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

March 19, 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 $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i\in[M]}$ 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.

We also consider a variant of universal-LTCs wherein the testing procedures are also given free access to a short proof, akin the MAPs of Gur and Rothblum (ITCS 2015). We call such codes "universal locally *verifiable* codes" (universal-LVCs). We show universal-LVCs of length $\widetilde{O}(n^2)$ for t-ary constraint satisfaction problems (t-CSP) over k variables, with proof length and query complexity $\widetilde{O}(n^{2/3})$, where $t = O(1)$ and $n \geq k$ is the number of constraints in the CSP instance. In addition, we prove a lower bound of $p \cdot q = \widetilde{\Omega}(k)$ for every polynomial length universal-LVC for CSPs (over k variables) having proof complexity p and query complexity q.

Lastly, we give an application for *interactive proofs of proximity* (IPP), introduced by Rothblum et al. (STOC 2013), which are interactive proof systems wherein the verifier queries only a sublinear number of input bits and soundness only means that, with high probability, the input is close to an accepting input. We show that using a small amount of interaction, our universal-LVC for CSP can be, in a sense, "emulated" by an IPP, yielding a 3-round IPP for CSP with sublinear communication and query complexity.

*This research was partially supported by the Israel Science Foundation (grant No. 671/13).

Contents

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).

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 $\Pi_i := \{C(x) \,:\, f_i(x) = 1\}$ 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 remark 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, note 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), and 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. However, the ability to test all subcodes comes at the cost of great redundancy, as 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*. 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 thus 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 wish to highlight a close connection between universal-LTCs and a relaxed form of local decodability. A code is said to be a relaxed locally decodable code (relaxed-LDC) [BSGH+06] if there exists a local algorithm (decoder) that is given a location to decode and query access to an alleged codeword. If the codeword is valid, the decoder successfully decodes, 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.[1].

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. Is is straightforward to 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.

1.1 Results for Universal Locally Testable Codes

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 a "canonical" universal-LTC for *any* family of functions.

Theorem 1 (informally stated, see Theorem 3.2)**.** *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 3.7)**.** *Let $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i \in [M]}$ 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 $n = M^{\Omega(1)}$. 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 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.

[1] We also note that, under certain conditions, universal-LTCs imply (non-relaxed) local decodability, see Appendix A

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,[2] with query complexity $\widetilde{O}(m)$; note that for constant m, the number of functions M is an arbitrarily large *polynomial* in the code's length $k^{1.01}$, 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 3.3 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 3.11 for a precise statement.)

1.2 Universal Locally Verifiable Codes

Next, we consider a variant of universal-LTC wherein the testing procedures are also given free access to a short proof. In more detail, we say that a property Π has an *$\mathcal{MA}$ proof of proximity* (MAP) [GR15] if there exists a probabilisitic algorithm (verifier) V that gets a proximity parameter $\varepsilon > 0$ and a short (sublinear) proof π as well as oracle access to a string w. The verifier satisfies, with high probability, the following conditions: If $w \in \Pi$, there exists proof π such that $V^w(\pi, \varepsilon)$ accepts, and if w is ε-far from Π, then for every alleged proof π, the verifier $V^w(\pi, \varepsilon)$ rejects.

We say that a code $C : \{0,1\}^k \to \{0,1\}^n$ is a *universal locally verifiable code* (universal-LVC) for a family of functions $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i \in [M]}$ with *proof complexity* p and *query complexity* q if for every $i \in [M]$ the subcode $\Pi_i \coloneqq \{C(x) \,:\, f_i(x) = 1\}$ has an MAP with proof length p and query complexity q.[3]

We show quadratic length universal-LVCs of sublinear proof and query complexity for a large and natural complexity class, for which every polynomial length universal-LTC must have almost linear query complexity. Specifically, let $n \geq k$, and denote by $\mathsf{CSP}_{n,k}$ the set of all instances of *constraint satisfaction problems* with n constraints of constant arity over k variables.

Theorem 3 (informally stated, see Theorem 4.3)**.** *There exists a* universal-LVC $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n^2)}$ *for* $\mathsf{CSP}_{n,k}$ *with proof and query complexity* $\widetilde{O}(n^{2/3})$*. More generally, for every* $\alpha > 0$ *it is possible to obtain proof length* $\widetilde{O}(n^{2\alpha})$ *and query complexity* $\widetilde{O}(n^{1-\alpha})$*.*

In contrast, as stated above, every polynomial length universal-LTC for $\mathsf{CSP}_{n,k}$ has query complexity that is roughly linear in k. Actually, we provide a lower bound on the tradeoff between the two complexity measures of universal-LVCs for $\mathsf{CSP}_{n,k}$.

Theorem 4 (informally stated, see Corollary 4.5)**.** *For every polynomial* (in k) *length* universal-LVC *for* $\mathsf{CSP}_{n,k}$ *with proof complexity* p *and query complexity* q *it holds that* $p \cdot q = \widetilde{\Omega}(k)$*. For* $p = 0$ *(i.e., a* universal-LTC*), the query complexity is* $\widetilde{\Omega}(k)$*.*

Note that for $n = \Theta(k)$, Theorem 3 gives a universal-LVC of length $\widetilde{O}(k^2)$, and proof and query complexity $\widetilde{O}(k^{2/3})$ each, whereas Theorem 4 shows that such a universal-LVC (of length $\mathsf{poly}(k)$) must have either query or proof complexity $\widetilde{\Omega}(\sqrt{k})$.

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

[3] We remark that if we do not restrict the length of the proof, then *every* property Π can be verified trivially using only a constant amount of queries, by considering an MAP proof that contains a full description of the input and testing identity between the proof and the input.

1.3 An Application for Interactive Proofs of Proximity

An *interactive proof of proximity* (IPP) [RVW13] can be thought of as a generalization of the notion of MAP, in which the verifier is allowed to interact with an omniscient prover (instead of a "static" proof). More accurately, an IPP is an interactive proof system wherein an all powerful (yet untrusted) prover interacts with a verifier that only has oracle access to an input x. The prover tries to convince the verifier that x has a particular property Π. Here, the guarantee is that for inputs in Π, there exists a prover strategy that will make the verifier accept with high probability, whereas for inputs that are far from Π, the verifier will reject with high probability no matter what prover strategy is employed.[4]

Rothblum et al. [RVW13] showed that, loosely speaking, every language in $\mathcal{NC}$ has an IPP with $\widetilde{O}(\sqrt{n})$ query and communication complexities, albeit this IPP requires a large $(\mathsf{polylog}(n))$ number of rounds of interaction. However, for IPPs that use a small number of rounds of interactive (in particular, MAPs) only results for much lower complexity are known (e.g., for context-free languages and languages that are accepted by small read-once branching programs [GGR15]).

We show that the universal-LVC in Theorem 3 can be, in a sense, "emulated" using a small (constant) amount of interaction rounds. This yields the following IPP.

Theorem 5 (informally stated, see Theorem 4.8)**.** *Let $n \geq k$. For every $\varphi \in \mathsf{CSP}_{n,k}$ there exists a 3-round IPP for the property $\pi_\varphi \coloneqq \{x \in \{0,1\}^k \ : \ \varphi(x) = 1\}$ with communication and query complexity $n^{6/7+o(1)}$. More generally, there exists an $O(1)$-round IPP for Π_φ with communication and query complexity $n^{0.501}$.*

1.4 Organization

In Section 2, we provide the required preliminaries for the main technical sections (i.e., Sections 3 and 4). In Section 3, we present our results regarding universal-LTCs. In Section 4, which is the more involved part of this work, we present our results regarding universal-LVCs and the application for interactive proofs of proximity. In addition, in Appendix A we show a sufficient condition for obtaining *locally decodable codes* from universal-LTCs, and in Appendix B we provide deferred details of proofs from Sections 3 and 4. The reason for deferring these details is that we think that they are quite standard and are likely to be skipped by most readers.

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 $\Delta(x,y) \coloneqq |\{x_i \neq y_i \ : \ i \in [n]\}|$ and their *relative distance* by $\delta(x,y) \coloneqq \frac{\Delta(x,y)}{n}$. 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 $\Delta(x,S) \coloneqq \min_{y \in S} \Delta(x,y)$ and the *relative distance* of x from S by $\delta(x,S) \coloneqq \min_{y \in S} \delta(x,y)$. 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 $x|_I$.

[4] Indeed, MAPs can be thought of as a restricted case of IPPs, in which the interaction is limited to a single message sent from the prover to the verifier.

- 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 (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 Property Testing and Proofs of Proximity

In this section we review the definitions of testers, MAPs and IPPs. We begin with the definition of IPPs and obtain the definitions of testers and MAPs as special cases of IPPs.

Definition 2.1 (Interactive Proof of Proximity [EKR04, RVW13])**.** *Let $n \in \mathbb{N}$. An* interactive proof of proximity *(*IPP*) for property $\Pi \subseteq \Sigma^n$ is an interactive protocol with two parties: a prover $\mathcal{P}$ that has* free *access to input $x \in \Sigma^n$, and a probabilistic verifier $\mathcal{V}$ that has oracle access to x. The parties exchange messages, and at the end of the communication the following two conditions are satisfied:*

1. Completeness: *For every proximity parameter $\varepsilon > 0$ and input $x \in \Pi$ it holds that*
$$\Pr\left[(\mathcal{V}^x, \mathcal{P}(x))(\varepsilon) = 1\right] \geq 2/3.$$
2. Soundness: *For every $\varepsilon > 0$, $x \in \Sigma^n$ that is ε-far from Π, and (cheating) prover $\mathcal{P}^*$ it holds that*
$$\Pr\left[(\mathcal{V}^x, \mathcal{P}^*)(\varepsilon) = 0\right] \geq 2/3.$$

If the completeness condition holds with probability 1, we say that the IPP *has a* one-sided error*, and otherwise we say that the* IPP *has a* two-sided error*.*

An IPP for property Π has query complexity (or locality) $q = q(n, \varepsilon)$ if for every $\varepsilon > 0$ and $x \in \Sigma^n$ the verifier $\mathcal{V}$ makes at most q queries to x, and communication complexity $c = c(n, \varepsilon)$ if for every $\varepsilon > 0$ and $x \in \Sigma^n$ the parties $\mathcal{V}$ and $\mathcal{P}$ exchange at most c bits. A round of communication consists of a single message sent from $\mathcal{V}$ to $\mathcal{P}$ followed by a single message sent from $\mathcal{P}$ to $\mathcal{V}$. An r-round IPP, where $r = (n, \varepsilon)$, is an IPP in which for every $\varepsilon > 0$ and $x \in \Sigma^n$ the number of rounds in the interaction between $\mathcal{V}$ and $\mathcal{P}$ on input x is at most r.

The definition of a tester can be derived from Definition 2.1 by allowing no communication (which effectively eliminates the prover). Similarly, the definition of an MAP can be derived by restricting the communication to a single message from $\mathcal{P}$ to $\mathcal{V}$ (see [GR15] for further details on MAPs). We shall sometimes refer to a tester with respect to proximity parameter ε as an ε-tester, and similarly, we refer to an IPP (or MAP) with respect to proximity parameter ε as an IPP_ε (or MAP_ε).

2.2 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 2.2 (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 2.3 (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).[5]

[5] 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]).

