Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 42 (2016)

Universal Locally Testable Codes*

Oded Goldreich
Weizmann Institute of Science
oded.goldreich@weizmann.ac.il

Tom Gur
Weizmann Institute of Science
tom.gur@weizmann.ac.il

November 25, 2016

Abstract

We initiate a study of "universal locally testable codes" (universal-LTCs). These codes admit local tests for membership in numerous possible subcodes, allowing for testing properties of the encoded message. More precisely, a universal-LTC $C : \{0,1\}^k \to \{0,1\}^n$ for a family of functions $\mathcal{F} = \{f_i : \{0,1\}^k \to \{0,1\}\}_{i \in [M]}$ is a code such that for every $i \in [M]$ the subcode $\{C(x) : f_i(x) = 1\}$ is locally testable.

We show a "canonical" $O(1)$-local universal-LTC of length $\widetilde{O}(M \cdot s)$ for any family $\mathcal{F}$ of M functions such that every $f \in \mathcal{F}$ can be computed by a circuit of size s, and establish a lower bound of the form $n = M^{1/O(k)}$, which can be strengthened to $n = M^{\Omega(1)}$ for any $\mathcal{F}$ such that every $f, f' \in \mathcal{F}$ disagree on a constant fraction of their domain.

*This work previously appeared as the first part of the ECCC Technical Report 16-042 (original version) [GG16a]. The second part now appears separately in [GG16b].

Research was partially supported by the Israel Science Foundation (grant number 671/13) and Irit Dinur's ERC grant number 239985.

ISSN 1433-8092

Contents

1 Introduction **1**
- 1.1 The Notion of Universal-LTC . . . 1
- 1.2 Our Results . . . 2
- 1.3 Previous Version and Universal Locally Verifiable Codes . . . 3

2 Preliminaries **3**
- 2.1 Locally Testable and Decodable Codes . . . 4

3 The Definition of Universal Locally Testable Codes **5**

4 The Canonical Universal-LTC **6**
- 4.1 Preliminaries: PCP of proximity . . . 6
- 4.2 Consistency-Testable Bundles . . . 7
- 4.3 Proof of Theorem 4.1 . . . 10

5 General Lower Bounds **10**

6 Trading off Length for Locality **12**
- 6.1 Universal-LTCs of Nearly-Linear Length . . . 12
- 6.2 The Actual Tradeoff . . . 13
- 6.3 Lower Bounds for Universal-LTCs for Juntas . . . 14

A On Obtaining Locally Decodable Codes from Universal-LTCs **16**

B Proof of Proposition 6.3 **18**

1 Introduction

Locally testable codes [FS95, RS96, GS06] are error-correcting codes that have local procedures for ascertaining the integrity of purported codewords. More accurately, a code C is a locally testable code (LTC) if there exists a probabilisitic algorithm (tester) that gets a proximity parameter $\varepsilon > 0$, makes a small number of queries to a string w, and with high probability accepts if w is a codeword of C and rejects if w is ε-far from C. The query complexity of the tester is the number of queries that it makes (also referred to as the locality of the LTC).

1.1 The Notion of Universal-LTC

In this work we initialize a study of a generalization of the notion of LTCs, which we call universal locally testable codes. A universal-LTC is a code that not only admits a local test for membership in the code C but also a local test for membership in a *family of subcodes of* C. More precisely, a binary code $C : \{0,1\}^k \to \{0,1\}^n$ is a q-local universal-LTC for a family of functions $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i\in[M]}$ if for every $i \in [M]$ the subcode $\Pi_i \coloneqq \{C(x) : f_i(x) = 1\}$ is locally testable with query complexity q. Viewed in an alternative perspective, such codes allow for testing properties of the encoded *message*; that is, testing whether $C(x)$ is an encoding of a message x that satisfies a function $f_i \in \mathcal{F}$.

Universal-LTCs implicit in previous works. We note that the notion of universal-LTCs is implicit in the literature. For instance, the *long code* [BGS98], which maps a message to its evaluations under all Boolean functions, can be thought of as the "ultimate" universal-LTC for all Boolean functions. To see this, recall that the long code is both locally testable and *correctable* (i.e., there exists a local algorithm that can recover any bit of a slightly corrupted codeword). Now, observe that we can test a subcode $\{\mathsf{LC}(x) : f(x) = 1\}$, where $\mathsf{LC} : \{0,1\}^k \to \{0,1\}^{2^{2^k}}$ is the long code and $f : \{0,1\}^k \to \{0,1\}$ is some Boolean function, by first running the codeword test (and rejecting if it rejects), and then running the local correcting algorithm with respect to the bit in $\mathsf{LC}(x)$ that corresponds to the evaluation of f on x. Note, however, the ability to test all subcodes comes at the cost of great redundancy, since the length of the long code is *doubly exponential* in the length of the message.

By an analogous argument, the *Hadamard code*, which maps a message to its evaluations under all *linear* Boolean functions, can be thought of as a universal-LTC for all *linear* Boolean functions. Note that the length of the Hadamard code is *exponential* in the length of the message. Another example is the inner PCP for satisfiability of quadratic equations [ALM+98], wherein the (exponentially long) PCP oracle is an encoding of an assignment, *independent from the set of quadratic equations it allegedly satisfies*. Hence, this PCP is an "universal" encoding that admits a local test for the satisfiability of *any* function that is given by a set of quadratic equations, and so it can be thought of as a universal-LTC for quadratic equations.

In this work, we ask whether universal-LTCs can be constructed for any family of functions $\mathcal{F}$, and what are the optimal parameters (i.e., the code's length, locality, and number of subcodes for which it admits a local test) that can be obtained by universal-LTCs.

Universal (relaxed) Locally Decodable Codes. Before proceeding to present our results, we highlight a close connection between universal-LTCs and a universal generalization of the notion

of relaxed local decodability. Recall that a code is said to be a relaxed locally decodable code (relaxed-LDC) [BSGH+06] if for every location i in the message there exists a local algorithm (decoder) that is given query access to an alleged codeword, and satisfies the following: If the codeword is valid, the decoder successfully decodes the i'th symbol, and if the codeword is corrupted, the decoder, with high probability, either decodes correctly or rejects (indicating it detected a corruption in the codeword). It turns out that universal-LTCs immediately imply a generalization of the notion of relaxed-LDCs, which we describe next. (We also note that, under certain conditions, universal-LTCs imply (non-relaxed) local decodability, see Appendix A.)

We define a universal relaxed locally decodable code (in short, universal-LDC) for a family of functions $\mathcal{F}$ (analogously to universal-LTCs) as a relaxed-LDC wherein, instead of local procedures for (relaxed) decoding of bits of the message x, we have local procedures for (relaxed) decoding of the value of $f(x)$ for every $f \in \mathcal{F}$.

Now, let $\mathcal{F}$ be a family of Boolean functions. Observe that a universal-LTC for $\mathcal{F} \cup (1 - \mathcal{F})$ (i.e., a code with a tester $T_{f,b}$ for each subcode $\{C(x) : f(x) = b\}$, where $f \in \mathcal{F}$ and $b \in \{0, 1\}$) implies a universal-LDC for $\mathcal{F}$, which is also locally *testable*, and vice versa. To see this, consider the following local decoding procedure for $f \in \mathcal{F}$: To decode $f(x)$, invoke $T_{f,0}$ and $T_{f,1}$. If one tester accepted and the other rejected, rule according to the accepting tester, and otherwise reject. The reader may verify that this is indeed a (relaxed) local decoding procedure (see Appendix A for discussion and generalizations). For the other direction, to test the subcode $\{C(x) : f(x) = 1\}$, first run the codeword test, then decode the value of $f(x)$ and accept if and only if it equals 1 (i.e., a decoded value of 0 and a decoding error both cause rejection). We remark that all universal-LTCs in this work can be easily extended to families of the type $\mathcal{F} \cup (1 - \mathcal{F})$, and thus we also obtain analogous results for universal-LDCs.

On "uniformity" with respect to $\mathcal{F}$. For simplicity, we defined universal LTCs and LDCs in a "non-uniform" manner with respect to the family of functions $\mathcal{F}$; that is, we required that for every function $f \in \mathcal{F}$, there exists a testing or decoding procedure. A stronger, "$\mathcal{F}$-uniform", definition would require that there exists a procedure that receives $f \in \mathcal{F}$ as a parameter and tests or decodes with respect to f. We remark that all of our upper bounds can be easily adapted to satisfy the stronger $\mathcal{F}$-uniform condition, while our lower bounds hold even without this condition.

1.2 Our Results

To simplify the presentation of our results, throughout the introduction we fix the proximity parameter ε to a small constant, and when we refer to "codes", we shall actually mean error-correcting codes with linear distance. Our first result shows "canonical" universal-LTCs for *any* family of functions.

Theorem 1 (informally stated, see Theorem 4.1)**.** *Let $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i \in [M]}$ be any family of Boolean functions that can each be computed by a size $s = s(k)$ circuit. Then, there exists a (one-sided error)* universal-LTC *$C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(M \cdot s)}$ for $\mathcal{F}$ with query complexity $O(1)$.*

We complement the foregoing"canonical" universal-LTC with a general lower bound on the query complexity of universal-LTCs, as a function of the encoding's length and number of subcodes for which it admits a local test.

