Electronic Colloquium on Computational Complexity, Report No. 142 (2020)

Relaxed Locally Correctable Codes with Improved Parameters

Vahid R. Asadi
vasadi@sfu.ca
Simon Fraser University

Igor Shinkar
ishinkar@sfu.ca
Simon Fraser University

Abstract

Locally decodable codes (LDCs) are error-correcting codes $C \colon \Sigma^k \to \Sigma^n$ that admit a local decoding algorithm that recovers each individual bit of the message by querying only a few bits from a noisy codeword. An important question in this line of research is to understand the optimal trade-off between the query complexity of LDCs and their block length. Despite importance of these objects, the best known constructions of constant query LDCs have super-polynomial length, and there is a significant gap between the best constructions and the known lower bounds in terms of the block length.

For many applications it suffices to consider the weaker notion of *relaxed LDCs (RLDCs)*, which allows the local decoding algorithm to abort if by querying a few bits it detects that the input is not a codeword. This relaxation turned out to allow decoding algorithms with constant query complexity for codes with *almost linear* length. Specifically, [Ben+06] constructed an $O(q)$-query RLDC that encodes a message of length k using a codeword of block length $n = O(k^{1+1/\sqrt{q}})$.

In this work we improve the parameters of [Ben+06] by constructing an $O(q)$-query RLDC that encodes a message of length k using a codeword of block length $O(k^{1+1/q})$. This construction matches (up to a multiplicative constant factor) the lower bounds of [KT00; Woo07] for constant query *LDCs*, thus making progress toward understanding the gap between LDCs and RLDCs in the constant query regime.

In fact, our construction extends to the stronger notion of relaxed locally *correctable* codes (RLCCs), introduced in [GRR18], where given a noisy codeword the correcting algorithm either recovers each individual bit of the codeword by only reading a small part of the input, or aborts if the input is detected to be corrupt.

Keywords: algorithmic coding theory; consistency test using random walk; reed-muller code; relaxed locally decodable codes; relaxed locally correctable codes

ISSN 1433-8092

Contents

1 Introduction

Locally decodable codes (LDCs) are error-correcting codes that admit a decoding algorithm that recovers each specific symbol of the message by reading a small number of locations in a possibly corrupted codeword. More precisely, a locally decodable code $C\colon \mathbb{F}^k \to \mathbb{F}^n$ with local decoding radius $\tau \in [0,1]$ is an error-correcting code that admits a local decoding algorithm $\mathcal{D}_C$, such that given an index $i \in [k]$ and a corrupted word $w \in \mathbb{F}^n$ which is τ-close to an encoding of some message $C(M)$, reads a small number of symbols from w, and outputs M_i with high probability. Similarly, we have the notion of *locally correctable codes (LCCs)*, which are error-correcting codes that not only admit a local algorithm that decode each symbol of the message, but are also required to correct an arbitrary symbol from the entire codeword. Locally decodable and locally correctable codes have many applications in different areas of theoretical computer science, such as complexity theory, coding theory, property testing, cryptography, and construction of probabilistically checkable proof systems. For details, see the surveys [Yek12; KS17] and the references within.

Despite the importance of LDCs and LCCs, and the extensive amount of research studying these objects, the best known construction of constant query LDCs has super-polynomial length $n = \exp(\exp(\log^{\Omega(1)}(k)))$, which is achieved by the highly non-trivial constructions of [Yek08] and [Efr12]. For constant query LCCs, the best known constructions are of exponential length, which can be achieved by some parameterization of Reed-Muller codes. It is important to note that there is huge gap between the best known lower bounds for the length of constant query LDCs and the length of best known constructions. Currently, the best known lower bound on the length of LDCs says that for $q \geq 3$ it must be at least $k^{1+\Omega(1/q)}$, where q stands for the query complexity of the local decoder. See [KT00; Woo07] for the best general lower bounds for constant query LDCs.

Motivated by applications to probabilistically checkable proofs (PCPs), Ben-Sasson, Goldreich, Harsha, Sudan, and Vadhan introduced in [Ben+06] the notion of *relaxed locally decodable codes (RLDCs)*. Informally speaking, a relaxed locally decodable code is an error-correcting code which allows the local decoding algorithm to abort if the input codeword is corrupt, but does not allow it to err with high probability. In particular, the decoding algorithm should always output correct symbol, if the given word is not corrupted. Formally, a code $C\colon \mathbb{F}^k \to \mathbb{F}^n$ is an RLDC with decoding radius $\tau \in [0,1]$ if it admits a relaxed local decoding algorithm $\mathcal{D}_C$ which given an index $i \in [k]$ and a possibly corrupted codeword $w \in \mathbb{F}^n$, makes a small number of queries to w, and satisfies the following properties.

Completeness: If $w = C(M)$ for some $M \in \mathbb{F}^k$, then $\mathcal{D}_C^w(i)$ should output M_i.

Relaxed decoding: If w is τ-close to some codeword $C(M) \in C$, then $\mathcal{D}_C^w(i)$ should output either M_i or a special *abort* symbol with probability at least 2/3.

This relaxation turns out to be very helpful in terms of constructing RLDCs with better block length. Indeed, [Ben+06] constructed of a q-query RLDC with block length $n = k^{1+O(1/\sqrt{q})}$.

The notion of *relaxed LCCs (RLCCs)*, recently introduced in [GRR18], naturally extends the notion of RLDCs. These are error-correcting codes that admit a correcting algorithm that is required to correct every symbol of the codeword, but is allowed to abort if noticing that the given word is corrupt. More formally, the local correcting algorithm gets an index $i \in [n]$, and a (possibly corrupted) word $w \in \mathbb{F}^n$, makes a small number of queries to w, and satisfies the following properties.

Completeness: If $w \in C$, then $\mathcal{D}_C^w(i)$ should output w_i.