Definition 2.4 (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.[6]

Theorem 2.5 (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 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 $\Pi_i \coloneqq \{C(x) \,:\, f_i(x) = 1\}$ 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$_\varepsilon$.

Section Organization. We start, in Section 3.1, 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 3.2, 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 test. Finally, in Section 3.3, 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.

3.1 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.

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

Theorem 3.2. *Let $t(k)$ be a proper complexity function, and let $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i\in[M]}$ 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., $f_i \in \mathsf{SIZE}\big(t(k)\big)$). Fix $n = M \cdot \widetilde{O}(t(k))$. Then, for every $\varepsilon > 1/\mathsf{polylog}(n)$ there exists a (one-sided error)* universal-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 3.7) in the low query complexity regime, for a reasonable setting of the parameters; see Section 3.2 for a more accurate statement.

The key idea for proving Theorem 3.2 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.

3.1.1 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 3.3. *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 3.4 (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 $t(k) \cdot \mathsf{polylog}(t(k))$.*

3.1.2 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 thrice in this work: In Section 3.1.3 each E_i will simply correspond to a Boolean function $f_i \in \mathcal{F}$, in Section 3.3 the E_i's will correspond to encodings of small chunks x, and in Section 4.1 E_i's will correspond to the encodings of x by different error-correcting codes (i.e., Reed-Muller codes over different finite fields).

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 3.5 (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* $\widetilde{O}(t(k))$,[7] *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 3.5):

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 $\mathsf{bundled\ encodings}$ $(\widetilde{E_1}(x), \ldots, \widetilde{E_s}(x)) \in \{0,1\}^{n\cdot s}$, 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 3.6. *For every bundle* $B(x)$*, as in Construction 3.5, 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. The proof of Proposition 3.6 is by standard case analysis, and so we defer it to Appendix B.1.

3.1.3 Proof of Theorem 3.2

Let $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i\in[M]}$ be a family of functions such that for every $i \in [M]$ it holds that $f_i \in \mathsf{SIZE}\big(t(k)\big)$. Fix $n = M \cdot \widetilde{O}(t(k))$ and $\varepsilon > 1/\mathsf{polylog}(n)$. We set $E_i = f_i$ for every $i \in [M]$,

[7] Note that $L_i \in \mathsf{SIZE}(m)$ by the hypothesis regarding ECC and E_i. Thus, by Theorem 3.4, such a PCPP exists.

bundle these encodings via Proposition 3.6, and denote the bundle by $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n)}$. Note that by Proposition 3.6, 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 3.6, 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),[8] and so T_i rejects.

3.2 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 3.7. *Let $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i \in [M]}$ 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 3.2 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 3.7 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.

Lemma 3.8 ([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* $\log\log|U| + O(1)$.

[8] Formally, Proposition 3.6 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.

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 3.8 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 $M \le (n+3)^{2^{q+\log(k)+O(1)}}$, and so $q \ge \log\log M - \log\log n - \log k - O(1)$.

For the furthermore claim of Theorem 3.7, 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,

$$\forall x \in \{0,1\}^k \quad \Pr[T_i^{C(x)} = f_i(x)] \ge 1 - \delta_{\mathsf{err}}. \tag{3.1}$$

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 $s \le (n+2)^{2^{q'}}$. 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. (3.1), for every x and $i \in [M]$,

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

In particular, we obtain that for every $i \in [M]$ there exists $j \in [s]$ such that $\Pr_x[D_j^{C(x)} = f_i(x)] \ge 1 - \delta_{\mathsf{err}}$. 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 $\Pr_x[f_{i_1}(x) = D_j^{C(x)} = f_{i_2}(x)] \ge 1 - 2\delta_{\mathsf{err}} = 1 - \beta$, in contradiction to the hypothesis. Thus $M \le s \le (n+1)^{2^{q'}}$, and since $q' = O(q)$, then $q = \Omega(\log\log M - \log\log n)$. $\square$

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 3.7 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 3.7 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 $i \neq j$.

3.3 Trading off Length for Locality

The general lower bound in Theorem 3.7 allows for a tradeoff between the universal-LTC's length and locality. We remark that while the canonical universal-LTC in Theorem 3.2 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.

3.3.1 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 3.9 matches the lower bound in Theorem 3.7 in Section 3.3.3.)

Observation 3.9. *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,[9] and so we can use the code in Theorem 2.5, 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 3.9, 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 2.5 as the base code). We remark that the argument underlying Observation 3.9 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 3.10. *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 $O(k^\beta m \log m + k^\beta/\varepsilon)$.*[10]

3.3.2 The Actual Tradeoff

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

Proposition 3.11. *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 3.11 only yields a universal-LTC of quadratic length, whereas Observation 3.9 achieves nearly-linear length.[11]

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 3.1 (where the encodings $\{E_i\}$ correspond to $\{\mathsf{LC}(x|_{S_i})\}$). This yields a universal-LTC *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 3.6 to

[9] 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 2.4).

[10] 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.

[11] It is possible to optimize Proposition 3.11 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 3.9 in the $\widetilde{O}(m)$-locality regime.

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 3.9); 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.2. □

3.3.3 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 3.7 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 3.12. *Let $k, m \in \mathbb{N}$ such $m \leq k$. There exists a universal constant $c > 0$ such that every* universal-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 3.9 up to a constant power. Furthermore, it is possible to improve Proposition 3.12 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 3.12. 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 3.12 is concluded by applying Theorem 3.7 to $\mathcal{F}$. □

Improving the lower bound. We point out a slackness in the proof of Proposition 3.12. Specifically, we apply Theorem 3.7 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 3.12 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 3.12) 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*,[12] juntas in such $\mathcal{F}$ are pairwise far, and so we can apply Theorem 3.7. 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 3.12.

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 3.7 to obtain $q = \Omega(\log\log(k^m \cdot 2^{2^m}) - \log\log n)$, which yields the aforementioned bound.

4 Universal Locally Verifiable Codes and Proofs of Proximity for CSP

Following the discussion in the introduction, we define the $\mathcal{MA}$ analogue of universal-LTCs, i.e., universal-LTCs with MAPs instead of testers. We refer to such codes as "universal locally *verifiable* codes".

Definition 4.1. *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 verifiable code (universal-LVC) for $\mathcal{F}$ with query complexity $q = q(k, \varepsilon)$ and proof complexity $p = p(k, \varepsilon)$ is a code $C : \{0,1\}^k \to \{0,1\}^n$ such that for every $i \in [M]$ and $\varepsilon > 0$, there exists an MAP, with respect to proximity parameter ε, for the subcode $\Pi_i \coloneqq \{C(x) : f_i(x) = 1\}$ with query complexity q and proof complexity p. A universal-LVC is said to have one-sided error if all of its MAPs satisfy perfect completeness.*

Notation. We shall refer to a universal-LVC with respect to a specific proximity parameter $\varepsilon > 0$ as a universal-LVC$_\varepsilon$.

Section Organization. In the first subsection (Section 4.1) we show an efficient universal-LVC for constraint satisfaction problems (CSPs). As discussed in the introduction, this universal-LVC can be viewed as a concise representation (or encoding) of assignments that allows for efficient MAPs for *every* CSP instance. We remark that the bundle consistency test (see Section 3.1) is used in the foregoing construction. Next, in Section 4.2 we show a lower bound on the complexity of universal-LVCs for conjugations (and in particular for CSPs). Finally, in Section 4.3 we show that using *interactive* verification procedures we can, in a sense, emulate the universal-LVC in Section 4.1 and obtain an *interactive proof of proximity* (IPP) for any CSP. Note that this result refers to the standard model of IPPs, where the verifier is given access to a plain assignment (rather than to its encoding).

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

4.1 A Universal Locally Verifiable Code for CSP

Throughout this section, let $k, n, t \in \mathbb{N}$ such that $t \leq k$ (the reader is encouraged to think of t as being relatively small with respect to k). A constraint of arity t on k variables is a predicate $\mathsf{c} : \{0,1\}^k \to \{0,1\}$ that only depends on t coordinates (i.e., a t-junta). We denote the set of all such constraints by $\mathsf{Constraint}_{t,k}$.

Definition 4.2. *A function $\varphi : \{0,1\}^k \to \{0,1\}$ is an instance of a constraint satisfaction problem with n constraints of arity t, denoted $\varphi \in \mathsf{CSP}_{n,t,k}$ (or $\varphi \in \mathsf{CSP}_n$, if t and k are clear from the context), if $\varphi(x) = \bigwedge_{i=1}^{n} \mathsf{c}_i(x_1, \ldots, x_k) = 1$, where $\mathsf{c}_1, \ldots, \mathsf{c}_n \in \mathsf{Constraint}_{t,k}$.*

For example, in our formulation, a k-variate, n-clause 3SAT instance $\varphi : \{0,1\}^k \to \{0,1\}$ can be expressed as a $\mathsf{CSP}_{n,3,k}$ by writing $\varphi(x) = \bigwedge_{i=1}^{n} \mathsf{c}_i(x_1, \ldots, x_k)$, where each c_i is a disjunction of 3 literals from $\{ x_1, \ldots, x_k \} \cup \{ 1 - x_1, \ldots, 1 - x_k \}$. We stress that in Definition 4.2 we allow the constraints to be *arbitrary and different* predicates of the same arity.

The following theorem shows an efficient universal-LVC for constraint satisfaction problems. For simplicity, we assume without loss of generality that $n \geq k$ (otherwise, we add $k - n$ empty clauses).

Theorem 4.3. *Let $n, k, t, m \in \mathbb{N}$ such that $t < k \leq n$ and $\varepsilon > 1/\mathsf{polylog}(n)$.*[13] *There exists a (one-sided error) universal-LVC_ε $C : \{0,1\}^k \to \{0,1\}^{\tilde{O}(m \cdot t \cdot n^2)}$ for $\mathsf{CSP}_{n,t,k}$ with linear distance such that for every $\ell \in [m/2]$, the universal-LVC has proof complexity $\widetilde{O}(m \cdot n^{2\ell/m} \cdot t^\ell)$ and query complexity $\widetilde{O}(m \cdot tn^{1-\ell/m}/\varepsilon)$.*

Note that for constant t, m, and ε we obtain code length $\tilde{O}(n^2)$,[14] proof length $\tilde{O}(n^{2\ell/m})$, and query complexity $\widetilde{O}(n^{1-\ell/m})$. In particular, for $\ell = m/3$ (e.g., for $m = 3$), Theorem 4.3 yields a (nearly) quadratic length universal-LVC with both proof and query complexity $\widetilde{O}(n^{2/3})$. We remark that the proof complexity of our MAP has a factor of t^ℓ (and ℓ may be as large as $m/2$), and so we shall want to choose $m = O(1)$ and work with individual degree $d = n^{1/m}$ polynomials, rather than the usual setting of $m = \log(n)/\log\log(n)$ and $d = \log(n)$.

4.1.1 Motivation and Overview of the Construction

In this subsection we give an overview of the key ideas underlying our universal-LVC for CSP. We assume basic familiarity with algebraic PCP systems. Our general approach follows the arithmetization paradigm, commonly used in many probabilistic proof systems. However, for reasons detailed next, we cannot use the standard arithmetizations used in the PCP literature. We focus on the first step of arithmetization, which is over the integers, and assume for simplicity that only one type of t-ary constraint, denoted c, is used.