Theorem 2 (informally stated, see Theorem 5.1)**.** *Let* $\mathcal{F} = \{f_i : \{0,1\}^k \to \{0,1\}\}_{i\in[M]}$ *be a family of* distinct *Boolean functions. Then, every* universal-LTC $C : \{0,1\}^k \to \{0,1\}^n$ *for* $\mathcal{F}$ *must have query complexity* $\Omega(\log\log M - \log\log n - \log k)$*. Furthermore, if the functions in* $\mathcal{F}$ *are* pairwise far *(i.e.,* $\Pr_x[f_i(x) \neq f_j(x)] = \Omega(1)$ *for every* $i \neq j$*), then the query complexity is* $\Omega(\log\log M - \log\log n)$*.*

Note that $\log\log M - \log\log n = O(1)$ implies a lower bound of $n \geq M^{\Omega(1)}$. In contrast, recall that Theorem 1 shows an upper bound of $n = \widetilde{O}(M \cdot s)$, where s bounds the circuit size for computing each $f \in \mathcal{F}$. Thus, for sufficiently large families of pairwise-far functions, Theorem 2 shows that the length of the canonical universal-LTC (in Theorem 1) is optimal, up to a constant power. This raises the question of whether the aforementioned slackness can be removed. We answer this question to the affirmative, albeit for a specific family of functions.

Specifically, we show a universal-LTC $C : \{0,1\}^k \to \{0,1\}^{k^{1.01}}$ for a family of $M = \binom{k}{m} \cdot 2^{2^m}$ functions, namely the family of m-juntas,[1] with query complexity $\tilde{O}(m)$; note that for a large constant m, the number of functions M is an arbitrarily large *polynomial* in the code's length $k^{1.01} < M^{1.01/m}$, whereas for the canonical universal-LTC the length is linear in M.

In addition, note that the lower bound in Theorem 2 allows for a tradeoff between the universal-LTC's length and locality (i.e., query complexity), whereas Theorem 1 only shows universal-LTCs in the constant locality regime. In Section 6 we show that for the family of m-juntas, there exists a universal-LTC that allows for a tradeoff between locality and length. (See Proposition 6.3 for a precise statement.)

1.3 Previous Version and Universal Locally Verifiable Codes

This work appeared previously as a part of our technical report [GG16a], which contained the foregoing results regarding universal-LTCs as well as results regarding a related notion called "universal locally *verifiable* codes". Since this combination caused the latter notion and results to be missed, we chose to split the original version into two parts. The current part contains the material regarding universal-LTCs. The part regarding universal locally verifiable codes appears in a companion paper [GG16b].

2 Preliminaries

We begin with standard notations:

- We denote the *absolute distance*, over alphabet Σ, between two strings $x \in \Sigma^n$ and $y \in \Sigma^n$ by $\Delta(x,y) := |\{x_i \neq y_i : i \in [n]\}|$ and their *relative distance* by $\delta(x,y) := \frac{\Delta(x,y)}{n}$. If $\delta(x,y) \leq \varepsilon$, we say that x is ε-close to y, and otherwise we say that x is ε-far from y. Similarly, we denote the *absolute distance* of x from a non-empty set $S \subseteq \Sigma^n$ by $\Delta(x,S) := \min_{y\in S}\Delta(x,y)$ and the *relative distance* of x from S by $\delta(x,S) := \min_{y\in S}\delta(x,y)$. If $\delta(x,S) \leq \varepsilon$, we say that x is ε-close to S, and otherwise we say that x is ε-far from S. We denote the projection of $x \in \Sigma^n$ on $I \subseteq [n]$ by $x|_I$.

- We denote by $A^x(y)$ the output of algorithm A given direct access to input y and oracle access to string x. Given two interactive machines A and B, we denote by $(A^x, B(y))(z)$ the output of A when interacting with B, where A (respectively, B) is given oracle access to x

[1] That is, all Boolean functions that only depend on m of their k variables.

(respectively, direct access to y) and both parties have direct access to z. Throughout this work, probabilistic expressions that involve a randomized algorithm A are taken over the inner randomness of A (e.g., when we write $\Pr[A^x(y) = z]$, the probability is taken over the coin-tosses of A).

Integrality. Throughout this work, for simplicity of notation, we use the convention that all (relevant) integer parameters that are stated as real numbers are implicitly rounded to the closest integer.

Uniformity. To facilitate notation, throughout this work we define all algorithms *non-uniformly*; that is, we fix an integer $n \in \mathbb{N}$ and restrict the algorithms to inputs of length n. Despite fixing n, we view it as a generic parameter and allow ourselves to write asymptotic expressions such as $O(n)$. We remark that while our results are proved in terms of non-uniform algorithms, they can be extended to the uniform setting in a straightforward manner.

Circuit Size. We define the size $s(k)$ of a Boolean circuit $C : \{0,1\}^k \to \{0,1\}$ as the number of *gates* C contains. We count the input vertices of C as gates, and so $s(k) \geq k$. We shall write $f \in \mathsf{SIZE}\big(s(k)\big)$ to state that a Boolean function $f : \{0,1\}^k \to \{0,1\}$ can be computed by a Boolean circuit of size $s(k)$.

2.1 Locally Testable and Decodable Codes

Let $k, n \in \mathbb{N}$. A code over alphabet Σ with distance d is a function $C : \Sigma^k \to \Sigma^n$ that maps messages to codewords such that the distance between any two codewords is at least $d = d(n)$. If $d = \Omega(n)$, we say that C has linear distance. If $\Sigma = \{0,1\}$, we say that C is a binary code. If C is a linear map, we say that it is a linear code. The relative distance of C, denoted by $\delta(C)$, is d/n, and its rate is k/n. When it is clear from the context, we shall sometime abuse notation and refer to the code C as the set of all codewords $\{C(x)\}_{x \in \Sigma^k}$. Following the discussion in the introduction, we define locally testable codes and locally decodable codes as follows.

Definition 2.1 (Locally Testable Codes)**.** *A code $C : \Sigma^k \to \Sigma^n$ is a locally testable code (*LTC*) if there exists a probabilistic algorithm (tester) T that, given oracle access to $w \in \Sigma^n$ and direct access to proximity parameter ε, satisfies:*

1. *Completeness: For any codeword $w = C(x)$, it holds that $\Pr[T^{C(x)}(\varepsilon) = 1] \geq 2/3$.*
2. *Soundness: For any $w \in \{0,1\}^n$ that is ε-far from C, it holds that $\Pr[T^w(\varepsilon) = 0] \geq 2/3$.*

The query complexity of a LTC *is the number of queries made by its tester (as a function of ε and k). A* LTC *is said to have one-sided error if its tester satisfy perfect completeness (i.e., accepts valid codewords with probability 1).*

Definition 2.2 (Locally Decodable Codes)**.** *A code $C : \Sigma^k \to \Sigma^n$ is a locally decodable code (*LDC*) if there exists a constant $\delta_{\mathsf{radius}} \in (0, \delta(C)/2)$ and a probabilistic algorithm (decoder) D that, given oracle access to $w \in \Sigma^n$ and direct access to index $i \in [k]$, satisfies the following condition: For any $i \in [k]$ and $w \in \Sigma^n$ that is δ_{radius}-close to a codeword $C(x)$ it holds that $\Pr[D^w(i) = x_i] \geq 2/3$. The query complexity of a* LDC *is the number of queries made by its decoder.*

We shall also need the notion of relaxed-LDCs (introduced in [BSGH+06]). Similarly to LDCs, these codes have decoders that make few queries to an input in attempt to decode a given location in the message. However, unlike LDCs, the relaxed decoders are allowed to output a special symbol that indicates that the decoder detected a corruption in the codeword and is unable to decode this location. Note that the decoder must still avoid errors (with high probability).[2]

Definition 2.3 (relaxed-LDC)**.** *A code $C : \Sigma^k \to \Sigma^n$ is a* relaxed-LDC *if there exists a constant $\delta_{\mathsf{radius}} \in (0, \delta(C)/2)$,*

1. *(Perfect) Completeness: For any $i \in [k]$ and $x \in \Sigma^k$ it holds that $D^{C(x)}(i) = x_i$.*
2. *Relaxed Soundness: For any $i \in [k]$ and any $w \in \Sigma^n$ that is δ_{radius}-close to a (unique) codeword $C(x)$, it holds that*
$$\Pr[D^w(i) \in \{x_i, \perp\}] \geq 2/3.$$

There are a couple of efficient constructions of codes that are both relaxed-LDCs and LTCs (see [BSGH+06, GGK15]). We shall need the construction in [GGK15], which has the best parameters for our setting.[3]

Theorem 2.4 (e.g., [GGK15, Theorem 1.1])**.** *For every $k \in \mathbb{N}$ and $\alpha > 0$ there exists a (linear) code $C : \{0,1\}^k \to \{0,1\}^{k^{1+\alpha}}$ with linear distance, which is both a* relaxed-LDC *and a (one-sided error)* LTC *with query complexity* $\mathsf{poly}(1/\varepsilon)$*.*

3 The Definition of Universal Locally Testable Codes

Following the discussion in the introduction, we define universal locally testable codes as follows.

Definition 3.1. *Let $k, M \in \mathbb{N}$, and $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i \in [M]}$ be a family of functions. A universal locally testable code (*universal-LTC*) for $\mathcal{F}$ with query complexity $q = q(k, \varepsilon)$ is a code $C : \{0,1\}^k \to \{0,1\}^n$ such that for every $i \in [M]$ and $\varepsilon > 0$ there exists an ε-tester for the subcode $\Pi_i := \{C(x) : f_i(x) = 1\}$ with query complexity q. A* universal-LTC *is said to have one-sided error if all of its testers satisfy perfect completeness.*

Notation (ε-testing). We shall refer to a universal-LTC with respect to a specific proximity parameter $\varepsilon > 0$ as a universal-LTC$_\varepsilon$.