Relaxed correcting: If w is τ-close to some codeword $c^* \in C$, then $\mathcal{D}_C^w(i)$ should output either c^*_i or a special *abort* symbol with probability at least 2/3.

Note that if the code C is systematic, i.e., the encoding of any message $M \in \mathbb{F}^k$ contains M in its first k symbols, then the notion of RLCC is stronger than RLDC.

Recently, building on the ideas from [GRR18], [CGS20] constructed RLCCs whose block length matches the RLDC construction of [Ben+06]. For the lower bounds, the only result we are aware of is the work of Gur and Lachish [GL20], who proved that for any RLDC the block length must be at least $n = k^{1+\Omega(1/q^2)}$.

Given the gap between the best constructions and the known lower bounds, it is natural to ask the following question:

$$\boxed{\text{What is the best possible trade-off between the query complexity and the block length of an RLDC?}}$$

In particular, [Ben+06] asked whether it is possible to obtain a q-query RLDC whose block length is strictly smaller than the best known lower bound on the length of LDCs. A positive answer to their question would show a separation between the two notions, thus proving that the relaxation is *strict*. See paragraph *Open Problem* in the end of Section 4.2 of [Ben+06].

In this work we make progress on this problem by constructing a relaxed locally decodable code $C\colon \mathbb{F}^K \to \mathbb{F}^N$ with query complexity $O(q)$ and block length $K^{1+O(1/q)}$. In fact, our construction gives the stronger notion of a relaxed locally correctable code.

Theorem 1 (Main Theorem)**.** *For every $q \in \mathbb{N}$ there exists an $O(q)$-query relaxed locally correctable code $C\colon \{0,1\}^K \to \{0,1\}^N$ with constant relative distance and constant decoding radius, such that the block length of C is*

$$N = q^{O(q^2)} \cdot K^{1+O(1/q)} \ .$$

Therefore, our construction improves the parameters of the $O(q)$-query RLDC construction of [Ben+06] with block length $N = K^{1+O(\sqrt{1/q})}$, and matches (up to a multiplicative factor in q) the lower bound of $\Omega(K^{1+\frac{1}{\lceil q/2\rceil - 1}})$ for the block length of q-query LDCs [KT00; Woo07].

Remark 1.1. In this paper we prove Theorem 1 for a code $C\colon \mathbb{F}^K \to \mathbb{F}^N$ over a large alphabet. Specifically, we show a code $C\colon \mathbb{F}^K \to \mathbb{F}^N$ satisfying Theorem 1, for a finite field $\mathbb{F}$ satisfying $|\mathbb{F}| \geq c_q \cdot K^{1/q}$, for some $c_q \in \mathbb{N}$ that depends only on q.

Using the techniques from [CGS20] it is not difficult to obtain an RLCC over the binary alphabet with almost the same block length. Indeed, this can be done by concatenating our code over large alphabet with an arbitrary binary code with constant rate and constant relative distance. See Section 7 for details.

1.1 Related works

RLDC and RLCC constructions: Relaxed locally decodable codes, were first introduced by [Ben+06], motivated by applications to constructing short PCPs. Their construction has a block length equal to $N = K^{1+O(1/\sqrt{q})}$. Since that work, there were no constructions with better block length, in the constant query complexity regime . Recently, [GRR18] introduced the related notion of relaxed locally correctable codes (RLCCs), and constructed q-query RLCCs with block length $N = \mathrm{poly}(K)$. Then, [CGS20] constructed relaxed locally correctable codes with block length matching that of [Ben+06] (up to a multiplicative constant factor q). The construction of [CGS20] had two main components, that we also use in the current work.

Consistency test using random walk (CTRW)**:** Informally, given a word w, and a coordinate i we wish to correct, CTRW samples a sequence of constraints $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t$ on w, such that the domains of $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$ intersect, with the guarantee that if w is close to some codeword $c^* \in C$, but $w_i \neq c^*_i$, then with high probability w will be far from satisfying at least one of the constraints. In other words, CTRW performs a random walk on the constraints graph and checks if w is consistent with c^* in the i'th coordinate. We introduce this notion in detail in Section 2.1, and prove that the Reed-Muller code admits a CTRW in Section 4.

Correctable canonical PCPPs (ccPCPP): These are PCPP systems for some specified language L satisfying the following properties: (i) for each $w \in L$ there is a unique proof $\pi(w)$ that satisfies the verifier with probability 1, (ii) the verifier accepts with high probability only pairs (x, π) that are close to some $(w, \pi(w))$ for some $w \in L$, i.e., only the pairs where x is close to some $w \in L$, and π is close to $\pi(w)$, and (iii) the set $\{w \circ \pi_w : w \in L\}$ is an RLCC. Canonical proofs of proximity have been studies in [DGG18; Par20]. We elaborate on these constructions in Section 5.

Lower bounds: For lower bounds, the only bound we are aware of is that of [GL20], who proved that any q-query relaxed locally decodable code must have a block length $N \geq K^{1+\Omega(\frac{1}{q^2})}$.

For the strict notion of locally decodable codes, it is known by [KT00; Woo07] that for $q \geq 3$ any q-query LDC must have block length $N \geq \Omega(K^{1+\frac{1}{\lceil q/2 \rceil - 1}})$. For $q = 3$ a slightly stronger bound of $N \geq \Omega(K^2/\log(K))$ is known, and furthermore, for 3-query *linear* LDC the block length must be $N \geq \Omega(K^2/\log\log(K))$ [Woo07]. For $q = 2$ [KW03] proved an exponential lower bound of $N \geq \exp(\Omega(K))$. See also [Des+02; Gol+02; Oba02; WW05; Woo10] for more related work on lower bounds for LDCs.