The most common arithmetization, which can be traced back to [FGL+91], represents the t-ary instance φ as a *generic* function $\phi : [k]^t \to \{0,1\}$ such that $\phi(i_1, \ldots i_t) = 1$ if and only if the i'th constraint of φ involves the variables $x_{i_1} \ldots, x_{i_t}$. The satisfiability of φ at x is then given by

$$\sum_{i_1,\ldots i_t \in [k]} \phi(i_1, \ldots i_t) \cdot \mathsf{c}(x_{i_1}, \ldots x_{i_t}) = n. \tag{4.1}$$

[13] We believe that the limitation on the proximity parameter can be eliminated, by adapting the techniques in [GGK15] to our setting. We leave the verification of this idea as an open problem.

[14] We remark that the *quadratic* length of our universal-LVC is inherent in our techniques, and it is an open question whether it is possible to obtain sub-quadratic length.

This leads to a PCP oracle of length at least k^t, and at best we can hope to implement it by a universal-LVC that has proof length p and query complexity q such that $p \cdot q \geq k^t$. Our goal is, however, to get both p and q to the sublinear (in k) level.

The large PCP length of Eq. (4.1) lead [BFLS91] to suggest a different representation. Using a *universal* circuit ϕ of size $n' = \tilde{O}(n)$, the satisfiability of φ at x is represented by

$$\exists y \in \{0,1\}^{n'} \qquad \sum_{i \in [k+n']} \phi(i) \cdot \mathsf{c}'\big((xy)|_{S_i}\big), \tag{4.2}$$

where c' is a fixed condition (which depends on c) and each $S_i \subseteq [k+n']$ is a subset of constant cardinality. The problem with Eq. (4.2) is that y is a sequence of auxiliary variables and its assignment in Eq. (4.2) depends on the instance φ (and not only on the assignment x).

Our alternative arithmetization composes the assignment $x \in \{0,1\}^k$ viewed as a function $x : [k] \to \{0,1\}$ with functions $\varphi_1, \ldots, \varphi_t : [n] \to [k]$ that represent the instance φ. Specifically, $\varphi_j(i) = i'$ if $x_{i'} = x(i')$ is the j'th variable of the i'th constraint of φ. Hence, φ is satisfiable if and only if

$$\sum_{i \in [n]} \mathsf{c}\big(x \circ \varphi_1(i), \ldots, x \circ \varphi_t(i)\big) = n. \tag{4.3}$$

Next, we consider the algebraic representation of Eq. (4.3) over a sufficiently large finite field $\mathbb{F}$ (discussed below). For simplicity, we assume throughout the rest of this overview that $n = k$ and $t = O(1)$. We identify $[n]$ (the number of constraints) with some set H^m, where $H \subset \mathbb{F}$. Throughout this work, we shall denote the low-degree extension of a function f by $\widehat{f}$. Let $\widehat{\varphi}_j : \mathbb{F}^m \to \mathbb{F}^m$ and $\widehat{X} : \mathbb{F}^m \to \mathbb{F}$ be the individual degree $n^{1/m}$ extensions of $\varphi_j : H^m \to \{0,1\}$ and the assignment $X : H^m \to \{0,1\}$ (respectively), and let $\widehat{\mathsf{c}} : \mathbb{F}^t \to \mathbb{F}$ be the degree t *multilinear* extension of the constraint $\mathsf{c} : \{0,1\}^t \to \{0,1\}$. Note that $\varphi(x) = 1$ if and only if

$$\sum_{z_1, \ldots, z_m \in H} \widehat{\mathsf{c}}\big(\widehat{X} \circ \widehat{\varphi}_1(z_1, \ldots, z_m), \ldots, \widehat{X} \circ \widehat{\varphi}_t(z_1, \ldots, z_m)\big) = n.$$

The straightforward way to implement an MAP for such arithmetization is as follows. Let $\ell \in [m/2]$ be a parameter that will be used to control a tradeoff between proof and query complexity. The purported proof for the MAP is the polynomial

$$\pi(z_1, \ldots, z_\ell) = \sum_{z_{\ell+1}, \ldots, z_m \in H} \widehat{\mathsf{c}}\big(\widehat{X} \circ \widehat{\varphi}_1(z_1, \ldots, z_m), \ldots, \widehat{X} \circ \widehat{\varphi}_t(z_1, \ldots, z_m)\big), \tag{4.4}$$

specified by its coefficients. Observe that the individual degree of both $\widehat{X}$ and $\widehat{\varphi}_j$ is $|H| = n^{1/m}$ and that the composition of $\widehat{X}$ with φ_j increases the individual degree to $|H|^2$. Note this is in contrast to standard arithmetizations, wherein typically the degree of the proof polynomial is $\widetilde{O}(|H|)$. In addition, note that $\widehat{\mathsf{c}}$ only contributes a factor of t to the degree of π, since the constraint is Boolean, and so we can take its *multilinear* extension (saving an $\exp(t)$ factor that would have arisen had we constructed a universal-LVC for 3-CNF formulas and use reductions to handle general t-ary CSPs.) Observe that the proof length of such MAP is $\deg(\pi)^\ell \cdot \log|\mathbb{F}| = t^\ell \cdot |H|^{2\ell} \cdot \log|\mathbb{F}|$ (where $\deg(\pi)$ is the *individual* degree of π, which equals $t \cdot |H|^2$).

Given the foregoing alleged proof π, the verifier can check that $\sum_{z_1, \ldots z_\ell \in H} \pi(z_1, \ldots, z_\ell) = n$. Thus, ascertaining the validity of the proof reduces to computing π at a random point $r \in \mathbb{F}^\ell$ and comparing it to the right hand side of Eq. (4.4). Recall that the formula φ is hardcoded in the verifier, and so it remains for the verifier to query $\widehat{X} \circ \widehat{\varphi}_j(r, z')$ at all $z' \in H^{m-\ell}$ (which is actually done via self-correction, preceded by a low-degree test). Therefore, it suffices to set the universal-LVC to $\widehat{X}$, the low-degree extension of the assignment (which does not depend on the formula). Observe that the query complexity of such MAP is $t \cdot n^{1-\frac{\ell}{m}} \cdot \log |\mathbb{F}|$ (which is primarily determined by the number summands in π).

Unfortunately, a straightforward application of the MAP above requires the order of the field $\mathbb{F}$ (to which we extend) to be greater than the sum we are checking (i.e., n, the number of constraints), because we cannot afford taking a (pseudo) random linear combination of the constraints, as often done in the PCP literature (since this would increase the length of the proof π and prevent us from obtaining sublinear complexity). This causes the length of the universal-LVC (i.e., the Reed-Muller encoding of the assignment to $\mathbb{F}$) to be roughly n^m.

We overcome this issue by arithmetizing over several (distinct) prime fields $\{\mathbb{F}_q\}_{q\in Q}$ such that: (1) for every $q \in Q$, the order of $\mathbb{F}_q$ is larger (by a constant multiplicative factor) than the individual degree of the proof polynomial, which is $O(|H|^2) = O(n^{2/m})$,[15] and (2) it holds that $\prod_{q\in Q} q > n$ (and so we shall set $|Q| = \Theta(m)$). We then invoke, in parallel, the foregoing MAP for each $\mathbb{F}_q$. This gives us the number of satisfied clauses modulo q, and since $\prod_{q\in Q} q > n$, we can use the Chinese remainder theorem to extract the number of satisfied clauses. Note that each $\mathbb{F}_q$ is of size $O(n^{2/m})$, and so the length of a universal-LVC that consists of the Reed-Muller encodings of the assignment to each field in $\{\mathbb{F}_q\}_{q\in Q}$ is $\tilde{O}(m \cdot n^2)$.

Finally, recall that we wish the verifier to have access to the low-degree extension of an assignment over several finite fields, and so the verifier needs to be able to verify that its input actually consists of several polynomials that are consistent with the low-degree extension of a single assignment. Towards this end we bundle the foregoing polynomials using the PCP-based consistency mechanism we showed in Section 3.1 (which also allows us to ascertain that the assignment is binary).

4.1.2 The actual proof of Theorem 4.3

We construct a universal-LVC that maps each assignment $x \in \{0,1\}^k$ to its low-degree extensions over m distinct finite fields, each of cardinality $O(n^{1/m})$, bundled (via Construction 3.5) in a way that allows for locally verifying that all codewords encode the same assignment. More precisely, fix $d = n^{1/m} - 1$, and let Q be the set of the first $m/2$ primes greater than $10(d^2 t + d)m = O(mt \cdot n^{2/m})$; note that each $q \in Q$ satisfies $q = O(mt \cdot n^{2/m})$ and that $\prod_{q\in Q} q > n$. For every $q \in Q$, denote by $\mathbb{F}_q$ the finite field with q elements.

The universal-LVC. Let $H = [d]$, and note that $H \subset \mathbb{F}_q$ for every $q \in Q$. We fix a bijection $H^m \leftrightarrow [n]$ and use these domains interchangeably. We denote by $X : H^m \to \{0,1\}$ the embedding of an assignment $x \in \{0,1\}^k$ in H^m, given by

$$X(z) = \begin{cases} x_z & \text{if } z \in [k] \\ 0 & \text{otherwise} \end{cases}.$$

For every $q \in Q$, let $\widehat{X}'_q : \mathbb{F}_q^m \to \mathbb{F}_q$ be the unique (individual) degree d extension of X to $\mathbb{F}_q$.

[15] This condition is required for the soundness of the MAP.

To reduce the alphabet to binary, let $C_0 : \mathbb{F}_q \to \{0,1\}^{100 \log |\mathbb{F}_q|}$ be a good linear code, and consider the concatenation of $\widehat{X}'_q$ with C_0 as the inner code, which we denote by $\widehat{X}_q : \mathbb{F}_q^m \to \{0,1\}^{100 \log |\mathbb{F}_q|}$. For convenience, we shall treat $\widehat{X}_q$ as if it maps to $\mathbb{F}_q$, and so whenever we query $\widehat{X}_q$ at a point $z \in \mathbb{F}_q^m$, we actually query the $100 \log |\mathbb{F}_q|$ bits of the codeword $C_0(\widehat{X}_q(z))$ and decode (the $\mathbb{F}_q$ element) $\widehat{X}_q(z)$.

Next, we bundle the Reed-Muller encodings $\{\langle \widehat{X}_q \rangle\}_{q \in Q}$ (where $\langle \widehat{X}_q \rangle$ denotes the evaluation of the function $\widehat{X}_q$ over its entire domain) according to Construction 3.5, so that we can locally test that all of these encodings are consistent with the same message (assignment). Recall that in Construction 3.5 we bundle encodings $E_i, \ldots, E_s$ with an (arbitrary) error-correcting code ECC (which can be computed by a circuit of quasilinear size 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 Reed-Muller encodings $\{\langle \widehat{X}_q \rangle\}_{q \in Q}$ of the assignment X. Note that (exact) verification of m-dimensional Reed-Muller codes over $\mathbb{F}_q$ can be done using circuits of size $m \cdot |\mathbb{F}_q|^m \cdot \mathsf{polylog}|\mathbb{F}_q| = \widetilde{O}(mt \cdot n^2)$, since $|\mathbb{F}_q| = O(mt \cdot n^2)$.[16] Hence, by Theorem 3.4, for every $q \in Q$ there exist a PCPP oracle ξ_q, as required in Construction 3.5, of length $\widetilde{O}(mt \cdot n^2)$. We obtain the code $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(mt \cdot n^2)}$ given by

$$C(x) = \Big(\mathsf{ECC}(x)^r, \big(\langle \widehat{X}_q \rangle \big)_{q \in Q}, \big(\xi_q(x) \big)_{q \in Q} \Big). \tag{4.5}$$

We show that C is a $\mathsf{universal\text{-}LVC}$ for CSP_n. This calls for describing a short (MAP) proof for each $\varphi \in \mathsf{CSP}_n$ and describing how it is verified.