Organization. We start, in Section 4, by showing a canonical universal-LTC for every family of functions. This construction relies on a PCP-based machinery for asserting consistency of encodings, which we shall use throughout this work. Next, in Section 5, we prove general lower bounds on the query complexity of universal-LTCs as a function of the code's length and number of functions it can

[2] The full definition of relaxed-LDCs, as defined in [BSGH+06] includes an additional condition on the success rate of the decoder. Namely, for every $w \in \{0,1\}^n$ that is δ_{radius}-close to a codeword $C(x)$, and for at least a ρ fraction of the indices $i \in [k]$, with probability at least $2/3$ the decoder D outputs the i^{th} bit of x. That is, there exists a set $I_w \subseteq [k]$ of size at least ρk such that for every $i \in I_w$ it holds that $\Pr[D^w(i) = x_i] \geq 2/3$. We omit this condition since it is irrelevant to our application, and remark that every relaxed-LDC that satisfies the first two conditions can also be modified to satisfy the third conditions (see [BSGH+06, Lemmas 4.9 and 4.10]).

[3] Specifically, the codes in [GGK15] are meaningful for every value of the proximity parameter, whereas the codes in [BSGH+06] require $\varepsilon > 1/\mathsf{polylog}(k)$.

test. Finally, in Section 6, we show a specific family of functions (namely, the family of m-juntas, i.e., Boolean functions that only depend on m of their variables) for which we can obtain a smooth tradeoff between the universal-LTC length and locality.

4 The Canonical Universal-LTC

In this subsection we show a methodology for constructing an $O(1)$-local universal-LTC for any family of Boolean functions.

Theorem 4.1. *Let $t(k)$ be a proper complexity function, and let $\mathcal{F} = \{f_i : \{0,1\}^k \to \{0,1\}\}_{i\in[M]}$ be a family of functions such that for every $i \in [M]$, the function f_i can be computed by a size $t(k)$ circuit (i.e., $f_i \in \mathsf{SIZE}(t(k))$). Fix $n = M \cdot \widetilde{O}(t(k))$. Then, for every $\varepsilon > 1/\mathsf{polylog}(n)$ there exists a (one-sided error) $\mathsf{universal\text{-}LTC}_\varepsilon$ $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n)}$ for $\mathcal{F}$ with linear distance and query complexity $O(1/\varepsilon)$.*

We remark that, loosely speaking, the "canonical" universal-LTC above tightly matches the lower bound (see Theorem 5.1) in the low query complexity regime, for a reasonable setting of the parameters; see Section 5 for a more accurate statement.

The key idea for proving Theorem 4.1 is to design a universal-LTC that includes, for every $f \in \mathcal{F}$, a PCP encoding of the message x, which asserts the value of $f(x)$; this way we obtain a local test for each function in $\mathcal{F}$, simply by running its corresponding PCP verifier. The main problem, however, is that given concatenated PCP oracles we cannot locally verify that all of these PCPs are consistent with the exact same message. To overcome this issue, we shall first show a machinery for "bundling" encodings together in a way that allows for locally testing that all of the encodings are consistent with the same message. The key components for this construction are PCPs of proximity, which we discuss below.

4.1 Preliminaries: PCP of proximity

PCPs of proximity (PCPPs) [BSGH+06, DR06] are a variant of PCP proof systems, which can be thought of as the PCP analogue of *property testing*. Recall that a standard PCP is given explicit access to a statement and oracle access to a proof. The PCP verifier is required to probabilistically verify whether the (explicitly given) statement is correct, by making few queries to proof. In contrast, a PCPP is given oracle access to a statement and a proof, and is only allowed to make a small number of queries to both the statement and the proof. Since a PCPP verifier only sees a small part of the statement, it cannot be expected to verify the statement precisely. Instead, it is required to only accept correct statements and reject statements that are far from being correct (i.e., far in Hamming distance from any valid statement). More precisely, PCPs of proximity are defined as follows.

Definition 4.2. *Let V be a probabilistic algorithm (verifier) that is given* explicit *access to a proximity parameter $\varepsilon > 0$,* oracle *access to an* input *$x \in \{0,1\}^k$ and to a* proof *$\xi \in \{0,1\}^n$. We say that V is a* PCPP *verifier for language L if it satisfies the following conditions:*

- *Completeness: If $x \in L$, there exists a proof ξ such that the verifier always accepts the pair (x, ξ); i.e., $V^{x,\xi}(\varepsilon) = 1$.*

- *Soundness: If x is ε-far from L, then for every ξ the verifier rejects the pair (x, ξ) with high probability; that is,* $\Pr[V^{x,\xi}(\varepsilon) = 0] \geq 2/3$.

The length of the PCPP *is n and the query complexity is the number of queries made by V to* both x *and* ξ.

We shall use the following PCPP due to Ben-Sasson and Sudan [BS05] and Dinur [Din07].

Theorem 4.3 (Short PCPPs for $\mathcal{NP}$)**.** *For every $L \subseteq \{0,1\}^k$ that can be computed by a circuit of size $t(k)$, there exists a* PCPP *with query complexity $q = O(1/\varepsilon)$ and length $t(k) \cdot \mathsf{polylog}(t(k))$.*

4.2 Consistency-Testable Bundles

Building on techniques of Ben-Sasson at el. [BSGH+06], we show a way to bundle together (possibly partial) encodings of the same message such that it possible to locally test that all these encodings are indeed consistent. That is, we are given some encodings $E_1, \ldots, E_s : \{0,1\}^k \to \{0,1\}^n$, and we wish to encode a *single* message $x \in \{0,1\}^k$ by all of these encodings (i.e., to bundle $E_1(x), \ldots, E_s(x)$) such that we can test that all of the encodings are valid and consistent with the same message x. We shall need such bundles twice in this work: In Section 4.3 each E_i will simply correspond to a Boolean function $f_i \in \mathcal{F}$, and in Section 6 the E_i's will correspond to encodings of small chunks x.

The main idea is to construct a bundle that consists of three parts: (1) the (explicit) message x, (2) the encodings $E_1(x), \ldots, E_s(x)$, and (3) PCPPs that asset the consistency of the first part (the message) with each purported encoding $E_i(x)$ in the second part. However, such PCPPs can only ascertain that each purported pair of message and encoding, denoted (y, z_i), is *close* to a valid pair $(x, E_i(x))$. Thus, in this way we can only verify that the bundle consists of encodings of pairwise-close messages, rather than being close to encodings of a single message (e.g., the PCPPs may not reject a bundle $(x, E_1(y_1), \ldots, E_s(y_s))$ wherein each y_i is close to x).

To avoid this problem, we also encode the message via an error-correcting code ECC, so the bundle is of the form $\big(\mathsf{ECC}(x), (E_1(x), \ldots, E_s(x)), (\mathsf{PCPP}_1(x), \ldots, \mathsf{PCPP}_s(x))\big)$. Now, each PCPP ascertains that a purported pair (y, z_i) is close to $(\mathsf{ECC}(x), E_i(x))$. Due to the distance of ECC, this allows to verify that the bundle consists of s (close to valid) encodings of the *same* message. Lastly, we repeat $\mathsf{ECC}(x)$ such that it constitutes most of the bundle's length, and so if an alleged bundle is far from valid, its copies of $\mathsf{ECC}(x)$ must be corrupted, and so the bundle itself constitutes an error-correcting code that is locally testable (by verifying at random one of the PCPPs in the bundle).

More precisely, consider the following way of bundling several encodings of the same message.

Construction 4.4 (Consistency-Testable Bundles)**.** *Let $E_1, \ldots, E_s : \{0,1\}^k \to \{0,1\}^n$ be encodings such that for every $i \in [s]$, the problem of (exactly) deciding whether $(x, y) \in \{0,1\}^{k+n}$ satisfies $y = E_i(x)$ can be computed by a size $t(k)$ circuit. The* consistency-testable bundle *of $\{E_i(x)\}_{i \in [s]}$ is the code $B(x) : \{0,1\}^k \to \{0,1\}^\ell$ that consists of the following ingredients.*

1. *An (arbitrary) code $\mathsf{ECC} : \{0,1\}^k \to \{0,1\}^{n'}$ with linear distance, which can be computed by a size $\tilde{O}(n')$ circuit, where $n' = \tilde{O}(k)$.*

2. *Encodings $E_1, \ldots, E_s$ (given by the application) that we wish to bundle.*

3. PCP *of proximity oracles* $\xi_1, \ldots, \xi_s$ *for the language*

$$L_i = \{(a,b) \ : \ \exists x \in \{0,1\}^k \text{ such that } a = \mathsf{ECC}(x)^{r_a} \text{ and } b = E_i(x)^{r_b}\}.$$

where and r_a, r_b *are set such that* $|a| \approx |b| = O(t(k))$.

Let $\varepsilon \geq 1/\mathsf{polylog}(s \cdot t(k))$. *Consider the bundle*

$$B(x) = \Big(\mathsf{ECC}(x)^r, \big(E_1(x), \ldots, E_s(x)\big), \big(\xi_1(x), \ldots, \xi_s(x)\big)\Big),$$

where the length of each PCPP *oracle* $\xi_i(x)$ *is* $\widetilde{O}(t(k))$,[4] *and where* r *is the minimal integer such that the first part of the bundle constitutes* $(1-\varepsilon/2)$ *fraction of the bundle's length (i.e.,* $|\mathsf{ECC}(x)|^r \geq (1-\varepsilon/2)\cdot \ell$*).*

Note that the length of B is $\ell = \widetilde{O}(s \cdot t(k))$ and that B has linear distance, because $|\mathsf{ECC}(x)|^r$ dominates B's length.