1.2 Organization

The rest of the paper is organized as follows. In Section 2, we informally discuss the construction and the correcting algorithm. In this discussion we focus on decoding the symbols corresponding to the message, i.e., on showing that the code is an RLDC. Section 3 introduces the formal definitions and notations we will use in the proof of Theorem 1. We present the notion of consistency test using random walk in Section 4, and prove that the Reed-Muller code admits such test. In Section 5 we present the PCPPs we will use in our construction, and state the properties needed for the correcting algorithm. In Section 6 we prove Theorem 1 by proving a composition theorem, which combines the instantiation of the Reed-Muller code with PCPPs from the previous sections.

2 Proof overview

In this section we informally describe our code construction. Roughly speaking, our construction consists of two parts:

The Reed-Muller encoding: Given a message $M \in \mathbb{F}^K$, its Reed-Muller encoding is the evaluation of an m-variate polynomial of degree at most d over $\mathbb{F}$, whose coefficients are determined by the message we wish to encode.

Proofs of proximity: The second part of the encoding consists of the concatenation of PCPPs, each claiming that a certain restriction of the first part agrees with some Reed-Muller codeword.

Specifically, given a message $M \in \mathbb{F}^K$, we first encode it using the Reed-Muller encoding $\mathsf{RM}_{\mathbb{F}}(m, d)$, where m roughly corresponds to the query complexity of our RLDC, and the field is large enough so that the distance of the Reed-Muller code, which is equal to $1 - \frac{d}{|\mathbb{F}|}$, is some constant, say $3/4$. That is, the first part of the encoding corresponds to an evaluation of some polynomial $f\colon \mathbb{F}^m \to \mathbb{F}$ of degree at most d. The second part of the encoding consists of a sequence of PCPPs claiming that the restrictions of a the Reed-Muller part to some carefully chosen planes in $\mathbb{F}^m$ are evaluations of some low-degree polynomial.

The planes we choose are of the form $\mathcal{P}_{\vec{a},\vec{h},\vec{h}'} = \{\vec{a} + t \cdot \vec{h} + s \cdot \vec{h}' : t, s \in \mathbb{F}\}$, where $\vec{a} \in \mathbb{F}^m$, and $\vec{h}, \vec{h}' \in \mathbb{H}^m$ for some $\mathbb{H}$ subfield of $\mathbb{F}$. We will call such planes $\mathbb{H}$***-planes***. In order to obtain the RLDC with the desired parameters, we choose the field $\mathbb{H}$ so that $\mathbb{F}$ is the extension of $\mathbb{H}$ of degree $[\mathbb{F} : \mathbb{H}] = m$. It will be convenient to think of $\mathbb{H}$ as a field and think of $\mathbb{F}$ as a vector space of $\mathbb{H}$ of dimension m (augmented with the multiplicative structure on $\mathbb{F}$). Indeed, the saving in the block length of the RLDC we obtain crucially relies on the fact that we ask for PCPPs for only a small collection of planes, and not all planes in $\mathbb{F}^m$. The actual constraints required to be certified by the PCPPs are slightly more complicated, and we describe the next.

The constraints of the first type correspond to $\mathbb{H}$-planes $\mathcal{P}$ and points $\vec{x} \in \mathcal{P}$. For each such pair $(\mathcal{P}, \vec{x})$ the code will contain a PCPP certifying that (i) the restriction of the Reed-Muller part to $\mathcal{P}$ is close to an evaluation of some polynomial of total degree at most d, (ii) and furthermore, this polynomial agrees with the value of the Reed-Muller part on $\vec{x}$. In order to define it formally, we introduce the following notation.

Notation 2.1. *Let $\mathbb{F}$ be a finite field of size n. Fix $f\colon \mathbb{F}^m \to \mathbb{F}$, a plane $\mathcal{P}$ in $\mathbb{F}^m$, and a point $\vec{x} \in \mathcal{P}$. Denote $f^{(\vec{x})}_{|\mathcal{P}} = f_{|\mathcal{P}} \circ (f(\vec{x}))^{n^2}$. That is, the length of $f^{(\vec{x})}_{|\mathcal{P}}$ is $2 \cdot n^2$, and it consists of $f_{|\mathcal{P}}$ concatenated with n^2 repetitions of $f(\vec{x})$.*

Given the notation above, if f is the first part of the codeword, corresponding to the Reed-Muller encoding of the message, then the PCPP for the pair $(\mathcal{P}, \vec{x})$ is expected to be the proof of proximity claiming that $f^{(\vec{x})}_{|\mathcal{P}}$ is close to the language

$$\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}} = \{Q \circ (Q(\vec{x}))^{(n^2)} : Q \text{ is the evaluation of a degree-}d\text{ polynomial on } \mathcal{P}\} \subseteq \mathbb{F}^{2n^2} \quad . \qquad (1)$$

Note that by repeating the symbol $Q(\vec{x})$ for n^2 times, the definition indeed puts weight 1/2 on the constraint that the input $f_{|\mathcal{P}}$ is close to some low-degree polynomial Q, and puts weight 1/2 of the constraint $f(\vec{x}) = Q(\vec{x})$. In particular, if $f_{|\mathcal{P}}$ is δ-close to some bivariate low degree polynomial Q for some small $\delta > 0$, but $f(\vec{x}) \neq Q(\vec{x})$, then $f_{|\mathcal{P}}$ is at least $(1 - \frac{d}{|\mathbb{F}|} - \delta)/2$-far from any bivariate low degree polynomial on $\mathcal{P}$.

The constraints of second type correspond to $\mathbb{H}$-planes $\mathcal{P}$ and lines $\ell \subseteq \mathcal{P}$. For each such pair $(\mathcal{P}, \ell)$ the code will contain a PCPP certifying that (i) the restriction of the Reed-Muller part to $\mathcal{P}$ is close to an evaluation of some polynomial of total degree at most d, (ii) and furthermore, this polynomial is close to $f_{|\ell}$. (In particular, this implies that $f_{|\ell}$ is close to some low-degree polynomial.)