Let $\varphi \in \mathsf{CSP}_n$, and write $\varphi(x) = \bigwedge_{i=1}^n \mathsf{c}'_i(x_1, \ldots, x_k) = 1$, where $\mathsf{c}'_1, \ldots, \mathsf{c}'_n \in \mathsf{Constraint}_{t,k,\{0,1\}}$. Recall that each c'_i is a t-junta, denote its influencing variables by I_i, and note that there exists $\mathsf{c}_i : \{0,1\}^t \to \{0,1\}$ such that $\mathsf{c}'_i(x) = \mathsf{c}_i(x|_{I_i})$. We stress that unlike the overview in Section 4.1.1, each constraint c_i may be a *different* predicate; this will make our arithmetization slightly more involved. Note that each c_i takes *binary* inputs, and so, for every $q \in Q$, we denote by $\widehat{\mathsf{c}}_{i,q} : \mathbb{F}_q^t \to \mathbb{F}_q$ the degree t *multilinear* extension of c_i to $\mathbb{F}_q$. We show an MAP for the subcode $\Pi_\varphi \coloneqq \{C(x) : \varphi(x) = 1\}$. We shall first describe the MAP proof and then describe how it is verified.

The MAP proof (for $C(x)$ being in Π_φ). For every $q \in Q$, consider the following functions.

- Constraint Indicator: For every $i \in [n]$, let $\chi_i : H^m \to \{0,1\}$ be the indicator of the i'th constraint, i.e., for every $z \in H^m = [n]$ it holds that $\chi_i(z) = 1$ if and only if $z = i$. Denote by $\widehat{\chi}_{i,q} : \mathbb{F}_q^m \to \mathbb{F}_q$ the unique, individual degree d, extension of χ_i to $\mathbb{F}_q$. (This component is necessary now since each constraint may be a different predicate.)

- Variable Indicator: For every $j \in [t]$, let $\varphi_j : H^m \to H^m$ be the function that maps a *constraint* index $z \in H^m$ to the j'th *variable* index that appears in the z'th constraint (e.g., if $\mathsf{c}_z = (x_5 \vee x_7 \vee x_{11})$, then $\varphi_1(z) = 5$, $\varphi_2(z) = 7$, and $\varphi_3(z) = 11$). Denote by $\widehat{\varphi}_{j,q} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ the unique, individual degree d, extension of φ_j to $\mathbb{F}_q$. (The variable indicator is the same as in the overview.)

[16] This can be done by checking that each one of the $m \cdot |\mathbb{F}_q|^{m-1}$ axis-parallel lines is a degree d univariate polynomial, and each such check can be done by a circuit of size $|\mathbb{F}_q| \cdot \mathsf{polylog}|\mathbb{F}_q|$.

- $\mathsf{Constraint\text{-}Satisfication\ Indicator}$: Let $\psi_q : \mathbb{F}_q^m \to \mathbb{F}_q$ be the individual degree $d^2t + d$ polynomial given by

$$\psi_q(z_1, \ldots, z_m) = \sum_{i=1}^{n} \widehat{\chi}_{i,q}(z_1, \ldots, z_m) \cdot \widehat{\mathsf{c}}_{i,q}\big(\widehat{X}_q \circ \widehat{\varphi}_{1,q}(z_1, \ldots, z_m), \ldots, \widehat{X}_q \circ \widehat{\varphi}_{t,q}(z_1, \ldots, z_m)\big), \tag{4.6}$$

  where the summation is over $\mathbb{F}_q$. Note that for every $z \in H^m$, the value of $\psi_q(z)$ indicates whether the z'th constraint of φ is satisfied by the assignment encoded in $\widehat{X}_q$. Note that the degree of ψ_q is $d + d^2t$, where d^2 is due to the composition of $\widehat{X}_q$ with $\widehat{\varphi}_{j,q}$.

The prescribed MAP proof for $C(x)$ being in Π_φ) is $\pi_\varphi = \{\pi_{\varphi,q}\}_{q \in Q}$, where $\pi_{\varphi,q} : \mathbb{F}_q^\ell \to \mathbb{F}_q$ is given by

$$\pi_{\varphi,q}(z_1, \ldots, z_\ell) = \sum_{z_{\ell+1},\ldots,z_m \in H} \psi_q(z_1, \ldots, z_\ell, z_{\ell+1}, \ldots, z_m), \tag{4.7}$$

where the summation is over $\mathbb{F}_q$. Note that the length of the MAP proof is $\sum_{q \in Q}(d^2t + d)^\ell \cdot 100 \log |\mathbb{F}_q| = \widetilde{O}(m \cdot n^{2\ell/m} \cdot t^\ell)$, and observe that $\sum_{z_1,\ldots,z_\ell \in H} \pi_{\varphi,q}(z_1, \ldots, z_\ell)$ counts the number of φ's constraints that are satisfied by the assignment encoded in $\widehat{X}_q$ *modulo* q (due to the field's characteristic).

The MAP verifier (for φ). Hereafter, *we shall use $\widetilde{z}$ to denote a string that is allegedly equal to z.* Consider the MAP_ε verifier V_φ for the subcode $\{C(x) : \varphi(x) = 1\}$, which has free access to a purported proof $\widetilde{\pi}_\varphi = \{\widetilde{\pi}_{\varphi,q}\}_{q \in Q}$, which is supposed to equal $\pi_\varphi = \{\pi_{\varphi,q}\}_{q \in Q}$ (as defined above), and oracle access to a purported bundle $w \in \{0,1\}^{\widetilde{O}(mt \cdot n^2)}$ that is supposed to equal Eq. (4.5); that is, w allegedly consists of three parts: (1) the purported anchor $\widetilde{\mathsf{ECC}}(x)$, (2) the purported Reed-Muller encodings $(\langle \widetilde{X}_q \rangle)_{q \in Q}$, and (3) the purported PCPs of proximity $(\widetilde{\xi}_q(x))_{q \in Q}$. Let T be the bundle consistency test in Proposition 3.6. Recall that T is given a proximity parameter ε, an encoding-index parameter $q \in Q$, and oracle access to a purported bundle w. The test T accepts, with high probability, if and only if w is ε-close to $C(x)$, and $\langle \widetilde{X}_q \rangle$ is ε-close to $\langle \widehat{X}_q \rangle$ (i.e., the low-degree extension of a *binary* assignment x).

The verifier V_φ performs the following checks for every $q \in Q$, in parallel, and accepts if none of the checks failed.

1. The MAP proof $\widetilde{\pi}_\varphi$ is consistent with a satisfying assignment: Check that

$$\sum_{z_1,\ldots,z_\ell \in H} \widetilde{\pi}_{\varphi,q}(z_1, \ldots, z_\ell) \equiv n \pmod q.$$

2. The universal-LTC itself is a bundle of Reed-Muller encodings of a binary assignment: Invoke the bundle consistency test T with respect to proximity parameter ε, encoding-index parameter q, and purported bundle w. (Hence, we may assume that $\langle \widetilde{X}_q \rangle$ is ε-close to $\langle \widehat{X}_q \rangle$, which is consistent with x; that is, all $\langle \widehat{X}_q \rangle$'s are pairwise consistent with the same *binary* assignment x.)

3. The MAP proof $\widetilde{\pi}_{\varphi,q}$ is consistent with the universal-LTC w: Compare the evaluation of $\widetilde{\pi}_{\varphi,q}$ and $\pi_{\varphi,q}$ at a random point. That is, recall that the verifier V_φ has the formula φ hard-coded, and so it can evaluate $\pi_{\varphi,q}$ (without help from the prover) by self-correcting $\widetilde{X}_q$,

as follows. Select uniformly at random $r_1, \ldots, r_\ell \in_R \mathbb{F}_q$, and for every $z_{\ell+1}, \ldots, z_m \in H$ and $j \in [t]$, decode $\widetilde{X}_q \circ \widehat{\varphi}_{j,q}(r_1, \ldots, r_\ell, z_{\ell+1}, \ldots, z_m)$ using the Reed-Muller self-corrector, repeated $O((m-\ell) \cdot t \cdot \log(|H|))$ times so that the error probability in the self-correction is $1/(10 \cdot t \cdot |H|^{m-\ell})$ for each point. Denoting the value read by $v_{j,q}(r_1, \ldots, r_\ell, z_{\ell+1}, \ldots, z_m)$, check that

$$\widetilde{\pi}_{\varphi,q}(r_1, \ldots, r_\ell) = \sum_{z_{\ell+1},\ldots,z_m \in H} \sum_{i=1}^{n} \widehat{\chi}_{i,q}(r_1, \ldots, r_\ell, z_{\ell+1}, \ldots, z_m) \cdot \widehat{\mathsf{c}}_{i,q}\big(v_{1,q}(r_1, \ldots, r_\ell, z_{\ell+1}, \ldots, z_m), \ldots, v_{t,q}(r_1, \ldots, r_\ell, z_{\ell+1}, \ldots, z_m)\big). \tag{4.8}$$

(Note that, assuming Test 2 passes (with high probability) and all invocations of the self-corrector were successful,[17] the right-hand side of Eq. (4.8) equals $\pi_{\varphi,q}(r_1, \ldots, r_\ell)$.)

Recall that for each $q \in Q$, the purported proof $\widetilde{\pi}_{\varphi,q}$ is a low-degree polynomial (like $\pi_{\varphi,q}$). Hence, if $\widetilde{\pi}_{\varphi,q}$ and $\pi_{\varphi,q}$ agree (with high probability) on a random point, as checked in Test 3, then $\widetilde{\pi}_{\varphi,q} = \pi_{\varphi,q}$. Note that $\sum_{z_1,\ldots,z_\ell \in H} \pi_{\varphi,q}(z_1, \ldots, z_\ell)$ counts the number of constraints of φ that the binary assignment x satisfies modulo q (where Test 2 asserts that all $\pi_{\varphi,q}$'s refer to the same assignment x). By Test 1, it follows that $\sum_{z_1,\ldots,z_\ell \in H} \pi_{\varphi,q}(z_1, \ldots, z_\ell)$ is congruent to n modulo q. Since this holds for all $q \in Q$, then by the Chinese remainder theorem, $\sum_{z_1,\ldots,z_\ell \in H} \pi_{\varphi,q}(z_1, \ldots, z_\ell) \equiv n \pmod{\prod_{q \in Q} q}$, and since $\prod_{q \in Q} q \geq n$, the assignment x satisfies the formula φ.