Notation for (alleged) bundled. For the analysis, when we consider an arbitrary string $w \in \{0,1\}^\ell$ (which we think of as an alleged bundle), we view $w \in \{0,1\}^{\ell_1+\ell_2+\ell_3}$ as a string composed of three parts (analogous to the three parts of Construction 4.4):

1. The anchor, $\widetilde{\mathsf{ECC}}(x) = (\widetilde{\mathsf{ECC}}(x)_1, \ldots, \widetilde{\mathsf{ECC}}(x)_r) \in \{0,1\}^{n' \cdot r}$, which consists of r alleged copies of $\mathsf{ECC}(x)$;
2. The bundled encodings $(\widetilde{E_1}(x), \ldots, \widetilde{E_s}(x)) \in \{0,1\}^{n \cdot s}$, which allegedly equals $(E_1(x), \ldots, E_s(x))$;
3. The PCPPs $(\widetilde{\xi_1}(x), \ldots, \widetilde{\xi_s}(x)) \in \{0,1\}^{\widetilde{O}(t(k)) \cdot s}$, which allegedly equals $(\xi_1(x), \ldots, \xi_s(x))$.

We show that there exists a local test that can ascertains the validity of the bundle as well as asserts the consistency of any encoding E_i in the bundle with the *anchor* of the bundle. Note that since the bundle's anchor dominates its length, it is possible that the bundle is very close to valid, and yet all of the E_i's are heavily corrupted. Thus, we also need to provide a test for the validity of each E_i and its consistency with the anchor.

Proposition 4.5. *For every bundle* $B(x)$, *as in Construction 4.4, there exists a consistency test* T *that for every* $\varepsilon \geq 1/\mathsf{polylog}(\ell)$ *makes* $O(1/\varepsilon)$ *queries to a string* $w \in \{0,1\}^\ell$ *and satisfies the following conditions.*

1. *If* $w = B(x)$, *then for every* $i \in \{0\} \cup [s]$ *it holds that* $\Pr[T^w(i) = 1] = 1$.
2. *If* w *is* ε*-far from* B, *then* $\Pr[T^w(0) = 0] \geq 2/3$.
3. *For every* $i \in [s]$, *if there exists* $x \in \{0,1\}^k$ *such that* w *is* ε*-close to* $B(x)$ *and* $\widetilde{E_i}(x)$ *is* ε*-far from* $E_i(x)$, *then* $\Pr[T^w(i) = 0] \geq 2/3$.

Note that $T^w(0)$ is a codeword test for B, whereas for every $i \in [s]$, the test $T^w(i)$ asserts that $\widetilde{E_i}$ is close to an encoding of the anchor. To verify that w is a bundle wherein all encodings refer to the same message (the anchor), we have to invoke $T^w(i)$ for all $i \in \{0\} \cup [s]$, but typically we will be interested only in the consistency of one encoding with the anchor, where this encoding is determined by the application.

[4] Note that $L_i \in \mathsf{SIZE}(m)$ by the hypothesis regarding ECC and E_i. Thus, by Theorem 4.3, such a PCPP exists.

Proof of Proposition 4.5. We show that for every bundle $B(x)$, as in Construction 4.4, there exists a consistency test T that, for every $\varepsilon \geq 1/\mathsf{polylog}(\ell)$, makes $O(1/\varepsilon)$ queries to a string $w \in \{0,1\}^\ell$ and satisfies the following conditions.

1. If $w = B(x)$, then for every $i \in \{0\} \cup [s]$ it holds that $\Pr_T[T^w(i) = 1] = 1$.
2. If w is ε-far from B, then $\Pr[T^w(0) = 0] \geq 2/3$.
3. For every $i \in [s]$, if there exists $x \in \{0,1\}^k$ such that w is ε-close to $B(x)$ and $\widetilde{E_i}$ is ε-far from $E_i(x)$, then $\Pr[T^w(i) = 0] \geq 2/3$.

Let $\varepsilon \geq 1/\mathsf{polylog}(\ell)$, and assume without loss of generality that $\varepsilon < \delta(\mathsf{ECC})/2$.[5] For every $i \in [s]$ denote by V_i the PCPP verifier for the language

$$L_i = \{(a,b) \ : \ \exists x \in \{0,1\}^k \text{ such that } a = \mathsf{ECC}(x)^{r_a} \text{ and } b = E_i(x)^{r_b}\},$$

with respect to proximity parameter $\varepsilon/6$ and soundness $9/10$. Consider the ε-tester T that is given $i \in \{0\} \cup [s]$ and oracle access to $w = (\widetilde{\mathsf{ECC}(x)}, (\widetilde{E_i})_{i\in[s]}, (\widetilde{\xi_i})_{i\in[s]}) \in \{0,1\}^\ell$ and accepts if both of the following tests accept.

1. Repetition Test: Query two random copies from the long-code part of w and check if they agree on a random location. More accurately, select uniformly at random $j, j' \in [r]$ and reject if and only if $\widetilde{\mathsf{ECC}}(x)_j$ and $\widetilde{\mathsf{ECC}}(x)_{j'}$ disagree on a *random* coordinate. Repeat this test $O(1/\varepsilon)$ times.
2. Consistency Test: Choose uniformly $j \in [r]$. If $i = 0$, choose uniformly $i' \in [s]$, otherwise set $i' = i$. Reject if the verifier $V_{i'}$ rejects on input $(\widetilde{\mathsf{ECC}}(x)_j^{r_a}, \widetilde{E_{i'}}(x)^{r_b})$ and proof $\widetilde{\xi_{i'}}(x)$.

The first condition of Proposition 4.5 follows by construction. For the other conditions, first observe that if $\widetilde{\mathsf{ECC}}(x)$ is far from consisting of r identical copies, then the repetition test rejects with high probability. That is, let $\hat{c} \in \{0,1\}^{n'}$ be a string that is closest on average to the copies in $\widetilde{\mathsf{ECC}}(x)$, i.e., a string that minimizes $\Delta(\widetilde{\mathsf{ECC}}(x), \hat{c}^r) = \sum_{j=1}^r \Delta(\widetilde{\mathsf{ECC}}(x)_j, \hat{c})$. Observe that

$$\mathop{\mathbf{E}}_{j,j' \in_R [r]}[\delta(\widetilde{\mathsf{ECC}}(x)_j, \widetilde{\mathsf{ECC}}(x)_{j'})] \geq \mathop{\mathbf{E}}_{j \in_R [r]}[\delta(\widetilde{\mathsf{ECC}}(x)_j, \widetilde{\mathsf{ECC}}(x))] = \delta(\widetilde{\mathsf{ECC}}(x), \hat{c}^r).$$

If $\delta(\widetilde{\mathsf{ECC}}(x), \hat{c}^r) > \varepsilon/60$, then by invoking the codeword repetition test $O(1/\varepsilon)$ times, with probability at least $2/3$ one of the invocations will reject. Otherwise, note that with probability at least $9/10$ the random copy $\widetilde{\mathsf{ECC}}(x)_j$ is $\varepsilon/6$-close to $\hat{c}$; assume hereafter that this is the case.

If w is ε-far from B, then since $\widetilde{\mathsf{ECC}}(x) \geq (1 - \varepsilon/2)\ell$, it follows that $\widetilde{\mathsf{ECC}}(x)$ is $\varepsilon/2$-far from ECC^r, and thus

$$\delta_{\mathsf{ECC}^r}(\hat{c}^r) \geq \delta_{\mathsf{ECC}^r}(\widetilde{\mathsf{ECC}}(x)) - \delta(\hat{c}^r, \widetilde{\mathsf{ECC}}(x)) = \varepsilon/2 - \varepsilon/60 > \varepsilon/3.$$

Recall that we assumed that $\delta(\widetilde{\mathsf{ECC}}(x)_j, \hat{c}) \leq \varepsilon/6$, and so $\delta_{\mathsf{ECC}}(\widetilde{\mathsf{ECC}}(x)_j) > \varepsilon/6$. Thus, $\Pr[V_{i'}^w = 0] \geq 9/10 \cdot 9/10$.

Finally, If there exists $x \in \{0,1\}^k$ such that w is ε-close to $B(x)$ and $\widetilde{E_i}(x)$ is ε-far from $E_i(x)$, then since $\delta(\widetilde{\mathsf{ECC}}(x), \hat{c}^r) \leq \varepsilon/60$, it follows that with probability at least $9/10$ the random copy $\widetilde{\mathsf{ECC}}(x)_j$ is $\varepsilon/6$-close to $\mathsf{ECC}(x)$. Hence, $(\widetilde{\mathsf{ECC}}(x)_j^{r_a}, \widetilde{E_i}(x)^{r_b})$ is at least $5\varepsilon/6$-far from L_i, and so $\Pr[V_i^w = 0] \geq 9/10 \cdot 9/10$. □

[5] The relative distance of ECC is constant, so if $\varepsilon \geq \delta(\mathsf{ECC})/2$, we can set the proximity parameter to $\delta(\mathsf{ECC})/2$, increasing the complexity by only a constant factor.

4.3 Proof of Theorem 4.1

Let $\mathcal{F} = \{f_i : \{0,1\}^k \to \{0,1\}\}_{i\in[M]}$ be a family of functions such that for every $i \in [M]$ it holds that $f_i \in \mathsf{SIZE}(t(k))$. Fix $n = M \cdot \widetilde{O}(t(k))$ and $\varepsilon > 1/\mathsf{polylog}(n)$. We set $E_i = f_i$ for every $i \in [M]$, bundle these encodings via Proposition 4.5, and denote the bundle by $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n)}$. Note that by Proposition 4.5, the code C has linear distance.