Next, we introduce the notation analogous to Notation 2.1 replacing the points with lines.

Notation 2.2. *Let $\mathbb{F}$ be a finite field of size n. Fix $f\colon \mathbb{F}^m \to \mathbb{F}$, a plane $\mathcal{P}$ in $\mathbb{F}^m$, and a line $\ell \subseteq \mathcal{P}$. Denote by $f^{(\ell)}_{|\mathcal{P}} = f_{|\mathcal{P}} \circ (f_{|\ell})^n$. That is, the length of $f^{(\ell)}_{|\mathcal{P}}$ is $2 \cdot n^2$, and it consists of $f_{|\mathcal{P}}$ concatenated with n repetitions of $f_{|\ell}$.*

If f is the Reed-Muller part of the codeword, corresponding to the Reed-Muller encoding of the message, then the PCPP for the pair $(\mathcal{P}, \ell)$ is expected to be the proof of proximity claiming that $f^{(\ell)}_{|\mathcal{P}}$ is close to the language

$$\mathsf{RM}^{(\ell)}_{|\mathcal{P}} = \{Q \circ (Q_{|\ell})^n : Q \text{ is the evaluation of some degree-}d\text{ polynomial on } \mathcal{P}\} \subseteq \mathbb{F}^{2n^2} \quad . \qquad (2)$$

Again, similarly to the first part, repeating the evaluation of $Q_{|\ell}$ for n times puts weight 1/2 on the constraint that the input $f_{|\mathcal{P}}$ is a close to some low-degree polynomial Q, and puts weight 1/2 of the constraint $f_{|\ell}$ is close to $Q_{|\ell}$.

With the proofs specified above, we now sketch the local correcting algorithm for the code. Below we only focus on correcting symbols from the Reed-Muller part. Correcting the symbols from the PCPP part follows a rather straightforward adaptation of the techniques from [CGS20], and we omit them from the overview.

Given a word $w \in \mathbb{F}^N$ and an index $i \in [N]$ of w corresponding to the Reed-Muller part of the codeword, let $f \colon \mathbb{F}^m \to \mathbb{F}$ be the Reed-Muller part of w, and let $\vec{x} \in \mathbb{F}^m$ be the input to f corresponding to the index i. The local decoder works in two steps.

Consistency test using random walk: In the first step the correcting algorithm invokes a procedure we call *consistency test using a random walk (*CTRW*)* for the Reed-Muller code. This step creates a sequence of $\mathbb{H}$-planes of length $(m+1)$, where each plane defines a constraint checking that the restriction of w to the plane is low-degree. Hence, we get $m+1$ constraints, each depending on n^2 symbols.

Composition using proofs of proximity: Then, instead of reading the entire plane for each constraint, we use the PCPPs from the second part of the codeword to reduce the arity of each constraint to $O(1)$, thus reducing the total query complexity of the correcting algorithm to $q = O(m)$. That is, for each constraint we invoke the corresponding PCPP verifier to check that the restrictions of f to each of these planes is (close to) a low-degree polynomial. If at least one of the verifiers rejects, then the word f must be corrupt, and hence the correcting algorithm returns $\perp$. Otherwise, if all the PCPP verifiers accept, the correcting algorithm returns $f(\vec{x})$.

In particular, if f is a correct Reed-Muller encoding, then the algorithm will always return $f(\vec{x})$, and the main part of the analysis is to show that if f is close to some $Q^* \in \mathsf{RM}_{\mathbb{F}}(m,d)$, but $f(\vec{x}) \neq Q^*(\vec{x})$, then the correcting algorithm catches an inconsistency, and returns $\perp$ with some constant probability. See Section 6.3 for details.

The key step in the analysis says that if f is close to some codeword $Q^* \in \mathsf{RM}$ but $f(\vec{x}) \neq Q^*(\vec{x})$, then with high probability f will be far from a low degree polynomial on at least one of these planes, where “far” corresponds to the notion of distances defined by the languages $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}$ and $\mathsf{RM}^{(\ell)}_{|\mathcal{P}}$. In particular, if on one of the planes f is far from the corresponding language, then the PCPP verifier will catch this with constant probability, thus causing the correcting algorithm to return $\perp$. We discuss this part in detail below.

It is important to emphasize that the main focus of this work is constructing a correcting algorithm for the Reed-Muller part. Using the techniques developed in [CGS20], it is rather straightforward to design the algorithm for correcting symbols from the PCPPs part of the code. See Section 6.4 for details.

2.1 CTRW on Reed-Muller codes

Below we define the notion of *consistency test using random walk (*CTRW*)* for the Reed-Muller code. This notion is a slight modification of the notion originally defined in [CGS20] for general codes. In this paper we define it only for the Reed-Muller code. Given a word $f \colon \mathbb{F}^m \to \mathbb{F}$ and some $\vec{x} \in \mathbb{F}^m$, the goal of the test is to make sure that $f(\vec{x})$ is consistent with the codeword of Reed-Muller code closest to f. [CGS20] describe a CTRW for the tensor power $C^{\otimes m}$ of an arbitrary codes C with good distance (e.g., Reed-Solomon). The CTRW they describe works by starting from the point we wish to correct, and choosing an axis-parallel line ℓ_1 containing the starting point. The test continues by choosing a sequence of random axis-parallel lines $\ell_2, \ell_3, \dots \ell_t$, such that each ℓ_i intersects the previous one, ℓ_{i-1}, until reaching a uniformly random coordinate of the tensor code. That is, the length of the sequence t denotes the *mixing time of the corresponding random* *walk*. The predicates are defined in the natural way; namely, the test expects to see a codeword of C on each line it reads.