Note that for each of the $O(m)$ primes in Q, the verifier V_φ makes $O(1/\varepsilon)$ queries during the bundle consistency test and then queries $t \cdot |H|^{m-\ell} = t \cdot n^{1-(\ell/m)}$ points in $\widehat{X}_q$ via (amplified) self-correction of $\widetilde{X}_q$. Thus, the total query complexity is

$$\sum_{q \in Q} \left(O\left(\frac{1}{\varepsilon}\right) + t n^{1-\frac{\ell}{m}} \cdot O(m \log(|H|)) \right) \cdot \log(|\mathbb{F}_q|) = \widetilde{O}\left(mt \cdot n^{1-\frac{\ell}{m}} \cdot \frac{1}{\varepsilon} \right).$$

Perfect completeness follows from the one-sided error of the bundle test and the self-correction procedure. The following claim establishes the soundness of V_φ.

Claim 4.3.1. *If w is ε-far from the subcode $\{C(x) : \varphi(x) = 1\}$, then for every alleged* MAP *proof $\widetilde{\pi}_\varphi$, it holds that $Pr[V_\varphi^w(\widetilde{\pi}_\varphi) = 0] \geq 2/3$.*

The proof of Claim 4.3.1 is a straightforward analysis of the construction, and so we defer its proof to Appendix B.3. This concludes the proof of Theorem 4.3.

4.2 Lower Bounds on Verifying Conjugation Properties

Denote by Conjugation the set of all conjugations (of *at most* k variables); that is, Conjugation $= \{ f_S(x_1, \ldots, x_k) = \wedge_{i \in S} x_i \}_{S \subseteq [k]}$. The following theorem shows a lower bound on the universal-LVC

[17] Note that $\pi_{\varphi,q}$ is well defined if the purported bundle w is close to a codeword $C(x)$, which Test 2 asserts. In this case,

$$\pi_{\varphi,q}(z_1, \ldots, z_\ell) = \sum_{z_{\ell+1},\ldots,z_m \in H} \sum_{i=1}^{n} \widehat{\chi}_{i,q}(z_1, \ldots, z_m) \cdot \widehat{\mathsf{c}}_{i,q}\big(\widehat{X}_q \circ \widehat{\varphi}_{1,q}(z_1, \ldots, z_m), \ldots, \widehat{X}_q \circ \widehat{\varphi}_{t,q}(z_1, \ldots, z_m)\big),$$

where $\widehat{X}_q$ is the low-degree extension of x to $\mathbb{F}_q$. Hence, the verifier V_φ, which has the formula φ hard-coded, can evaluate $\pi_{\varphi,q}$ by self-correcting $\widetilde{X}_q$.

complexity of $\mathsf{Conjugation}$, which in particular, yields a lower bound on the $\mathsf{universal\text{-}LVC}$ complexity of CSP.

Theorem 4.4. *Suppose $C : \{0,1\}^k \to \{0,1\}^n$ is a code of constant relative distance $\delta(C)$, and fix $\varepsilon < \delta(C)$. If C is a $\mathsf{universal\text{-}LVC}_\varepsilon$ for $\mathsf{Conjugation}$ with proof complexity p and query complexity q, then $p \cdot q = \Omega(k/\log n)$.*

Note that the foregoing lower bound trivializes for $n = 2^k$, and indeed there exists a $\mathsf{universal\text{-}LTC}$ for $\mathsf{Conjugation}$ of roughly such length (see Theorem 3.2). As an immediate consequence of Theorem 4.4, we obtain the following corollary.

Corollary 4.5. *Suppose $C : \{0,1\}^k \to \{0,1\}^n$ is a code of constant relative distance $\delta(C)$, and fix $\varepsilon < \delta(C)$. If C is a $\mathsf{universal\text{-}LVC}_\varepsilon$ for $\mathsf{CSP}_{n,k}$ with proof complexity p and query complexity q, then $p \cdot q = \Omega(k/\log n)$.*

We prove Theorem 4.4 by a reduction from $\mathcal{MA}$ communication complexity protocols, which we briefly recall next.

4.2.1 Preliminaries: $\mathcal{MA}$ communication complexity

In $\mathcal{MA}$ communication protocols we have a function $f : X \times Y \to \{0,1\}$, for some finite sets X and Y, and three computationally unbounded parties: Merlin, Alice, and Bob. The function f is known to all parties. Alice gets an input $x \in X$, and Bob gets an input $y \in Y$. Merlin sees both A, B, but Alice and Bob share a random string r that Merlin does not see. The protocol starts with a message $\pi = \pi(x,y)$ sent from Merlin to both Alice and Bob, which is supposed to be a proof that $f(x,y) = 1$. Then, the two players exchange messages to verify that indeed $f(x,y) = 1$.

Definition 4.6. *Let $f : X \times Y \to \{0,1\}$. An $\mathcal{MA}$ communication protocol for f, with proof complexity p and communication complexity c is a probabilistic protocol between two parties who share a random string r, and also receive a p-bit string $\pi = \pi(x,y)$, which is a functions of x and y, but independent of r. The parties communicate c bits and output $\langle A(x), B(y)\rangle(r,\pi)$ such that:*

1. *Completeness: for every Yes-input $(x,y) \in f^{-1}(1)$, there exists a proof $\pi \in \{0,1\}^p$ such that*
$$\Pr_r\left[\langle A(x), B(y)\rangle(r,\pi) = 1\right] \geq 2/3.$$
2. *Soundness: for every No-input $(x,y) \in f^{-1}(0)$ and for any alleged proof $\pi \in \{0,1\}^p$,*
$$\Pr_r\left[\langle A(x), B(y)\rangle(r,\pi) = 0\right] \geq 2/3.$$

We shall use the following (tight) lower bound on the $\mathcal{MA}$ communication complexity of the $\mathsf{set\text{-}disjointness}$ problem, in which Alice has input $S \subseteq [k]$, Bob has input $T \subseteq [k]$, and the parties need to decide whether their sets are disjoint; that is, compute the predicate

$$\mathsf{DISJ}_k(S,T) = \begin{cases} 1 & \text{if } |S \cap T| = 0 \\ 0 & \text{if } |S \cap T| \geq 1 \end{cases}.$$

It is well-known (see [KS92]) that the randomized communication complexity of the set-disjointness problem is linear in the length of the inputs. Moreover, Klauck [Kla03] showed the following (tight) lower bound on the $\mathcal{MA}$ communication complexity of set-disjointness.

Theorem 4.7 ([Kla03]). *Every $\mathcal{MA}$ communication complexity protocol for DISJ_k with proof complexity p and communication complexity c satisfies $p \cdot c = \Omega(k)$.*

4.2.2 Proof of Theorem 4.4

Consider the communication complexity problem, in which Alice has input $A \subseteq [k]$, Bob has input $B \subseteq [k]$, and the parties need to decide whether Alice's set is a subset of Bob's set; that is, compute the predicate $\mathsf{SUBSET}_k(A, B) = \begin{cases} 1 & \text{if } A \subseteq B \\ 0 & \text{otherwise} \end{cases}$.

Claim 4.7.1. *Every $\mathcal{MA}$ communication complexity protocol for SUBSET_k with proof complexity p and communication complexity c satisfies $p \cdot c = \Omega(k)$.*

Proof. We reduce from DISJ_k. Let $\mathsf{Prot}_{\mathsf{SUBSET}}$ be an $\mathcal{MA}$ protocol for SUBSET_k with proof complexity p and communication complexity c, and let $S, T \subseteq [k]$ be the inputs of Alice and Bob to the DISJ_k problem. The parties emulate $\mathsf{Prot}_{\mathsf{SUBSET}}$ on inputs $A \coloneqq S$ and $B \coloneqq [k] \setminus T$. Note that if $S \cap T = \emptyset$, then $A = S \subseteq [k] \setminus T = B$. Otherwise, there exists $i \in S \cap T$ such that $i \notin [k] \setminus T = B$, and $A \not\subseteq B$ follows. We stress that the reduction maps 1-instances to 1-instances, and so it preserves membership in the class $\mathcal{MA}$. □

We prove the following claim by adapting the methodology in [BBM11], in which property testing lower bounds are obtained via reductions from communication complexity, to the setting of universal-LTCs.

Claim 4.7.2. *If the universal-LVC C has proof complexity p and query complexity q, then there exists an $\mathcal{MA}$ communication complexity protocol for SUBSET_k with proof complexity p and communication complexity $q \cdot (1 + \log n)$.*

Proof. Let $A, B \subseteq [k]$ be the inputs of Alice and Bob (respectively) to the SUBSET_k problem. Bob computes the codeword $C(B)$, where B is viewed as a k-bit string.[18] Then, Alice invokes the MAP verifier for the subcode $C_A \coloneqq \{C(x) : \wedge_{i \in A} x_i = 1\}$, and answers each of its q queries by communicating with Bob as follows. On query $i \in [n]$, Alice sends i (communicating $\log n$ bits) to Bob, who responds with (a single bit) $C(B)_i$, which Alice provides as answer to the MAP verifier for C_A, denoted V_A. If $A \subseteq B$, then $\wedge_{i \in A} B_i = 1$, and so $C(B) \in C_A$; thus there exists a proof $\pi \in \{0,1\}^p$ such that $\Pr[V_A^{C(B)} = 1] \geq 2/3$. Otherwise (i.e., $A \not\subseteq B$), there exists $i \in A$ such that $i \notin B$, hence $\wedge_{i \in A} B_i = 0$, and so $C(B)$ is $\delta(C)$-far from C_A, and for every $\pi \in \{0,1\}^p$ it holds that $\Pr[V_A^{C(B)} = 0] \geq 2/3$. □

Combining Claim 4.7.1 and Claim 4.7.2 concludes the proof of the Theorem 4.4.

4.3 Constant-Round IPPs for CSP

Recall that an interactive proof of proximity (hereafter, IPP) is an interactive proof system in which the verifier only queries a sublinear number of input bits and soundness only means that, with high probability, the input is close to an accepting input (see Definition 2.1). In this section, we show that using $O(1)$ rounds of interaction, an IPP protocol wherein the verifier has oracle access to an assignment $x \in \{0,1\}^k$ can, in a sense, emulate the universal-LVC for CSP of Theorem 4.3; thus, we obtain an efficient IPP for satisfiability of fixed CSPs. We shall make an effort to keep the round complexity of such IPP to a minimum. We warn that Section 4.1 is a prerequisite for this section.

[18] Via the standard mapping in which the i'th bit of the string is 1 if $i \in B$ and 0 otherwise.

Let $k \in \mathbb{N}$. We consider $\mathsf{CSP}_n = \mathsf{CSP}_{n,t,k}$, where for simplicity of presentation, in this subsection we fix $n = k$ and $t = O(1)$ (generalizing to general values of n, k, t can handled similarly as in Section 4.1). Recall that each *round* of an IPP consists of two messages, one from the prover and one from the verifier (see Section 2.1). We prove the following.

Theorem 4.8. *For every $\varepsilon \geq 1/n^{6/7}$ and $\varphi \in \mathsf{CSP}_n$ there exists a 3-round (one-sided error)* IPP *for the property $\Pi_\varphi = \left\{ x \in \{0,1\}^k : \varphi(x) = 1 \right\}$ with communication and query complexity $O(n^{6/7+o(1)})$.*

We remark that by allowing additional $O(1)$ rounds of interaction, it is possible to obtain both query and communication complexity n^α for any constant $\alpha > 1/2$, see Section 4.3.3.