Fixing $f_i \in \mathcal{F}$, we show an $O(1/\varepsilon)$-local ε-tester T_i for the subcode $\Pi_i := \{C(x) : f_i(x) = 1\}$. Given input $w \in \{0,1\}^{\widetilde{O}(n)}$, the tester T_i simply invokes the bundle consistency test on w (which makes $O(1/\varepsilon)$ queries to w), with respect to proximity parameter ε and the purported copy of $f_i(x)$ in the bundle, which is a bit, denoted by z_i. The tester accepts if and only if the consistency test accepts and $z_i = 1$.

The perfect completeness of T_i follows by the one-sided error of the bundle consistency test. For the soundness, assume that w is ε-far from Π_i. By Proposition 4.5, we can assume that there exists $y \in \{0,1\}^k$ such that w is ε-close to $C(y)$ (otherwise the consistency test fails with probability $2/3$), and since w is ε-far from Π_i, it holds that $f_i(y) = 0$; furthermore, the value of w at f_i is uncorrupted (i.e., it actually equals 0),[6] and so T_i rejects.

5 General Lower Bounds

In this section we prove a general lower bound on the query complexity of universal-LTCs for *any* family of functions $\mathcal{F}$, as a function of the universal-LTC's length and the number of functions in $\mathcal{F}$. We also prove a stronger lower bound for the case that the functions in $\mathcal{F}$ are "pairwise far".

Theorem 5.1. *Let $\mathcal{F} = \{f_i : \{0,1\}^k \to \{0,1\}\}_{i\in[M]}$ be a family of distinct functions. Then, every q-local* universal-LTC$_\varepsilon$ *$C : \{0,1\}^k \to \{0,1\}^n$ for $\mathcal{F}$ with linear distance and $\varepsilon < \delta(C)/2$ must satisfy*

$$q \geq \log\log M - \log\log n - \log(k) - O(1).$$

Furthermore, if there exists $\beta = \Omega(1)$ such that $\Pr_{x\in\{0,1\}^k}[f_i(x) \neq f_j(x)] > \beta$ for every $i \neq j$, then $q = \Omega(\log\log M - \log\log n)$.

Note that in the constant locality regime (i.e., where $q = O(1)$), the lower bound for "pairwise far" functions implies that $n \geq M^c$ for some constant $c > 0$. On the other hand, recall that the canonical universal-LTC in Theorem 4.1 has query complexity $O(1)$ and length $\widetilde{O}(M \cdot t(k))$, for any family of functions that can be computed by a circuit of size $t(k)$ each (recall that $t(k) \geq k$, by definition). Thus, for sufficiently large families of "pairwise far" functions, the lower bound above matches the upper bound of the canonical universal-LTC up to a constant power, where by "sufficiently large" we mean that $t(k) = \mathsf{poly}(M)$.

Proof. We prove Theorem 5.1 using two different representations of testers: when proving the main claim we view testers as *randomized* decision trees, whereas in the proof of the furthermore claim we view testers as a distribution over *deterministic* decision trees. We begin with the main claim, for which we use the following lemma, due to Goldreich and Sheffet [GS10], which shows that the amount of randomness that suffices for testing is roughly doubly logarithmic in the size of the universe of objects it tests.

[6] Formally, Proposition 4.5 guaranties that w contains a copy of $f_i(y)$ that is ε-close to z_i, but since $f_i(y)$ is a single bit, this means that $f_i(y)$ is uncorrupted.

Lemma 5.2 ([GS10, Lemma 3.7] restated)**.** *Let $k \in \mathbb{N}$, $U \subseteq \{0,1\}^k$, and let $\Pi \subseteq U$ be a property. Assume that Π has a tester with randomness complexity r, which makes q queries to a string in U. Then, Π has a tester that makes q queries and has randomness complexity $\log\log|U| + O(1)$.*

Let $C : \{0,1\}^k \to \{0,1\}^n$ be a universal-LTC for $\mathcal{F}$, and assume that each tester T_i for the subcode $\Pi_i = \{C(x) : f_i(x) = 1\}$ is given the *promise* that its input is a valid codeword of C; that is, we only consider the behavior of T_i given a codeword $C(x)$ out of the universe $U \coloneqq \{C(x) : x \in \{0,1\}^k\}$, which consists of 2^k codewords. We shall prove a lower bound of the query complexity of the foregoing testers, and this, in particular, implies a lower bound on standard testers (which are *not* given a promise regarding their input).

Here we view a randomized decision tree is a decision tree wherein the vertices are also allowed to be labeled with a special coin-flip symbol $*$ that indicates that during computation, one of the children of each $*$-labeled vertex is chosen uniformly at random. Note that any tester with query complexity q and randomness complexity r can be represented by a randomized decision tree of depth $q + r$ in which all vertices in the first r layers are $*$-labeled. By Lemma 5.2 we can assume without loss of generality that $r = \log\log|U| + O(1) = \log(k) + O(1)$. Observe that there are at most $(n+3)^{2^{q+\log(k)+O(1)}}$ such randomized decision trees (we bound the number of depth d decision trees over n variables by counting all possible labeling of a depth d binary tree with the names of the variables, the two terminals, and the coin-flip symbol).

Recall that for every $i \neq j$ the functions f_i and f_j are different, hence there exist $x \in \{0,1\}^k$ such that $C(x) \in \Pi_i \triangle \Pi_j$, and so by the distance of C, a tester for Π_i cannot also be a tester for Π_j. Therefore $M \leq (n+3)^{2^{q+\log(k)+O(1)}}$, and so $q \geq \log\log M - \log\log n - \log k - O(1)$.

For the furthermore claim of Theorem 5.1, for every $i \in [M]$, denote by T_i the q-query ε-tester for the subcode $\Pi_i \coloneqq \{C(x) : f_i(x) = 1\}$, and by amplification, assume that T_i makes $q' = O(q)$ queries and obtains completeness and soundness error of at most $\delta_{\mathsf{err}} = \beta/2$. Note that if x satisfies $f_i(x) = 1$, then $C(x) \in \Pi_i$, thus the tester T_i accepts (i.e., outputs 1) with high probability, and if x satisfies $f_i(x) = 0$, then $C(x)$ is ε-far from Π_i, and thus the tester T_i rejects (i.e., outputs 0) with high probability; that is,

$$\forall x \in \{0,1\}^k \quad \Pr[T_i^{C(x)} = f_i(x)] \geq 1 - \delta_{\mathsf{err}}. \tag{5.1}$$

Hence, testing codewords of C for membership in Π_i amounts to *computing* $f_i(x)$.

Let $D_1, \ldots, D_s$ be all (binary, deterministic) depth q' decision trees over n variables, and note that $s \leq (n+2)^{2^{q'}}$. Here we view each T_i is a distribution over $\{D_j\}_{j\in[s]}$; that is, for every $i \in [M]$ there exists a distribution μ_i over $[M]$ such that for every $w \in \{0,1\}^n$, the output of T_i^w is obtained by drawing $j \sim \mu_i$ and outputting D_j^w. By Eq. (5.1), for every x and $i \in [M]$,

$$\sum_{j=1}^{s} \mu_i(j) \cdot \Pr_{x\in\{0,1\}^k}[D_j^{C(x)} = f_i(x)] \geq 1 - \delta_{\mathsf{err}}.$$

In particular, we obtain that for every $i \in [M]$ there exists $j \in [s]$ such that $\Pr_x[D_j^{C(x)} = f_i(x)] \geq 1 - \delta_{\mathsf{err}}$. Observe that if $M > s$ (i.e., there are more f_i's than depth-q' decision trees), then there exists $i_1, i_2 \in [M]$, where $i_1 \neq i_2$ and $j \in [s]$, such that $\Pr_x[f_{i_1}(x) = D_j^{C(x)} = f_{i_2}(x)] \geq 1 - 2\delta_{\mathsf{err}} = 1 - \beta$, in contradiction to the hypothesis. Thus $M \leq s \leq (n+1)^{2^{q'}}$, and since $q' = O(q)$, then $q = \Omega(\log\log M - \log\log n)$. □

On the gap between "pairwise far" and general families of functions. Recall that there is an additive difference of $\Omega(\log k)$ between the lower bound for general families of functions and the stronger lower bound for families of functions that are "pairwise far". We leave open the question of whether the lower bound for general families of functions can be improved to match the stronger lower bound for "pairwise far" functions, or whether there exists a universal-LTC for a family of functions, which are *not* "pairwise far", that matches the lower bound for general functions. We point out two observations regarding the forgoing question:

1. The argument in the furthermore claim of Theorem 5.1 also shows that any universal-LTC with *deterministic* testers must satisfy $q = \Omega(\log\log M - \log\log n)$, even for families of functions that are not "pairwise far" and when given the proviso that the input is a valid codeword. Therefore, to construct a universal-LTC that matches the general lower bound, the testers must use a randomized strategy, not only for checking the validity of the encoding, but also for computing the function of the message. (We remark that all of the universal-LTCs in this work use randomness only for codeword testing.)

2. The proof of the furthermore claim of Theorem 5.1 actually yields a stronger statement regarding "pairwise far" functions. Specifically, it only requires that the functions should be "pairwise far" under *some* distribution (and not necessarily the uniform distribution); that is, it suffices that there exists a distribution $\mathcal{D}$ over $\{0,1\}^k$ such that $\Pr_{x\sim\mathcal{D}}[f_i(x) \neq f_j(x)] = \Omega(1)$ for every $i \neq j$.

6 Trading off Length for Locality

The general lower bound in Theorem 5.1 allows for a tradeoff between the universal-LTC's length and locality. We remark that while the canonical universal-LTC in Theorem 4.1 matches this lower bound, it is limited to the extreme end of the tradeoff, wherein the locality is minimized (i.e., the query complexity is constant). In this subsection we show a specific family of functions (namely, the family of m-juntas) for which we can obtain a smooth tradeoff between the universal-LTC's length and locality.