In this work, we present a CTRW for the Reed-Muller code, which is a variant of the CTRW described above. The main differences compared to the description above are that (i) the test chooses a sequence of planes $\mathcal{P}_1, \mathcal{P}_3, \dots \mathcal{P}_t$ (and not lines), (ii) and every two planes intersect on a line (and not on a point). Roughly speaking, the algorithm works as follows.

1. Given a point $\vec{x} \in \mathbb{F}^m$ the test picks a uniformly random $\mathbb{H}$-plane $\mathcal{P}_0$ containing $\vec{x}$.
2. Given $\mathcal{P}_0$, the test chooses a random line $\ell_1 \subseteq \mathcal{P}_0$, and then chooses another random $\mathbb{H}$-plane $\mathcal{P}_1 \subseteq \mathbb{F}^m$ containing ℓ_1.
3. Given $\mathcal{P}_1$, the test chooses a random line $\ell_2 \subseteq \mathcal{P}_1$, and then chooses another random $\mathbb{H}$-plane $\mathcal{P}_2 \subseteq \mathbb{F}^m$ containing ℓ_2.
4. The algorithm continues for some predefined number of iterations, choosing $\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, \dots \mathcal{P}_t$. Roughly speaking, the number of iterations is equal to the mixing time of the corresponding Markov chain. More specifically, the process continues until a uniformly random point in $\mathcal{P}_t$ is close to a uniform point in $\mathbb{F}^m$.
5. The constraints defined for each $\mathcal{P}_i$ are the natural constraints; namely checking that the restriction of f to $\mathcal{P}_i$ is a polynomial of degree at most d.

One of the important parameters, directly affecting the query complexity of our construction is the mixing time of the random walk. Indeed, as explained above, the query complexity of our RLDC is proportional to the mixing time of the random walk. We prove that if $[\mathbb{F} : \mathbb{H}] = m$, then the mixing time is upper bounded by m. In order to prove this we use the following claim, saying that if $\mathbb{F}$ is the field extension of $\mathbb{H}$ of degree m, and $\vec{h}_1, \dots, \vec{h}_m \in \mathbb{H}^m$ and $t_1, \dots, t_m \in \mathbb{F}$ are sampled uniformly, independently from each other, then $\sum_{i=1}^{m} t_i \cdot \vec{h}_i$ is close to a uniformly random point in $\mathbb{F}^m$. See Claim 3.5 for the exact statement.

As explained above, the key step of the analysis is to prove that if f is close to some codeword $Q^* \in \mathsf{RM}$ but $f(\vec{x}) \neq Q^*(\vec{x})$, then with high probability at least one of the predicates defined will be violated. Specifically, we prove that with high probability the violation will be in the following strong sense.

Theorem 2.3 (informal, see Theorem 4.3)**.** *If f is close to some codeword $Q^* \in \mathsf{RM}$ but $f(\vec{x}) \neq Q^*(\vec{x})$, then with high probability*

1. either $f^{(\vec{x})}_{|\mathcal{P}_0}$ is $\Omega(1)$-far from $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}_0}$,

2. or $f^{(\ell_i)}_{|\mathcal{P}_i}$ is $\Omega(1)$-far from $\mathsf{RM}^{(\ell_i)}_{|\mathcal{P}_i}$ for some $i \in [m]$.

Indeed, this strong notion of violation allows us to use the proofs of proximity in order to reduce the query complexity to $O(1)$ queries for each $i \in [m]$. We discuss proofs of proximity next.

2.2 PCPs of proximity and composition

The second building block we use in this work is the notion of *probabilistic checkable proofs of proximity (PCPPs)*. PCPPs were first introduced in [Ben+06] and [DR04]. Informally speaking, a PCPP verifier for a language L, gets an oracle access to an input x and a proof π claiming that x is close to some element of L. The verifier queries x and π in some small number of (random) locations, and decides whether to accept or reject. The completeness and soundness properties of a PCPP are as follows.

Completeness: If $x \in L$, then there exists a proof causing the verifier to accept with probability 1.

Soundness: If x is far from L, then no proof can make the verifier to accept with probability more than 1/2.

In fact, we will use the slightly stronger notion of *canonical PCPP (cPCPP) systems*. These are PCPP systems satisfying the following completeness and soundness properties. For completeness, we demand that for each w in the language there is a unique *canonical* proof $\pi(w)$ that causes the verifier to accept with probability 1. For soundness, the demand is that the only pairs (x, π) that are accepted by the verifier with high probability are those where x is close to some $w \in L$ and π is close to $\pi(w)$. Such proof system have been studies in [DGG18; Par20], who proved that such proof systems exist for every language in $\mathcal{P}$.

Furthermore, for our purposes we will demand a stronger notion of correctable canonical PCPP systems (ccPCPP). These are canonical PCPP systems where the set $\{w \circ \pi^*(w) : w \in L\}$ is a q-query RLCC for some parameter q, with $\pi^*(w)$ denoting the canonical proof for $w \in L$. It was shown in [CGS20] how to construct ccPCPP by combining a cPCPP system with *any* systematic RLCC. Informally speaking, for every $w \in L$, and its canonical proof $\pi(w)$, we define $\pi^*(w)$ by encoding $w \circ \pi(w)$ using a systematic RLCC. The verifier for the new proof system is defined in a straightforward manner. See [CGS20] for details.

The PCPPs we use throughout this work, are the proofs of two types, certifying that

1. $f^{(\vec{x})}_{|\mathcal{P}}$ is close to $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}$ for some plane $\mathcal{P}$ and some $\vec{x} \in \mathcal{P}$, and
2. $f^{(\ell)}_{|\mathcal{P}}$ is close to $\mathsf{RM}^{(\ell)}_{|\mathcal{P}}$ for some plane $\mathcal{P}$ and some line $\ell \subseteq \mathcal{P}$.