4.3.1 Overview of the Proof of Theorem 4.8

We start with a brief overview of the main ideas behind the proof of Theorem 4.8. Fixing any $\varphi \in \mathsf{CSP}_n$, let $C(x)$ be the $\mathsf{universal\text{-}LVC}$ encoding of an assignment $x \in \{0,1\}^k$, as used in Theorem 4.3. Recall that $C(x)$ consists of a bundle of Reed-Muller encodings of x over several prime fields $\{\mathbb{F}_q\}_{q \in Q}$,[19] and let V_φ be the MAP verifier for $\Pi_i = \{C(x) : \varphi(x) = 1\}$.

Let $C(x)$ be a *valid* codeword (where $\varphi(x) \in \{0,1\}$). Then, by Theorem 4.3: (1) if $\varphi(x) = 1$, then there exists a proof π such that $\Pr[V_\varphi^{C(x)}(\pi) = 1] = 1$, and (2) if $\varphi(x) = 0$, then for every alleged proof π it holds that $\Pr[V_\varphi^{C(x)}(\pi) = 1] < 1/3$. A closer inspection of the proof of Theorem 4.3 shows that, for every $q \in Q$, the verifier $V_\varphi(\pi)$ generates, as a function of the alleged proof π and its own randomness, a subset of indices $J_q \subseteq [|C(x)|]$ and a vector of values $\vec{v}_q \in \{0,1\}^{|J_q|}$ such that: (1) if $\varphi(x) = 1$, then for every $q \in Q$ there exists a proof π such that $\Pr_{(J_q,\vec{v}_q) \leftarrow V_\varphi(\pi)}[C(x)|_{J_q} = \vec{v}_q] = 1$, and (2) if $\varphi(x) = 0$, then for every alleged proof π there exists $q \in Q$ such that $\Pr_{(J_q,\vec{v}_q) \leftarrow V_\varphi(\pi)}[C(x)|_{J_q} = \vec{v}_q] < 1/3$.[20] Hence, we view V_φ as a *reduction* of verifying that x satisfies φ to verifying that $C(x)|_{J_q} = \vec{v}_q$ for every $q \in Q$. Hereafter, we fix $q \in Q$ and omit it from subscripts.

Recall, however, that in the setting of Theorem 4.8 the verifier does *not* have access to the encoding $C(x)$, but rather only oracle access to the plain assignment x itself. Aiming at sublinear query complexity, the verifier cannot read all of x. Instead the verifier sends the set of locations J to the prover and asks it to prove to it that $C(x)|_J = \vec{v}$. To this end, we use techniques from [RVW13] that allow us to verify claims regarding $C(x)$ by only making a small number of queries to x. This is performed in two steps, which we describe next.

The first step is to strengthen the soundness condition of V_φ such that, with high probability, if x is ε-far from $\Pi_i \coloneqq \{z \in \{0,1\}^k : \varphi(z) = 1\}$, not only $C(x)|_J \neq \vec{v}$, but also for every x' that is ε-close to x (simultaneously) it holds that $C(x')|_J \neq \vec{v}$. That is, if x is ε-far from Π_i, then it is ε-far from $\{z \in \{0,1\}^k : C(z)|_J = \vec{v}\}$. The second step is to invoke an IPP (due to [RVW13]) for verifying membership in $\{z \in \{0,1\}^k : C(z)|_J = \vec{v}\}$, where C consists of Reed-Muller encodings. Details follow.

[19] Actually, C consists of the foregoing Reed-Muller encodings, bundled with PCPPs that ascertain the consistency of the encodings (see Construction 3.5). However, in the context of Theorem 4.8, we shall not need these PCPPs, and we view C as consisting solely of the low-degree extensions.

[20] This is because (1) the verifier is *non-adaptive*, and (2) assuming $C(x)$ is valid, the verifier only needs to make queries to the Reed-Muller encodings (and do *not* need to query the PCPP oracles that are used for consistency testing).

Denote the query complexity of the verifier V_φ by ℓ. We start by reducing the soundness error of V_φ, via S parallel repetitions (at the cost of increasing the the query complexity to $S \cdot \ell$). Note that the amplified verifier V'_φ generates a pair $(J, \vec{v})$ of $O(S \cdot \ell)$ locations and values, such that if $\varphi(x) = 0$, then $\Pr_{(J,\vec{v})}[C(x)|_J = \vec{v}] = \exp(-S)$. Observe that if x is ε-far from satisfying φ (and in particular $\varphi(x') = 0$), then the probability there exists x' that is ε-close to x such that $C(x')|_J = \vec{v}$ is at most $\binom{n}{\varepsilon n} \cdot \exp(-S)$.

Therefore, by setting $S = \Theta(\varepsilon \cdot n \log n)$ we obtain that with high probability no x' that is ε-close to x satisfies $C(x')|_J = \vec{v}$. Thus, if x is ε-far from $\{x \in \{0,1\}^k : \varphi(x) = 1\}$, then with high probability (over the pair $(J, \vec{v})$, chosen by V'_φ) the assignment x is ε-far from the affine subspace $A_{J,\vec{v}} \coloneqq \{x \in \{0,1\}^k : C(x)|_J = \vec{v}\}$.

Therefore, the foregoing constitutes a 2-message "reduction": The prover sends the MAP proof (constructed as in Theorem 4.3) that x satisfies φ, and the verifier sends back a set of random locations J, asking the prover to provide a vector $\vec{v}$ and prove that it is equal to $C(x)|_J$. Hence, we performed a randomized reduction of verifying that x satisfies φ to verifying membership in the affine subspace $A_{J,\vec{v}}$. Fortunately, 3-message IPPs with sublinear communication and query complexity are known for testing membership in affine subspaces that are induced by Reed-Muller codes. Furthermore, these IPPs also have sublinear communication and query complexity for sub-constant values of ε. This is crucial since we perform $S = \Theta(\varepsilon \cdot n \log n)$ parallel repetitions of V_φ, which adds a factor of $\Theta(\varepsilon \cdot n \log n)$ to the communication complexity, and since we aim for sublinear communication complexity, the proximity parameter must be sub-constant. Finally, we compose the aforementioned reduction protocol with an IPP for membership in $A_{J,\vec{v}}$, and hence obtain an IPP for $\left\{ x \in \{0,1\}^k : \varphi(x) = 1 \right\}$.

To present the actual proof of Theorem 4.8, we shall need to define the following property of membership in the affine subspace that corresponds to the Reed-Muller code.

Definition 4.9 (PVAL). *Let $\mathbb{F}$ be a finite field, $J \subseteq \mathbb{F}^m$, and $\vec{v} \in \mathbb{F}^{|J|}$. The property $\mathsf{PVAL}^{\mathbb{F},d,m}_{J,\vec{v}}$ (or just $\mathsf{PVAL}^{\mathbb{F}}_{J,\vec{v}}$, when d and m are clear from the context) consists of all strings $x \in \{0,1\}^{d^m}$ such that their (individual) degree d extension to $\mathbb{F}$, denoted $\widehat{X} : \mathbb{F}^m \to \mathbb{F}$, takes the values $\vec{v}$ on the coordinates J; that is,*

$$\mathsf{PVAL}^{\mathbb{F}}_{J,\vec{v}} = \{x \in \{0,1\}^{d^m} : \widehat{X}(J) = \vec{v}\}.$$

The following theorem, due to Rothblum et al. [RVW13], shows that PVAL has efficient IPPs.

Theorem 4.10 ([RVW13, Theorem 3.12]). *Let $d, m \in N$, and let $\mathbb{F}$ be a finite field. Fix parameters r and q such that $r \leq \min(d, |F|/10)$ and $q > \max\{(d^r)^{1+o(1)}, |\mathbb{F}|\}$.*

Then, for every $J \subseteq \mathbb{F}^m$, $\vec{v} \in \mathbb{F}^{|J|}$, and any $\varepsilon \geq 1/q^{1-o(1)}$ there exists a one-sided error, $(2r+1)$-message (where the first message is sent by the prover) IPP_ε for $\mathsf{PVAL}^{\mathbb{F},d,m}_{J,\vec{v}}$ with communication complexity $(d^{m-r} + |J| \cdot d) \cdot q^{o(1)}$ and query complexity q.

We remark that the product of the proof and query complexities in Theorem 4.10 can be made almost linear in some cases; specifically, for $r = \frac{\log q}{\log d}$ we obtain communication complexity $\frac{d^m}{q^{1-o(1)}} + |J| \cdot d \cdot q^{o(1)}$ and query complexity q. We shall, however, use $r = O(1)$.

4.3.2 The Actual Proof of Theorem 4.8

Let $\varphi \in \mathsf{CSP}_n$, and write $\varphi(x) = \bigwedge_{i=1}^n \mathsf{c}'_i(x_1, \ldots, x_k)$, where $\mathsf{c}'_1, \ldots, \mathsf{c}'_n \in \mathsf{Constraint}_{t,k,\{0,1\}}$. Recall that each c'_i is a t-junta, denote its influencing variables by I_i, and note that there exists $\mathsf{c}_i : \{0,1\}^t \to \{0,1\}$ such that $\mathsf{c}'_i(x) = \mathsf{c}_i(x|_{I_i})$.

We show an IPP for the property $\Pi_\varphi \coloneqq \{x \in \{0,1\}^k \ : \ \varphi(x) = 1\}$. As discussed in the overview, we begin by using a similar construction to that of Theorem 4.3, to the end of performing a randomized reduction of verifying that the assignment x satisfies φ to verifying membership in the affine subspace induced by Reed-Muller encodings of x. More accurately, we shall use a "bare-bones" version of the foregoing $\mathsf{universal\text{-}LTC}$, which only consists of Reed-Muller encodings of x over several prime fields (note that we omit both the alphabet reduction, and the PCP-based consistency testing mechanism), and whose MAP verifiers do not query the $\mathsf{universal\text{-}LTC}$, but rather send to the prover the queries they wish to make. We stress that this construction do *not* include the anchor and PCPPs in Construction 3.5.

For the convenience of the reader, we briefly review the following definitions from Section 4.1, which are needed to describe the foregoing "bare-bones" version of the $\mathsf{universal\text{-}LTC}$ in Theorem 4.3.

Review of the arithmetization in Theorem 4.3. Let $m = O(1)$, to be determined later, and fix $d = n^{1/m} - 1$. Let Q be the set of the first $m/2$ primes that are greater than $2(d^2t + d)m = O(n^{2/m})$. Note that $\prod_{q \in Q} q > n$. Let $q \in Q$. Denote by $\mathbb{F}_q$ the finite field with q elements. Denote by $\widehat{\mathsf{c}}_{i,q} : \mathbb{F}_q^t \to \mathbb{F}_q$ the multilinear extension of c_i to $\mathbb{F}_q$. Let $H = [d]$ (note that $H \subset \mathbb{F}_q$); we fix a bijection $H^m \leftrightarrow [n]$ and use these domains interchangeably. For every $x \in \{0,1\}^k$ consider $X_q : H^m \to \{0,1\}$ given by $X_q(z) = x_z$. Let $\widehat{X}_q : \mathbb{F}_q^m \to \mathbb{F}_q$ be the unique (individual) degree d extension of X_q to $\mathbb{F}_q$.