6.1 Universal-LTCs of Nearly-Linear Length

Let $m, k \in \mathbb{N}$ such that $m \leq k$, and denote by $\mathsf{Junta}_{m,k}$ the set of all $\binom{k}{m} \cdot 2^{2^m}$ k-variate Boolean functions that only depend on m coordinates. We start by showing that using super-constant query complexity, we can obtain universal-LTCs that are shorter than the canonical universal-LTC. More precisely, we prove that there exists a universal-LTC for $\mathsf{Junta}_{m,k}$ with linear distance, *nearly-linear length*, and query complexity that is quasilinear in m. (We discuss how Observation 6.1 matches the lower bound in Theorem 5.1 in Section 6.3.)

Observation 6.1. *Let $k, m \in \mathbb{N}$ such $m \leq k$, and let $\alpha > 0$ be a constant. For every $\varepsilon > 0$ there exists a (one-sided error)* universal-LTC$_\varepsilon$ $C : \{0,1\}^k \to \{0,1\}^{k^{1+\alpha}}$ *for* $\mathsf{Junta}_{m,k}$ *with linear distance and query complexity* $\widetilde{O}(m) + \mathsf{poly}(1/\varepsilon)$.

Sketch of proof. The idea is to use a code C that is both locally testable and decodable, and obtain a tester for each subcode $\{C(x) : f(x) = 1\}$ (where $f \in \mathsf{Junta}_{m,k}$) by invoking the tester for membership in C, using the decoder to recover the values of the m influencing variables of f (for

which we shall need to reduce the error probability of the decoder to $1/m$), and ruling accordingly. Recall, however, that there are no known LDCs with constant query complexity and polynomial length (let alone such with nearly-linear length). Instead, we observe that for the foregoing idea it suffices that C is a relaxed-LDC,[7] and so we can use the code in Theorem 2.4, which is both a (one-sided error) LTC and a relaxed-LDC, with nearly-linear length. The implementation of the aforementioned ideas is straightforward, and so, we omit it.

□

Digression: Universal-LTCs with optimal rate. In Observation 6.1, we are concerned with minimizing the locality of the universal-LTCs, while settling for nearly-linear length (and so, we use the code in Theorem 2.4 as the base code). We remark that the argument underlying Observation 6.1 holds for *any* base code that is both locally testable and (possibly relaxed) locally decodable. Thus, different base codes may be used to obtain universal-LTCs in other regimes. For example, allowing large query complexity (which depends on k) and focusing on optimizing the rate and the distance, we can obtain the following corollary by using the recent construction, due to Meir [Mei14, Theorem 1.1, 1.2, and Remark 1.5], of codes that are both locally testable and decodable with constant rate and optimal distance, and query complexity that is an arbitrary small power of the input length.

Corollary 6.2. *For every $0 < r < 1$, $\alpha, \beta > 0$ there exists a finite field $\mathbb{H}$ of characteristic 2 such that for every $m \leq k$, there exists a* universal-LTC $C : \mathbb{F}_2^k \to \mathbb{H}^n$ *for* $\mathsf{Junta}_{m,k}$ *with rate at least r, relative distance at least $1 - r - \alpha$, and query complexity $O(k^\beta m \log m + k^\beta/\varepsilon)$.*[8]

6.2 The Actual Tradeoff

Next, we show a universal-LTC for $\mathsf{Junta}_{m,k}$ with a smooth tradeoff between length and query complexity.

Proposition 6.3. *Let $k, m \in \mathbb{N}$ such that $m \leq k$. For every $\tau < m$ and $\varepsilon \geq 1/\mathsf{polylog}(n)$, where $n \leq \frac{k^{m+1}}{k^t} \cdot (2^{2^m})^{1/2^t}$, there exists exists a (one-sided error)* universal-LTC$_\varepsilon$ $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n)}$ *for* $\mathsf{Junta}_{m,k}$ *with linear distance and query complexity $\widetilde{O}(\tau) + O(1/\varepsilon)$.*

We remark that in the $\widetilde{O}(m)$-locality regime (the query-heavy extreme of the tradeoff), Proposition 6.3 only yields a universal-LTC of quadratic length, whereas Observation 6.1 achieves nearly-linear length.[9]

Sketch of proof. The basic idea is to map $x \in \{0,1\}^k$ to the long code encoding of the *projection* of x to each m-subset of $[k]$; that is, $x \to \big(\mathsf{LC}(x|_{S_1}), \ldots, \mathsf{LC}(x|_{S_N})\big)$, where $S_1, \ldots, S_N$ are all $N = \binom{m}{k}$ distinct m-subsets of $[k]$ and $\mathsf{LC} : \{0,1\}^m \to \{0,1\}^{2^{2^m}}$ is the corresponding long code.

Next, to ascertain that all the long code encodings are consistent with restrictions of a single x, we bundle these encodings with PCPs according to the consistency-testable bundling mechanism presented in Section 4 (where the encodings $\{E_i\}$ correspond to $\{\mathsf{LC}(x|_{S_i})\}$). This yields a universal-LTC

[7] Recall that relaxed-LDCs are a relaxation of locally decodable codes that requires local recovery of individual information-bits, yet allow for recovery-failure, but not error, on the rest (see Definition 2.3).

[8] Recall that the query complexity measures the number of queries made, rather than the number of bits that were read, but since p is a constant, the difference is immaterial.

[9] It is possible to optimize Proposition 6.3 such that in the query-heavy extreme of the tradeoff it will yield universal-LTCs of linear length, by adapting techniques from [BSGH+06, Section 4] to our setting. However, this methodology is far more involved than simply using Observation 6.1 in the $\widetilde{O}(m)$-locality regime.

for m-juntas with query complexity $O(1)$ and length $\binom{k}{m} \cdot \tilde{O}(2^{2^m} + k)$: To test that x satisfies the junta $f(x) = f'(x|_S)$, where $S \subseteq [k]$ such that $|S| = m$, we first use Proposition 4.5 to ensure the consistency of the bundle (i.e., the consistency of f with the anchor), then we extract the value of $f(x)$ by locally correcting the point that corresponds to f' in the purported copy of $\mathsf{LC}(x|_S)$.

Finally, to obtain a smooth tradeoff, we modify the foregoing construction such that x is mapped to the long code encoding of the projection of x to each $(m-\tau)$-subset of $[k]$ (instead of m-subset), for the given parameter $\tau \in [m]$. The idea is that now, to test that x satisfies $f'(x|_S) = 1$, we first arbitrarily choose t bits of $x|_S$ and decode them one-by-one (as in Observation 6.1); this induces a function f'' on the remaining $m - \tau$ bits, which we compute by self-correcting the single bit that corresponds to f'' in the long code encoding of x projected to these $m - \tau$ bits. The implementation of the foregoing ideas is straightforward and is presented in Appendix B. □

6.3 Lower Bounds for Universal-LTCs for Juntas

We conclude this subsection by proving a lower bound on the query complexity of universal-LTCs for $\mathsf{Junta}_{m,k}$. Observe that the family of all m-juntas do *not* satisfy the "pairwise far" condition, and thus Theorem 5.1 only gives us a lower bound of $q \geq m - \log\log n - O(\log k)$. However, we show that while the family $\mathsf{Junta}_{m,k}$ is not "pairwise far", it contains a dense subset of functions that are "pairwise far", and so we can strengthen the foregoing lower bound as follows.

Proposition 6.4. *Let $k, m \in \mathbb{N}$ such $m \leq k$. There exists a universal constant $c > 0$ such that every $\mathsf{universal\text{-}LTC}_\varepsilon$ $C : \{0,1\}^k \to \{0,1\}^n$ for $\mathsf{Junta}_{m,k}$ with linear distance and $\varepsilon < \delta(C)/2$ must have query complexity $\Omega(m - \log\log(n) + c)$.*

We remark that that for $m = \big(1 + \Omega(1)\big) \cdot \log\log(n)$, the lower bound simplifies to $\Omega(m)$ and matches Observation 6.1 up to a constant power. Furthermore, it is possible to improve Proposition 6.4 such that it gives a non-trivial lower bound when $m < \log\log(k)$ (see discussion at the end of the section).

Proof of Proposition 6.4. We show that $\mathsf{Junta}_{m,k}$ contains a dense subset that is "pairwise far". Specifically, fix $S \subseteq [k]$ such that $|S| = m$, and let $\mathsf{Junta}^S_{m,k} \subseteq \mathsf{Junta}_{m,k}$ denote all m-juntas that depend only on coordinates in S. We prove that there exists a family $\mathcal{F} \subseteq \mathsf{Junta}^S_{m,k}$ of $M = 2^{\Omega(2^m)}$ distinct functions such that every distinct f and g in $\mathcal{F}$ satisfies $\Pr_{x \in \{0,1\}^k}[f(x) \neq g(x)] = \Omega(1)$.