Indeed, it is easy to see that the first type of proofs checks that (i) the restriction of f to $\mathcal{P}$ is close to an evaluation of some polynomial Q^* of total degree at most d, (ii) and $f(\vec{x}) = Q^*(\vec{x})$. Similarly, the second type proof certifies that (i) the restriction of f to $\mathcal{P}$ is close to an evaluation of some polynomial Q^* of total degree at most d, (ii) and $f_{|\ell}$ is close to $Q^*_{|\ell}$.

These notions of distance go together well with the guarantees we have for CTRW in Theorem 2.3. This allows us to *compose* CTRW with the PCPPs to obtain a correcting algorithm with query complexity $q = O(m)$. Informally speaking, the composition theorem works as follows. We first run the CTRW to obtain a collection of $m + 1$ constraints on the planes $\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_m$. By Theorem 2.3, we have the guarantee that with high probability either $f^{(\vec{x})}_{|\mathcal{P}_0}$ is $\Omega(1)$-far from $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}_0}$, or $f^{(\ell_i)}_{|\mathcal{P}_i}$ is $\Omega(1)$-far from $\mathsf{RM}^{(\ell_i)}_{|\mathcal{P}_i}$ for some $i \in [m]$. Then, instead of actually reading the values of f on all these planes, we run the PCPP verifier on $f^{(\vec{x})}_{|\mathcal{P}_0}$ to check that it is close to $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}_0}$, and running the PCPP verifier on each of the $f^{(\ell_i)}_{|\mathcal{P}_i}$ to check that they are close to $\mathsf{RM}^{(\ell_i)}_{|\mathcal{P}_i}$. Each execution of the PCPP verifier makes $O(1)$ queries to f and to the proof, and thus the total query complexity will be indeed $O(m)$. As for soundness, if $f^{(\vec{x})}_{|\mathcal{P}_0}$ is $\Omega(1)$-far from $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}_0}$, or $f^{(\ell_i)}_{|\mathcal{P}_i}$ is $\Omega(1)$-far from $\mathsf{RM}^{(\ell_i)}_{|\mathcal{P}_i}$ for some $i \in [m]$, then the corresponding verifier will notice an inconsistency with constant probability, causing the decoder to output $\perp$.

We discuss proofs of proximity in Section 5. The composition is discussed in Section 6.

3 Preliminaries

We begin with standard notation. The relative distance between two strings $x, y \in \Sigma^n$ is defined as

$$\mathrm{dist}(x, y) := \frac{|\{i \in [n] : x_i \neq y_i\}|}{n} .$$

If $\mathrm{dist}(x, y) \leq \epsilon$, we say that x is *ϵ-close* to y; otherwise we say that x is *ϵ-far* from y. For a non-empty set $S \subseteq \Sigma^n$ define the distance of x from S as $\mathrm{dist}(x, S) := \min_{y \in S} \mathrm{dist}(x, y)$. If $\mathrm{dist}(x, S) \leq \epsilon$, we say that x is *ϵ-close* to S; otherwise we say that x is *ϵ-far* from S.

We will also need a more general notion of a distance, allowing different coordinates to have different weight. In particular, we will need the distance that gives constant weight to a particular subset of the coordinates, and spreads the rest of the weight uniformly between all coordinates.

Definition 3.1. *Fix $n \in \mathbb{N}$ and an alphabet Σ. For a set $A \subseteq [n]$ define the distance dist_A between two strings $x, y \in \Sigma^n$ as*

$$\mathrm{dist}_A(x, y) = \frac{|\{i \in A : x_i \neq y_i\}|}{2|A|} + \frac{|\{i \in [n] : x_i \neq y_i\}|}{2n} \quad .$$

In particular, if x differs from y on $\delta|A|$ coordinates in A, then $\mathrm{dist}_A(x, y)$ is at least $\frac{\delta}{2} + \frac{\delta|A|}{2n}$.

We define dist_A between a string $x \in \Sigma^n$ and a set $S \subseteq \Sigma^n$ as

$$\mathrm{dist}_A(x, S) = \min_{y \in S} \mathrm{dist}_A(x, y) \quad .$$

Remark 3.2. This definition generalizes the definition of [CGS20] of dist_k for a coordinate $k \in [n]$. Indeed, the notion of dist_k for a coordinate $k \in [n]$ corresponds to the singleton set $A = \{k\}$.

When the set A is a singleton $A = \{k\}$ we will write $\mathrm{dist}_k(x, y)$ to denote $\mathrm{dist}_{\{k\}}(x, y)$, and we will write $\mathrm{dist}_k(x, S)$ to denote $\mathrm{dist}_{\{k\}}(x, S)$.

3.1 Basic coding theory

Let $k < n$ be positive integers, and let Σ be an alphabet. An *error correcting code* $C \colon \Sigma^k \to \Sigma^n$ is an *injective* mapping from messages of length k over the alphabet Σ to codewords of length n. The parameter k is called the *message length* of the code, and n is its *block length* (which we view as a function of k). The *rate* of the code is defined as k/n, and the *relative distance* of the code is defined as $\min_{M \neq M' \in \Sigma^k} \mathrm{dist}(C(M), C(M'))$. We sometimes abuse the notation and use C to denote the set of all of its codewords, i.e., identify the code with $\{C(M) : M \in \Sigma^k\} \subseteq \Sigma^n$.

Linear codes. Let $\mathbb{F}$ be a finite field. A code $C \colon \mathbb{F}^k \to \mathbb{F}^n$ is *linear* if it is an $\mathbb{F}$-linear map from $\mathbb{F}^k$ to $\mathbb{F}^n$. In this case the set of codewords C is a subspace of $\mathbb{F}^n$, and the message length of C is also the dimension of the subspace. It is a standard fact that for any linear code C, the relative distance of C is equal to $\min_{x \in C \setminus \{0^n\}} \mathrm{dist}(x, 0^n)$.