For every $i \in [n]$, let $\chi_i : H^m \to \{0,1\}$ be the indicator of the i'th constraint, i.e., for every $z \in H^m = [n]$ it holds that $\chi_i(z) = 1$ if and only if $z = i$. Denote by $\widehat{\chi}_{i,q} : \mathbb{F}_q^m \to \mathbb{F}_q$ the unique, individual degree d, extension of χ_i to $\mathbb{F}_q$. For every $j \in [t]$, let $\varphi_j : H^m \to H^m$ be the function that maps a *constraint* index $z \in H^m$ to the j'th *variable* index that appears in the z'th constraint. Denote by $\widehat{\varphi}_{j,q} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ the unique, individual degree d, extension of φ_j to $\mathbb{F}_q$. For every $i \in [n]$ and $j \in [t]$, denote by $\widehat{\chi}_{i,q}$ and $\widehat{\varphi}_{j,q}$ the low-degree extension of χ_i and φ_j to $\mathbb{F}_q$. Finally, let $\psi_q(z_1, \ldots, z_m) : \mathbb{F}_q^m \to \mathbb{F}_q$, given by

$$\psi_q(z_1, \ldots, z_m) = \sum_{i=1}^{n} \widehat{\chi}_{i,q}(z_1, \ldots, z_m) \cdot \widehat{\mathsf{c}}_{i,q}\big(\widehat{X}_q \circ \widehat{\varphi}_{1,q}(z_1, \ldots, z_m), \ldots, \widehat{X}_q \circ \widehat{\varphi}_{t,q}(z_1, \ldots, z_m)\big).$$

Having reviewed the foregoing definitions, we are ready to proceed with the proof of Theorem 4.8.

The 3-round IPP. Let $\varepsilon > 0$, $\ell \in [m/2]$, and $S \in \mathbb{N}$, to be determined later. Consider the following 3-round IPP_ε for the property $\Pi_\varphi \coloneqq \{x \in \{0,1\}^k \ : \ \varphi(x) = 1\}$. The protocol starts by emulating a "bare-bones" version of the MAP verifier of Theorem 4.3, which differs in the following aspects: (1) the consistency test and alphabet reduction are omitted, (2) the soundness of the verifier is amplified via $S = O(\varepsilon n \log n)$ parallel repetitions, and (3) the verifier does *not* make queries to its input, but rather communicates to the prover the queries it wishes to make and asks the prover to assert the values of these queries. Details follow.

Hereafter, we shall denote by $\widetilde{f}$ a function, sent by the prover, which allegedly equals f. For every $q \in Q$, the prover sends a polynomial $\widetilde{\pi}_q : \mathbb{F}_q^\ell \to \mathbb{F}_q$, which allegedly equals $\pi_q(z_1, \ldots, z_\ell) \coloneqq \sum_{z_{\ell+1}, \ldots, z_m \in H} \psi_q(z_1, \ldots, z_\ell, z_{\ell+1}, \ldots, z_m)$, where the summation is over $\mathbb{F}_q$. The verifier first checks that all π_q's are consistent with a satisfying assignment (i.e., checks that $\sum_{z_1, \ldots, z_\ell \in H} \widetilde{\pi}_q(z_1, \ldots, z_\ell) \equiv n \pmod{q}$, for all $q \in Q$). Then, the verifier wishes to evaluate each π_q on S randomly chosen points and compare it to the value of $\widetilde{\pi}_q$ on these points,[21] which amounts to evaluating the low-degree extensions $\{\widehat{X}_q\}_{q\in Q}$ of the assignment x at $S \cdot |H|^{m-\ell}$ points; denote these points by J_q.

Recall, however, that the verifier only has access to the plain assignment x, and not to its encodings $\{\widehat{X}_q\}_{q\in Q}$ (note that evaluating $\widehat{X}_q$ at any point, without assistance from the prover, may require reading the assignment x entirely). Instead the verifier asks the prover to assert the values of $\{\widehat{X}_q\}_{q\in Q}$ at the points it wishes to probe. To that end, the verifier selects uniformly at random $r_q^{(s)} := (r_1^{(s)}, \ldots, r_\ell^{(s)}) \in \mathbb{F}_q^\ell$, for every $s \in [S]$ and sends it to the prover, which in turns sends a vector $\vec{v}_q$ of the evaluations of $\widehat{X}_q$ at J_q, for every $q \in Q$. Finally the parties invoke the IPP in Theorem 4.10 with respect to $(J_q, \vec{v}_q)$, for every $q \in Q$, and accept if and only if all of the invocations accepted. More accurately, the IPP is described as follows. For every $q \in Q$, in parallel, perform the following steps:

1. The prover sends an (individual) degree $d^2t + d$ polynomial $\widetilde{\pi}_q : \mathbb{F}_q^\ell \to \mathbb{F}_q$ (by specifying its coefficients), which allegedly equals:
$$\pi_q(z_1, \ldots, z_\ell) = \sum_{z_{\ell+1},\ldots,z_m \in H} \psi_q(z_1, \ldots, z_\ell, z_{\ell+1} \ldots, z_m).$$
2. The verifier checks that $\sum_{z_1,\ldots,z_\ell \in H} \widetilde{\pi}_q(z_1, \ldots, z_\ell) \equiv n \pmod q$.
3. The verifier selects uniformly at random and sends $r_q^{(s)} := (r_1^{(s)}, \ldots, r_\ell^{(s)}) \in \mathbb{F}_q^\ell$, for every $s \in [S]$.
4. The prover sends $\vec{v}_q \in \mathbb{F}_q^{S\cdot|H|^{m-\ell}\cdot t}$ such that allegedly $\vec{v}_q[s, \vec{z}, i] = \widehat{X}_q \circ \widehat{\varphi}_{i,q}\big(r_1^{(s)}, \ldots, r_\ell^{(s)}, \vec{z}\big)$, for every $s \in [S]$, $\vec{z} \in H^{m-\ell}$, and $i \in [t]$.
5. The verifier checks that, for every $s \in [S]$,
$$\sum_{\vec{z}\in H^{m-\ell}} \sum_{i=1}^{n} \widehat{\chi}_{i,q}\big(r_1^{(s)}, \ldots, r_\ell^{(s)}, z\big) \cdot \widehat{\mathsf{c}}_{i,q}\big(\vec{v}_q[s, z, 1], \ldots, \vec{v}_q[s, z, t]\big) \equiv n \pmod q.$$
6. Fix $J_q = \big(\widehat{\varphi}_{i,q}(r_1^{(s)}, \ldots, r_\ell^{(s)}, \vec{z})\big)_{s\in[S],\vec{z}\in H^{\ell-m},i\in[t]}$, and invoke the IPP for PVAL (Theorem 4.8) on input x (the assignment), field $\mathbb{F}_q$, location set J_q, and evaluation vector $\vec{v}_q$.

Note that in Step 1 the prover communicates $\sum_{q\in Q}(d^2t + d)^\ell \cdot \log |\mathbb{F}_q|$ bits, in Step 3 the verifier sends $\sum_{q\in Q} S \cdot \ell \cdot \log |\mathbb{F}_q|$ bits, and in Step 4, the prover sends $\sum_{q\in Q} S \cdot |H|^{m-\ell} \cdot t \cdot \log |\mathbb{F}_q|$ bits. Hence, prior to the final step (i.e., Step 6), $\widetilde{O}\big(n^{2\ell/m} + S \cdot n^{m-\ell/m}\big)$ bits are being communicated and *no* queries are being made to the assignment x by the verifier.

Finally, the parties invoke the 3-message (starting with the prover) PVAL IPP (in Step 6), whose communication complexity is
$$\left(d^{m-1} + \sum_{q\in Q} |J_q| \cdot d\right) \cdot q^{o(1)} = \left(n^{\frac{m-1}{m}} + S \cdot n^{\frac{m-\ell+1}{m}}\right) \cdot q^{o(1)}$$

[21] Note that by the proof of Theorem 4.3, evaluating each π_q on a *single* randomly chosen point yields constant soundness, and so, in the setting of Theorem 4.8, as discussed in the overview, we obtain soundness $\exp(-S)$ by evaluating each π_q on S randomly chosen points.

and query complexity is q. (Note that only the PVAL protocol actually makes queries to the input x). Fixing $\varepsilon = 1/n^{6/7}$, $q = n^{6/7+o(1)}$, $S = O(\varepsilon n \log n)$, $m = 7$, and $\ell = 3$ yields the claimed complexity. Perfect completeness follows by construction. To show soundness, we shall first need the following claim

Claim 4.10.1. *If $x \notin \Pi_\varphi$, then there exists $q \in Q$ such that* $\Pr_{(J_q,\vec{v}_q)}[\widehat{X}_q(J_q) = \vec{v}_q] < (1/10)^S$.

The proof of Claim 4.10.1 is by a straightforward analysis of the construction, and thus we defer its proof to Appendix B.4. Next, assume that x is ε-far from Π_φ, and observe that by Claim 4.10.1 there exists $q \in Q$ such that

$$\begin{aligned}
\Pr_{(J_q,\vec{v}_q)}[\forall x' \in N_\varepsilon(x) \;\; \widehat{X'}_q|(J_q) \neq \vec{v}_q] &\geq 1 - \binom{n}{\varepsilon n} \cdot \max_{x' \notin \Pi_\varphi} \left\{ \Pr_{(J_q,\vec{v}_q)}[\widehat{X'}_q(J_q) = \vec{v}_q] \right\} \\
&\geq 1 - \binom{n}{\varepsilon n} \cdot (1/10)^S && \text{(Claim 4.10.1)} \\
&\geq 9/10. && (S = O(\varepsilon n \log n))
\end{aligned}$$

(where $N_\varepsilon(x)$ consists of all strings that are ε-close to x). Thus, there exists $q \in Q$ such that with probability $9/10$ over the verifier's randomness, the assignment x is ε-far from $\mathsf{PVAL}^{\mathbb{F}_q}_{J_q,\vec{v}_q}$, and so, by Theorem 4.8, x is rejected with probability at least $9/10 \cdot 9/10$ in the last step of the IPP (the invocation of the PVAL protocol). This concludes the proof of Theorem 4.8.

4.3.3 Round Complexity versus Communication and Query Complexity Tradeoff

The proof of Theorem 4.8 naturally extends to IPPs with a higher round complexity, admitting $O(1)$-round IPPs with proof and query complexity n^α for any constant $\alpha > 1/2$. We sketch below how such extension is performed.

The idea is to replace the emulation of the "bare-bones" MAP verifier V_φ (Steps 1-3 of the IPP in Theorem 4.8) with a sumcheck protocol [LFKN92], in which the summation is striped down in iterations, coordinate-by-coordinate. That is, the protocols starts with m rounds (recall that we arithmetize over m-variate polynomials), where in the j'th round, for every $q \in Q$ and $s \in [S]$, the prover sends a degree $d^2t + d$ *univariate* polynomial $\widetilde{\pi}^{(s)}_{j,q} : \mathbb{F}_q \to \mathbb{F}_q$ that allegedly equals:

$$\pi^{(s)}_{j,q}(z) = \sum_{z_{j+1},\ldots,z_m \in H} \psi_q(r^{(s)}_1, \ldots, r^{(s)}_{j-1}, z, z_{j+1} \ldots, z_m).$$

The verifier then checks the consistency of each $\widetilde{\pi}^{(s)}_{j,q}$ with $\widetilde{\pi}^{(s)}_{j-1,q}$; i.e., verifies that

$$\widetilde{\pi}^{(s)}_{j-1,q}\left(r^{(s)}_{j-1}\right) = \sum_{z \in H} \pi^{(s)}_{j,q}(z),$$

and the j'th round is concluded by letting the verifier select uniformly at random $r^{(s)}_j \in \mathbb{F}_q$ and send it to the prover.

