^e) = 0$. But this holds for all β if and only if $\sum_i \lambda_i \alpha_i^e = 0$.

Thus for every degree $d \in \mathrm{supp}_{\deg}(f)$, it holds that $\forall e \leq_q d, \sum_i \lambda_i \alpha_i^e = 0$. Therefore, $d \in D$ and hence $f \in \mathcal{C}$. □

We now conclude the proof of Lemma 3.6.

Proof of Lemma 3.6. The first claim in the lemma is exactly Lemma 9.1. One direction of the second claim is given by Lemma 9.3 and the other direction by Lemma 9.4. □

References

Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing Reed-Muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005.

László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 21–32, 1991.

Eli Ben-Sasson and Madhu Sudan. Limits on the rate of locally testable affine-invariant codes. *Electronic Colloquium on Computational Complexity (ECCC)*, (108), 2010. URL http://eccc.hpi-web.de/eccc-reports/2010/TR10-108/index.html.

Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF properties are hard to test. *SIAM J. on Computing*, 35(1):1–21, 2005. URL http://epubs.siam.org/SICOMP/volume-35/art_44544.html.

Eli Ben-Sasson, Venkatesan Guruswami, Tali Kaufman, Madhu Sudan, and Michael Viderman. Locally testable codes require redundant testers. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC)*, pages 52–61, 2009.

Oded Goldreich and Tali Kaufman. Proximity oblivious testing and the role of invariances. *Electronic Colloquium on Computational Complexity (ECCC)*, (058), 2010. URL http://eccc.hpi-web.de/eccc-reports/2010/TR10-058/index.html.

Oded Goldreich and Madhu Sudan. Locally testable codes and pcps of almost-linear length. *J. ACM*, 53(4):558–655, 2006.

Elena Grigorescu, Tali Kaufman, and Madhu Sudan. 2-transitivity is insufficient for local testability. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity (CCC)*, pages 259–267, 2008.

Elena Grigorescu, Tali Kaufman, and Madhu Sudan. Succinct representation of codes with applications to testing. In *APPROX-RANDOM, volume 5687 of Lecture Notes in Computer Science*, pages 534–547, 2009.

Tali Kaufman and Simon Litsyn. Almost orthogonal linear codes are locally testable. In *FOCS*, pages 317–326. IEEE Computer Society, 2005. ISBN 0-7695-2468-0.

Tali Kaufman and Shachar Lovett. New extension of the weil bound for character sums with applications to coding. *Electronic Colloquium on Computational Complexity (ECCC)*, (065), 2010. URL http://eccc.hpi-web.de/eccc-reports/2010/TR10-065/index.html.

Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *48th Symposium on Foundations of Computer Science (FOCS)*, pages 590–600, 2007.

Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 403–412, 2008.

Florence J. MacWilliams and Neil J. A. Sloane. *The theory of error-correcting codes*. North-Holland Amsterdam, 1978.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de