3.2 Reed-Muller codes

Reed-Muller codes [Mul54] are among the most well studied error correcting codes, with many theoretical and practical applications in different areas of computer science and information theory. Let $\mathbb{F}$ be a finite field of order $|\mathbb{F}| = n$, and let d and m be integers. The code $\mathsf{RM}_{\mathbb{F}}(m, d)$ is the linear code whose codewords are the evaluations of polynomials $f \colon \mathbb{F}^m \to \mathbb{F}$ of total degree at most d over $\mathbb{F}$. We will allow ourselves to write $\mathsf{RM}(m, d)$, since the field is fixed throughout the paper. We will also sometimes omit the parameters m and d, and simply write RM, when the parameters are clear from the context.

In this paper we consider the setting of parameters where $d < |\mathbb{F}| = n$. It is well known that for $d < n$ the relative distance of $\mathsf{RM}_{\mathbb{F}}(m, d)$ is $1 - \frac{d}{n}$. The dimension of RM can be computed by counting the number of monomials of total degree at most d. For $d < n$ the number of such monomials is $\binom{d+m}{m} \geq (\frac{d+m}{m})^m > (\frac{d}{m})^m$. Since the length of each codeword is n^m, it follows that the rate of the code is $\frac{\binom{d+m}{d}}{n^m} > (\frac{d}{mn})^m$.

Definition 3.3. *For $\vec{x}, \vec{y} \in \mathbb{F}^m$ denote by $\ell_{\vec{x},\vec{y}}$ the line*

$$\ell_{\vec{x},\vec{y}} = \{\vec{x} + t \cdot \vec{y} : t \in \mathbb{F}\} \ .$$

Also, for $\vec{x}, \vec{y}, \vec{z} \in \mathbb{F}^m$ denote by $\mathcal{P}_{\vec{x},\vec{y},\vec{z}}$ the plane

$$\mathcal{P}_{\vec{x},\vec{y},\vec{z}} = \{\vec{x} + t \cdot \vec{y} + s \cdot \vec{z} : t, s \in \mathbb{F}\} \ .$$

An important property of $\mathsf{RM}(m, d)$ (and multivariate low-degree polynomials, in general) that we use throughout this work is that their restrictions to lines and planes in $\mathbb{F}^m$ are also polynomials of degree at most d. In other words, if $f \in \mathsf{RM}(m, d)$, and ℓ is a line ($\mathcal{P}$ is a plane) in $\mathbb{F}^m$, then the restriction of f to ℓ (or to $\mathcal{P}$) is a codeword of the Reed-Muller code of the same degree and lower dimension.

The following lemma is a standard lemma in the PCP literature, saying that random lines sample well the space $\mathbb{F}^m$.

Lemma 3.4. *Let $\mathbb{F}$ be a finite field. For any subset $A \subseteq \mathbb{F}^2$ of density $\mu = |A|/|\mathbb{F}^2|$, and for any $\epsilon > 0$ it holds that*

$$\Pr_{\vec{x} \in \mathbb{F}^2, \vec{y} \in \mathbb{F}^2} \left[\left| \frac{|\ell_{\vec{x},\vec{y}} \cap A|}{|\ell_{\vec{x},\vec{y}}|} - \mu \right| > \epsilon \right] \leq \frac{1}{|\mathbb{F}|} \cdot \frac{\mu}{\epsilon^2} \ .$$

Proof. For each $t \in \mathbb{F}$, let X_t be an indicator random variable for the event $\vec{x} + t \cdot \vec{y} \in A$. Since each point is a uniform point in the plane, we have $\mathbb{E}[X_t] = \Pr[X_t = 1] = \mu$, Therefore, denoting $X = \sum_{t \in \mathbb{F}} X_t$, it follows that $\mathbb{E}[\ell_{\vec{x},\vec{y}} \cap A] = \mathbb{E}[X] = \mu \cdot |\mathbb{F}|$.

We are interested in bounding the deviation of $X = \sum_t X_t$ from its expectation. We do it by bounding the variance of X. Note first that $\mathbf{Var}[X_t] = \mu - \mu^2 \leq \mu$. By the pairwise independence of the points on a line, it follows that $\mathbf{Var}[X] = \sum_{t \in \mathbb{F}} \mathbf{Var}[X_t] \leq \mu \cdot |\mathbb{F}|$. Therefore, by applying Chebyshev's inequality we get

$$\Pr \left[\left| \frac{|\ell_{\vec{x},\vec{y}} \cap A|}{|\ell_{\vec{x},\vec{y}}|} - \mu \right| > \epsilon \right] = \Pr \left[|X - \mu|\mathbb{F}|| > \epsilon|\mathbb{F}| \right] \leq \frac{\mathbf{Var}[X]}{(\epsilon|\mathbb{F}|)^2} \leq \frac{\mu}{|\mathbb{F}| \cdot \epsilon^2} \ ,$$

as required. ☐

The following claim will be an important step in our analysis.

Claim 3.5. *Let $m \in \mathbb{N}$ be a parameter, let $\mathbb{H}$ be a finite field, and let $\mathbb{F}$ be its extension of degree m. Let $\vec{h}_1, \dots, \vec{h}_m \in \mathbb{H}^m$ and $t_1, \dots, t_m \in \mathbb{F}$ be chosen independently uniformly at random from their domains.*

Then for any set $A \subseteq \mathbb{F}^m$ of size $|A| = \alpha \cdot |\mathbb{F}^m|$ it holds that

$$\Pr \left[\sum_{i=1}^{m} t_i \cdot \vec{h}_i \in A \right] \leq \alpha + 2/\mathbb{H} \ .$$

Proof. In order to prove the claim let us introduce some notation. We write each element in $\mathbb{F}$ as an m-dimensional row vector over $\mathbb{H}$. Also, we will represent an element $\vec{x} \in \mathbb{F}^m$ as a $m \times m$ matrix over $\mathbb{H}$, where the i'th row represents $\vec{x}_i \in \mathbb{F}$, the i'th coordinate of $\vec{x}$. Using this notation we need to prove that the random matrix corresponding to the sum $\sum_{i=1}^m t_i \cdot \vec{h}_i$ is close to a random matrix with entries chosen uniformly from $\mathbb{H}$ independently from each other.