Note that the set of truth tables, restricted to inputs supported on S, of all $f \in \mathsf{Junta}^S_{m,k}$ is isomorphic to $\{0,1\}^{2^m}$, and thus we can choose a subset of it that constitutes a good code. That is, for every $f \in \mathsf{Junta}^S_{m,k}$, note that $f(x) = f'(x|_S)$ for some $f' : \{0,1\}^m \to \{0,1\}$, and denote the truth table of f' by $\langle f' \rangle$. Let C_0 be a code with linear distance, constant rate, and codewords of length 2^m, and observe that by the rate and distance of the code C_0, the set $\mathcal{F} = \{f \in \mathsf{Junta}^S_{m,k} \ : \ \langle f' \rangle \in C_0\}$ is a collection of $2^{\Omega(2^m)}$ functions such that every distinct $f, g \in \mathcal{F}$ satisfy

$$\Pr_{x \in \{0,1\}^k}[f(x) \neq g(x)] = \Pr_{\substack{x \in \{0,1\}^k \\ x|_{[k] \setminus S} = \mathbf{0}}}[f(x) \neq g(x)] = \Omega(1).$$

The proof of Proposition 6.4 is concluded by applying Theorem 5.1 to $\mathcal{F}$. □

Improving the lower bound. We point out a slackness in the proof of Proposition 6.4. Specifically, we apply Theorem 5.1 to a subset $\mathcal{F}$ of m-juntas that depend on a *single* set $S \subset [k]$ of cardinality m, and so we lose all dependency in k (the dimension of the code). We sketch below how to tighten this slackness and obtain a slightly stronger lower bound of $\Omega(\max\{m, \Omega(\log(m)) + \log\log(k)\} - \log\log(n))$, which gives a non-trivial lower bound also when $m < \log\log(k)$ and $n < k^m$ (while noting that Proposition 6.4 trivializes for this range of parameters).

As a first attempt, we can consider a *partition* of $[k]$ to sets $S_1, \ldots, S_{k/m}$ of cardinality m, and (similarly to Proposition 6.4) include in $\mathcal{F}$ a subset of functions from each $\mathsf{Junta}_{m,k}^{S_i}$ whose truth-tables form a good code. Inspection shows that as long as the foregoing good code is *balanced*,[10] juntas in such $\mathcal{F}$ are pairwise far, and so we can apply Theorem 5.1. The problem is, however, that such argument only strengthens the lower bound by a constant factor; that is, it yields $q = \Omega\left(\log\log\left(\frac{k}{m} \cdot 2^{2^m}\right) - \log\log n\right)$, which is not asymptotically better than $q = \Omega(\log\log(2^{2^m}) - \log\log n)$, established in Proposition 6.4.

To obtain an asymptotical strengthening, we can choose $k^{\Omega(m)}$ distinct subsets of $[k]$ with small (say, $m/100$) pairwise intersection (using the Nisan-Wigderson combinatorial designs [NW94]), and for each such subset S, include in $\mathcal{F}$ juntas from $\mathsf{Junta}_{m,k}^{S}$ whose truth-tables form a *random* code. On inspection, it turns out that juntas in such $\mathcal{F}$ are pairwise far, and thus we can apply Theorem 5.1 to obtain $q = \Omega(\log\log(k^m \cdot 2^{2^m}) - \log\log n)$, which yields the aforementioned bound.

References

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998. 1.1

[BGS98] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, pcps, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998. 1.1

[BS05] Eli Ben-Sasson and Madhu Sudan. Simple PCPs with poly-log rate and query complexity. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 266–275. ACM, 2005. 4.1

[BSGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006. 1.1, 2.1, 2.1, 2, 3, 4.1, 4.2, 9, A, A

[Din07] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12, 2007. 4.1

[DR06] Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006. 4.1

[FS95] Katalin Friedl and Madhu Sudan. Some improvements to total degree tests. In *Theory of Computing and Systems, 1995. Proceedings., Third Israel Symposium on the*, pages 190–198. IEEE, 1995. 1

[10] That is, a code wherein each codeword consists of an equal number of 0's and 1's.

[GG16a] Oded Goldreich and Tom Gur. Universal locally testable codes (original version). *Electronic Colloquium on Computational Complexity (ECCC)*, 23:42, 2016. ∗, 1.3

[GG16b] Oded Goldreich and Tom Gur. Universal locally verifiable codes. *Electronic Colloquium on Computational Complexity (ECCC TR16-192)*, 2016. ∗, 1.3

[GGK15] Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 1–41, 2015. 2.1, 2.4, 3

[GS06] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *J. ACM*, 53(4):558–655, 2006. 1

[GS10] Oded Goldreich and Or Sheffet. On the randomness complexity of property testing. *Computational Complexity*, 19(1):99–133, 2010. 5, 5.2

[Mei14] Or Meir. Locally correctable and testable codes approaching the singleton bound. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:107, 2014. 6.1

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994. 6.3

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. 1

A On Obtaining Locally Decodable Codes from Universal-LTCs

In this appendix we show that universal-LTCs for the family of linear functions (and more generally, for self-correctable families of functions) imply local decodability in the strong (non relaxed) sense. More accurately, denote the set of all k-variate linear functions over $\mathsf{GF}(2)$ by Linear_k. The following theorem shows that any universal-LTC for Linear_k implies a LDC with roughly the same parameters.

Theorem A.1. *If there exists an* universal-LTC C *for* Linear_k *with linear distance, rate* r*, and query complexity* $q = q(\varepsilon)$*, then there exists a binary* LDC *with linear distance, rate* $\Omega(r)$*, and query complexity* $O(1)$*.*

Proof. Fix $\varepsilon = \delta(C)/3$. For every linear function $f \in \mathsf{Linear}_k$ and $b \in \{0,1\}$, let $T_{f,b}$ be the ε-tester for the subcode $\Pi_{f,b} \coloneqq \{C(x) \,:\, f(x) = b\}$ guarantied by the universal-LTC $C : \{0,1\}^k \to \{0,1\}^n$. These testers admit a natural candidate for a local decoding procedure: to decode x_i, simply invoke $T_{f,0}$ and $T_{f,1}$ for the linear function $f(x) = x_i$, and rule according to the tester that accepted.

The problem is that given a slightly corrupted copy of $C(x)$, the testers $T_{f,0}$ and $T_{f,1}$ may both reject, since they are not necessarily *tolerant*;[11] in this case we cannot decode. (Indeed, if the aforementioned testers are tolerant, then the foregoing procedure is a local decoder.[12]) Nevertheless, since the foregoing case only happens when the input is *not* a valid codeword, we obtain a procedure

[11] Recall that tolerant testers accept strings that are (say) $\delta(C)/3$-close to being valid and reject strings that are (say) $\delta(C)/2$-far from being valid (with high probability).

[12] In fact, the argument above shows that a tolerant universal-LTC for *any* family of functions $\mathcal{F}$ that contain the dictator functions, i.e., such that $\{f(x) = x_i\}_{i\in[k]} \subseteq \mathcal{F}$, implies a LDC with roughly the same parameters.

that either decodes correctly or detects a corruption in the encoding and aborts (similarly to relaxed-LDCs, see Definition 2.3). Then, by slightly modifying the code, we can bound the number of linear functions on which we are forced to abort and use the linear functions that we are able to compute to recover *any* linear function, including $f(x) = x_i$. Details follow.

Assume without loss of generality that the testers of the universal-LTC have soundness error of at most $1/10$. Consider the algorithm $\mathcal{A}$ that, given $f \in \mathsf{Linear}_k$ and oracle access to $w \in \{0,1\}^n$, invokes $T_{f,0}$ and $T_{f,1}$ on w; if one tester accepted and the other rejected, $\mathcal{A}$ rules according to the accepting tester, and otherwise it outputs $\perp$. Hence, $\mathcal{A}$ has query complexity $O(q(\varepsilon)) = O(1)$. The following claim shows that indeed $\mathcal{A}$ succeeds in locally computing $f(x)$ in the following sense (which is analogous to that of relaxed-LDCs).

Claim A.1.1. *For every $f \in \mathsf{Linear}_k$, the algorithm $\mathcal{A}$ satisfies the following two conditions.*

1. *If $w = C(x)$ for some $x \in \{0,1\}^k$, then $\Pr\left[\mathcal{A}^{C(x)}(f) = f(x)\right] \geq 2/3$.*
2. *If w is $\delta(C)/3$-close to a codeword $C(x)$, then $\Pr\left[\mathcal{A}^{w}(f) \in \{f(x), \perp\}\right] \geq 2/3$.*

Proof. Let $w = C(x)$ for $x \in \{0,1\}^k$ such that $f(x) = 1$ (the case in which $f(x) = 0$ is symmetrical). Since $T_{f,1}$ is a tester for $\Pi_{f,1} \coloneqq \{C(x) \,:\, f(x) = 1\}$, then $\Pr[T^w_{f,1} = 1] \geq 9/10$, and since $T_{f,0}$ is a $\delta(C)/3$-tester for $\Pi_{f,0} \coloneqq \{C(x) \,:\, f(x) = 0\}$ and w is $\delta(C)$-far from $\Pi_{f,0}$, then $\Pr[T^w_{f,0} = 0] \geq 9/10$. Thus, by the definition of $\mathcal{A}$ it holds that $\Pr\left[\mathcal{A}^w(f) = f(x)\right] \geq (9/10)^2$. Next, assume that w is $\delta(C)/3$-close to a codeword $C(x)$ such that $f(x) = 1$ (again, the case in which $f(x) = 0$ is symmetrical). Then, $\Pr[T^w_{f,0} = 1] < 1/10$ and $\Pr\left[\mathcal{A}^w(f) \in \{f(x), \perp\}\right] \geq 9/10$ follows. □

The second condition of Claim A.1.1 does not bound the number of linear functions on which the algorithm $\mathcal{A}$ is allowed to abort (and so, given a corrupted codeword, $\mathcal{A}$ can potentially output $\perp$ on all inputs). However, by adapting of the techniques of Ben-Sasson et al. [BSGH+06, Lemmas 4.9 and 4.10] to the setting of universal-LTCs, we obtain the following claim, which shows that C and $\mathcal{A}$ can be modified to allow for such bound.