Standard analysis of the sumcheck protocol shows that the larger m is (which in turn dictates the round complexity), the smaller the communication and query complexity of such protocols; in particular for $O(1)$-rounds, we can obtain both query and proof complexity n^β, where $\beta = \beta(m)$ is an arbitrarily small constant. The bottleneck in both query and proof complexity is, however, due to the final step of our IPP, which is an invocation of IPP in Theorem 4.10, wherein both query and proof complexity are inherently $\omega(\sqrt{n})$.

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

[BBM11] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. In *IEEE Conference on Computational Complexity*, pages 210–220, 2011. 4.2.2

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC*, pages 21–31, 1991. 4.1.1

[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

[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. 3.1.1

[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, 2.2, 5, 2.2, 6, 3.1.1, 3.1.2, 11, A, A

[Din07] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12, 2007. 3.1.1

[DR06] Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006. 3.1.1

[EKR04] Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004. 2.1

[FGL+91] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Approximating clique is almost np-complete. 1991. 4.1.1

[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

[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. 2.2, 2.5, 6, 13

[GGR15] Oded Goldreich, Tom Gur, and Ron D. Rothblum. Proofs of proximity for context-free languages and read-once branching programs - (extended abstract). In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 666–677, 2015. 1.3

[GR15] Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 133–142. ACM, 2015. 1.2, 2.1

[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. 3.2, 3.8

[Kla03] Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *Computational Complexity, 2003. Proceedings. 18th IEEE Annual Conference on*, pages 118–134. IEEE, 2003. 4.2.1, 4.7

[KS92] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992. 4.2.1

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. 4.3.3

[Mei14] Or Meir. Locally correctable and testable codes approaching the singleton bound. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:107, 2014. 3.3.1

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994. 3.3.3

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

[RVW13] Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: Delegating computation in sublinear time. In *Proceedings of the 45th annual ACM Symposium on Theory of Computing (STOC)*, 2013. 1.3, 2.1, 4.3.1, 4.3.1, 4.10

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*;[22] in this case we cannot decode. (Indeed, if the aforementioned testers are tolerant, then the foregoing procedure is a local decoder.[23]) Nevertheless, since the foregoing case only happens when the input is *not* a valid codeword, we obtain a procedure that either decodes correctly or detects a corruption in the encoding and aborts (similarly to relaxed-LDCs, see Definition 2.4). 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 $\Pi_{f,1} := \{C(x) : f(x) = 1\}$, then $\Pr[T^w_{f,1} = 1] \geq 9/10$, and since $T_{f,0}$ is a $\delta(C)/3$-tester for $\Pi_{f,0} := \{C(x) : f(x) = 0\}$ 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

[22] 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).

[23] 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.

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 $\delta(w, C'(x)) \le \delta_{\mathsf{radius}}$ 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 Deferred Details of Proofs

In this appendix, we provide the full details of claims that were sketched in Sections 3 and 4. We believe that these details are straightforward implementations of the proof ideas presented in the foregoing sections.

B.1 Proof of Proposition 3.6

We show that for every bundle $B(x)$, as in Construction 3.5, there exists a consistency test T that, for every $\varepsilon \ge 1/\mathsf{polylog}(\ell)$, 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 $\Pr[T^w(0) = 0] \ge 2/3$.
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 $\Pr[T^w(i) = 0] \ge 2/3$.

Let $\varepsilon \ge 1/\mathsf{polylog}(\ell)$, and assume without loss of generality that $\varepsilon < \delta(\mathsf{ECC})/2$.[24] 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.

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

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 3.6 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 $\Pr[V_i^w = 0] \geq 9/10 \cdot 9/10$.

B.2 Proof of Proposition 3.11

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) universal-LTC_ε $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n)}$ 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 3.5.

Recall that in Construction 3.5 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 3.4, for every $S' \in \mathcal{S}^{(m-\tau)}$

there exist a PCPP oracle $\xi_{S'}$, as required in Construction 3.5, of length $\widetilde{O}(n)$. We obtain the code $C : \{0,1\}^k \to \{0,1\}^{\widetilde{O}(n)}$ given by

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

We show that C is a universal-LTC_ε 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 3.6) 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 $\big(\widetilde{\mathsf{LC}}(x|_{S'}\big)_{S' \in \mathcal{S}^{(m-\tau)}}$, and (3) the purported PCPs of proximity $\big(\widetilde{\xi}_{S'}(x)\big)_{S' \in \mathcal{S}^{(m-\tau)}}$. Note that we use $\widetilde{z}$ to denote a string that is allegedly equal to z. The tester T_f operates as follows:

1. 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 3.6, 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 3.11.

B.3 Proof of Claim 4.3.1

We show that if w is ε-far from the subcode $\{C(x) : \varphi(x) = 1\}$, then for every alleged MAP proof $\widetilde{\pi}_\varphi$, it holds that $Pr[V^w_\varphi(\widetilde{\pi}_\varphi) = 0] \geq 2/3$. Assume, without loss of generality, that $\varepsilon < 1/3$. By Proposition 3.6, the consistency test (Step 2 of V_φ) rejects with probability $2/3$ unless there exists $x \in \{0,1\}^k$ such that: (1) the input w is ε-close to the codeword $C(x)$, and (2) for every $q \in Q$, the purported function $\widetilde{X}_q$ in w is ε-close to $\widehat{X}_q$, the low-degree extension of x to $\mathbb{F}_q$. Note that, in particular, the polynomial $\widehat{X}_q$ takes *binary* values over H^m (i.e., encodes a binary assignment). Since w is ε-far from the subcode $\{C(x) : \varphi(x) = 1\}$, this implies that the assignment x does not satisfy φ. In addition, we may also assume that for every $q \in Q$ the purported proof $\widetilde{\pi}_{\varphi,q}$ satisfies

$$\sum_{z_1,\ldots,z_\ell \in H} \widetilde{\pi}_{\varphi,q}(z_1,\ldots,z_\ell) \equiv n \pmod q,$$

since otherwise the verifier rejects in Step 1.

On the other hand, observe that there exists $q^* \in Q$ such that

$$\sum_{z_1,\ldots,z_\ell \in H} \pi_{\varphi,q^*}(z_1,\ldots,z_\ell) \not\equiv n \pmod{q^*}.$$

To see this, first recall that $\sum_{z_1,\ldots,z_\ell \in H} \pi_{\varphi,q}(z_1,\ldots,z_\ell)$ counts the number of clauses that the assignment satisfies, modulo q. Note that since the assignment is binary, then $\sum_{z_1,\ldots,z_\ell \in H} \pi_{\varphi,q}(z_1,\ldots,z_\ell) \leq n$, where the summation is over the integers, and that $\prod_{q \in Q} q > n$. Thus, if $\sum_{z_1,\ldots,z_\ell \in H} \pi_{\varphi,q}(z_1,\ldots,z_\ell)$ is congruent to n for all $q \in Q$, then by the Chinese remainder theorem, the assignment satisfies all n constraints, in contraction to our assumption.

Therefore, the ℓ-variate individual degree $d^2t + d$ polynomials $\pi_{\varphi,q}$ and $\widetilde{\pi}_{\varphi,q}$ are not identical, and so, by the Schwartz-Zippel Lemma, they disagree on a randomly chosen point with probability at least $1 - \frac{(d^2t+d)\ell}{\mathbb{F}_q} \geq 9/10$.

To complete the argument, note that the (amplified) self-correctability of low-degree polynomials guarantees that every location in $\widehat{X}_q$ can be reconstructed from $\widetilde{X}_q$ with probability $1 - 1/10|H|^{m-\ell}$. Therefore, all points are read correctly with probability at least $9/10$, and thus, with probability $9/10 \cdot 9/10$, the verifier rejects (in Step 3) when checking whether $\pi_{\varphi,q}(r_1,\ldots,r_\ell)$ equals $\widetilde{\pi}_{\varphi,q}(r_1,\ldots,r_\ell)$.

B.4 Proof of Claim 4.10.1

We show that if $x \notin \Pi_\varphi$, then there exists $q \in Q$ such that $\Pr_{(J_q,\vec{v}_q)}[\widehat{X}_q(J_q) = \vec{v}_q] \leq (1/2)^S$. Fix $s \in S$. For every $q \in Q$, denote $J_{q,s} = \{\widehat{\varphi}_i(r_1^{(s)},\ldots,r_\ell^{(s)},\vec{z})\}_{\vec{z} \in H^{m-\ell}, i \in [t]}$ and $\vec{v}_{q,s} \in \mathbb{F}_q^{|H|^{m-\ell} \cdot t}$ such that $\vec{v}_{q,s}[\vec{z},i] = \vec{v}_q[s,\vec{z},i]$ for all $\vec{z} \in H^{m-\ell}$ and $i \in [t]$ (recall that $\vec{v}_q[s,z,i]$ allegedly equals $\widehat{X}_q \circ \widehat{\varphi}_i(r_1^{(s)},\ldots,r_\ell^{(s)},\vec{z})$). We first show that there exists $q \in Q$ such that $\Pr_{(J_{q,s},\vec{v}_{q,s})}[\widehat{X}_q(J_{q,s}) = \vec{v}_{q,s}] \leq 1/2$.

Similarly to the case in Theorem 4.3, observe that there exists $q \in Q$ such that $\sum_{z \in H^\ell} \pi_q(z) \not\equiv n \pmod q$, since otherwise, by the Chinese remainder theorem,

$$\sum_{z \in H^m} \sum_{i=1}^{n} \chi_i(z) \cdot \mathsf{c}_i\big(X \circ \varphi_1(z),\ldots,X \circ \varphi_t(z)\big) \equiv n \pmod{\prod_{q \in Q} q},$$

in contradiction to the assumption that $\varphi(x) = 0$; fix such $q \in Q$. Therefore the ℓ-variate individual degree $d^2t + d$ polynomials π_q and π_q differ, and so, by the Schwartz-Zippel Lemma,

$$\Pr_{r_1^{(s)},\ldots,r_\ell^{(s)} \in \mathbb{F}_q}[\pi_q(r_1^{(s)},\ldots,r_\ell^{(s)}) \neq \pi_q(r_1^{(s)},\ldots,r_\ell^{(s)})] \geq 1 - \frac{(d^2t+d)\ell}{\mathbb{F}_q} \geq 1 - \frac{(d^2t+d)m}{\mathbb{F}_q} \geq 9/10.$$

In other words, it holds that $\Pr_{(J_{q,s},\vec{v}_{q,s})}[\widehat{X}_q(J_{q,s}) = \vec{v}_{q,s}] < 1/10$. Finally, since $\{(J_{q,s}, \vec{v}_{q,s})\}_{s\in[S]}$ are independently selected, it holds that

$$\Pr_{(J_q,\vec{v}_q)}[\widehat{X}_q(J_q) = \vec{v}_q] = \left(\Pr_{(J_{q,s},\vec{v}_{q,s})}[\widehat{X}_q(J_{q,s}) = \vec{v}_{q,s}]\right)^S \le \left(\frac{1}{10}\right)^S .$$

This concludes the proof of Claim 4.10.1.