Claim A.1.2. *If there exists a code $C : \{0,1\}^k \to \{0,1\}^n$ with distance d and rate r, and an algorithm $\mathcal{A}$ with query complexity q, which satisfies the conditions of Claim A.1.1, then there exists a constant $\delta_{\mathsf{radius}} > 0$, a code $C' : \{0,1\}^k \to \{0,1\}^{n'}$ with distance $\Theta(d)$ and rate $\Theta(r)$, and an algorithm $\mathcal{B}$ that for every (explicitly given) $f \in \mathsf{Linear}_k$ makes $O(q)$ queries to a string $w \in \{0,1\}^{n'}$ and satisfies the following condition: If w is δ_{radius}-close to a codeword $C'(x)$, then there exists a family $\mathcal{F}$ of at least $(9/10) \cdot 2^k$ functions in Linear_k such that for every $f' \in \mathcal{F}$ it holds that $\Pr\left[\mathcal{B}^w(f') = f'(x)\right] \geq 9/10$.*

We omit the proof of Claim A.1.2, since it follows by a trivial adaptation of [BSGH+06, Lemmas 4.9 and 4.10] to our setting. We mention that the main idea is that by repeating heavily probed locations in the code, we can modify $\mathcal{A}$ such that on an average f it make queries that are nearly uniformly, and then use this "average smoothness" to bound the fraction of functions on which we are forced to abort.

The proof of Theorem A.1 follows by noting that given a slightly corrupted copy of $C'(x)$, for every $f \in \mathsf{Linear}_k$ we can use the algorithm $\mathcal{B}$ of Claim A.1.2 to extract the value of $f(x)$ using the self correctability of linear functions. In more detail, let $w \in \{0,1\}^{n'}$ such that $\delta(w, C'(x)) \leq \delta_{\mathsf{radius}}$ for some $x \in \{0,1\}^k$, and let $i \in [k]$. To decode x_i, we uniformly choose $g \in \mathsf{Linear}_k$, invoke $\mathcal{B}^w(g)$ and $\mathcal{B}^w(g + x_i)$, and output $\mathcal{B}^w(g) + \mathcal{B}^w(g + x_i)$. By the union bound, with probability at least

$1 - 2/10$ both g and $g + x_i$ are functions on which $\mathcal{B}$ succeeds with probability at least $9/10$. Thus, with probability at least $(8/10) \cdot (9/10)$, both $\mathcal{B}^w(g) = g(x)$ and $\mathcal{B}^w(g + x_i) = g(x) + x_i$, and so their summation (over $\mathsf{GF}(2)$) is x_i. □

Generalizing to Self-Correctable Families of Functions. We remark that the only place in which the proof of Theorem A.1 relies on $\mathcal{F}$ being the family of all linear functions is that the latter family admits self correction. Therefore, the same proof holds for any family of functions $\mathcal{F} = \left\{f_i + b : \{0,1\}^k \to \{0,1\}\right\}_{i \in [M], b \in \{0,1\}}$ that is self correctable.

B Proof of Proposition 6.3

Let $k, m \in \mathbb{N}$ such that $m \leq k$. We show that for every $\tau < m$ and $\varepsilon \geq 1/\mathsf{polylog}(n)$, where $n = \binom{k}{m-\tau} \cdot \max\{2^{2^{m-\tau}}, k\}$, there exists exists a (one-sided error) $\mathsf{universal\text{-}LTC}_\varepsilon$ $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n)}$ for $\mathsf{Junta}_{m,k}$ with linear distance and query complexity $\widetilde{O}(\tau) + O(1/\varepsilon)$.

Let $\tau < m$ and $\varepsilon \geq 1/\mathsf{polylog}(n)$. We bundle the long code encoding of each projection of x to $(m - \tau)$ coordinates; that is, denote the $(m - \tau)$-dimensional long code by $\mathsf{LC} : \{0,1\}^{m-\tau} \to \{0,1\}^{2^{2^{m-\tau}}}$, denote the set of all subsets of $[k]$ of cardinality $m - \tau$ by $\mathcal{S}^{(m-\tau)} = \{S' \subseteq [k] \ : \ |S'| = m - \tau\}$. We bundle the encodings $\{\mathsf{LC}(x|_{S'})\}_{S' \in \mathcal{S}^{(m-\tau)}}$ according to Construction 4.4.

Recall that in Construction 4.4 we bundle encodings $E_i, \ldots, E_s$ with an (arbitrary) error-correcting code ECC (which can be encoded by a circuit of quasilinear size in k and has linear distance) and with a PCPP for every E_i, which ascertains that a pair (a, b) satisfies $a = \mathsf{ECC}(y)$ and $b = E_i(y)$ for some y. Here, the encodings will correspond to the long code encodings of x projected to $(m - \tau)$-subsets in $\mathcal{S}^{(m-\tau)}$, i.e., $\{\mathsf{LC}(x|_{S'})\}_{S' \in \mathcal{S}^{(m-\tau)}}$. Note that each $\mathsf{LC}(x|_{S'})$ can be computed by a circuit of size $O(2^{2^m} \cdot m) = \widetilde{O}(n)$. Hence, by Theorem 4.3, for every $S' \in \mathcal{S}^{(m-\tau)}$ there exist a PCPP oracle $\xi_{S'}$, as required in Construction 4.4, of length $\widetilde{O}(n)$. We obtain the code $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n)}$ given by

$$C(x) = \left(\mathsf{ECC}(x)^r, \left(\mathsf{LC}(x|_{S'})\right)_{S' \in \mathcal{S}^{(m-\tau)}}, \left(\xi_{S'}(x)\right)_{S' \in \mathcal{S}^{(m-\tau)}}\right). \tag{B.1}$$

We show that C is a $\mathsf{universal\text{-}LTC}_\varepsilon$ for $\mathsf{Junta}_{m,k}$ with query complexity $\widetilde{O}(\tau) + O(1/\varepsilon)$.

Fix $\varepsilon > 0$, $f \in \mathsf{Junta}_{m,k}$, and write $f(x) = f'(x|_S)$, where S denotes the m influencing coordinates of f. Denote by T the first τ coordinates in S. For every $i \in T$, let $S'_i \in \mathcal{S}^{(m-\tau)}$ be a $(m - \tau)$-subset that contains i. Denote by D the $O(1)$-query corrector of the long code. Using amplification, assume that the corrector D and the bundle consistency-test (see Proposition 4.5) make at most $O(\log(\tau))$ and $O(\log(\tau)/\varepsilon)$ queries (respectively) and obtain soundness error that is (strictly) less than $1/(10(\tau + 1))$.

Consider the ε-tester T_f for the subcode $\Pi_f = \{x \in \{0,1\}^k \ : \ f(x) = 1\}$, which has oracle access to a purported bundle $w \in \{0,1\}^{\widetilde{O}(n)}$ that is supposed to equal Eq. (B.1); that is, w allegedly consists of three parts: (1) the purported anchor $\widetilde{\mathsf{ECC}}(x)$, (2) the purported long code encodings $\left(\widetilde{\mathsf{LC}}(x|_{S'})\right)_{S' \in \mathcal{S}^{(m-\tau)}}$, and (3) the purported PCPs of proximity $\left(\widetilde{\xi}_{S'}(x)\right)_{S' \in \mathcal{S}^{(m-\tau)}}$. Note that we use $\widetilde{z}$ to denote a string that is allegedly equal to z. The tester T_f operates as follows:

1. $\mathsf{Consistency\ Test}$: Invoke the bundle consistency test on w, with respect to proximity parameter ε and the purported encoding $\widetilde{\mathsf{LC}}(x|_{S \setminus T})$, as well as $\widetilde{\mathsf{LC}}(x|_{T_i})$, for every $i \in T$. Reject if any of the tests fail. (The query complexity of this step is $O(\tau \cdot \log(\tau)/\varepsilon)$.)

2. Direct recovery of t variables: Decode $x|_T$ using the self correction of the long code; that is, for every $i \in T$ decode x_i from $\widetilde{\mathsf{LC}}(x|_{S'_i})$ (recall that S'_i is a $(m-\tau)$-subset that contains i), using the corrector D. Denote the string of recovered values by z. (The query complexity of this step is $O(\tau \cdot \log(\tau))$.)

3. Computing the induced $(m-\tau)$-junta: Choose $f'' : \{0,1\}^{m-\tau} \to \{0,1\}$ such that $f''(y) = f'(z \circ y)$, decode f'' from the purported long code encoding $\widetilde{\mathsf{LC}}(x|_{S \setminus T})$ using the corrector D, and accept if and only if it returns 1. (The query complexity of this step is $O(\log(\tau))$.)

The perfect completeness of T_f follows by the one-sided error of the bundle consistency test and the long code corrector D. For the soundness, assume that w is ε-far from Π_f. By Proposition 4.5, we can assume that there exists $y \in \{0,1\}^k$ such that w is ε-close to $C(y)$, and since w is ε-far from Π_f, it holds that $f(y) = 0$; furthermore, $\widetilde{\mathsf{LC}}(y|_{S \setminus T})$ is ε-close to $\mathsf{LC}(y|_{S \setminus T})$, and each $\widetilde{\mathsf{LC}}(y|_{S'_i})$ is ε-close to $\mathsf{LC}(y|_{S'_i})$, otherwise the bundle consistency test rejects with probability at most $(\tau+1)/(10(\tau+1))$. Thus, in Step 2, the corrector D successfully recovers $y|_T$ with probability $(1/10) \cdot \tau/(10(\tau+1))$, and so, with probability at least $2/3$, in Step 3 the tester T_f correctly computes $f''(y|_{S \setminus T}) = f'(y|_T \circ y|_{S \setminus T}) = f(y) = 0$ and rejects. This concludes the proof of Proposition 6.3.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de