Using the notation above, write each $t_i \in \mathbb{F}$ as a row vector $(t_{i,1}, \ldots, t_{i,m}) \in \mathbb{H}^m$. Observe that for any vector $\vec{h}_i = (\vec{h}_{i,1}, \ldots, \vec{h}_{i,m})^T \in \mathbb{H}^m$ we can represent $t_i \cdot \vec{h}_i \in \mathbb{F}^m$ as the outer product

$$t_i \cdot \vec{h}_i = \begin{bmatrix} t_{i,1} \cdot \vec{h}_{i,1} & t_{i,2} \cdot \vec{h}_{i,1} & \ldots & t_{i,m} \cdot \vec{h}_{i,1} \\ t_{i,1} \cdot \vec{h}_{i,2} & t_{i,2} \cdot \vec{h}_{i,2} & \ldots & t_{i,m} \cdot \vec{h}_{i,2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{i,1} \cdot \vec{h}_{i,m} & t_{i,2} \cdot \vec{h}_{i,m} & \ldots & t_{i,m} \cdot \vec{h}_{i,m} \end{bmatrix} = \begin{bmatrix} \vec{h}_{i,1} \\ \vec{h}_{i,2} \\ \vdots \\ \vec{h}_{i,m} \end{bmatrix} \cdot \begin{bmatrix} t_{i,1} & t_{i,2} & \ldots & t_{i,m} \end{bmatrix}$$

Therefore, the sum $\sum_{i=1}^m t_i \cdot \vec{h}_i$ is represented as

$$\sum_{i=1}^m \begin{bmatrix} \vec{h}_{i,1} \\ \vec{h}_{i,2} \\ \vdots \\ \vec{h}_{i,m} \end{bmatrix} \cdot \begin{bmatrix} t_{i,1} & t_{i,2} & \ldots & t_{i,m} \end{bmatrix} = H \cdot T \ ,$$

where H is the $m \times m$ matrix with $H_{i,j} = \vec{h}_{j,i}$, and T is the $m \times m$ matrix with $T_{i,j} = t_{i,j}$. That is, the sum $\sum_{i=1}^m t_i \cdot \vec{h}_i$ is represented as a product of two uniformly random matrices over $\mathbb{H}$.

Next we show that if $H, T \in \mathbb{H}^{m \times m}$ are chosen uniformly at random and independently, then for any collection A of matrices of size $|A| = \alpha \cdot |\mathbb{H}^{m^2}|$ it holds that $\Pr[H \cdot T \in A] \leq \alpha + 2/\mathbb{H}$. Indeed,

$$\Pr[H \cdot T \in A] \leq \Pr[H \cdot T \in A | H \text{ is invertible}] + \Pr[H \text{ is not invertible}] \ .$$

If H is invertible, then for a uniformly random $T \in \mathbb{H}^{m \times m}$ the probability that $H \cdot T \in A$ is exactly α, and it is easy to check that $\Pr[H \text{ is not invertible}] = \sum_{i=1}^m \frac{1}{|\mathbb{H}|^i} \leq \frac{2}{|\mathbb{H}|}$. □

3.3 Relaxed locally correctable codes

Following the discussion in the introduction, we provide a formal definition of relaxed LCCs, and state some related basic facts and known results.

Definition 3.6 (Relaxed LCC). *Let $C \colon \Sigma^K \to \Sigma^N$ be an error correcting code with relative distance δ, and let $q \in \mathbb{N}$, $\tau_{\text{cor}} \in (0, \delta/2)$,and $\epsilon \in (0,1]$ be parameters. Let $\mathcal{D}$ be a randomized algorithm that gets an oracle access to an input $w \in \Sigma^n$ and an explicit access to an index $i \in [n]$. We say that $\mathcal{D}$ is a q-***query relaxed local correction algorithm*** for C with correction radius τ_{cor} and soundness ϵ if for all inputs the algorithm $\mathcal{D}$ reads explicitly the coordinate $i \in [N]$, reads at most q (random) coordinates in w, and satisfies the following conditions.*

1. *For every $w \in C$, and every coordinate $i \in [N]$ it holds that* $\Pr[\mathcal{D}^w(i) = w_i] = 1$.
2. *For every $w \in \Sigma^n$ that is τ_{cor}-close to some codeword $c^* \in C$ and every coordinate $i \in [N]$ it holds that* $\Pr[\mathcal{D}^w(i) \in \{c_i^*, \perp\}] \geq \epsilon$, *where $\perp \notin \Sigma$ is a special* abort *symbol.*

*The code C is said to be a $(\tau_{\text{cor}}, \epsilon)$-***relaxed locally correctable code (RLCC)*** with query complexity q if it admits a q-query relaxed local correction algorithm with correction radius τ_{cor} and soundness ϵ.*

Observation 3.7. *Note that for* systematic *codes it is clear from Definition 3.6 that RLCC is a stronger notion than RLDC, as it allows the local correction algorithm not only to decode each symbol of the message, but also each symbol of the codeword itself. That is, any systematic RLCC is also an RLDC with the same parameters.*

Finally, we recall the following theorem of Chiesa, Gur, and Shinkar [CGS20].

Theorem 3.8 ([CGS20])**.** *For any finite field $\mathbb{F}$, and parameters $K, q \in \mathbb{N}$, there exists an explicit construction of a systematic linear code $C_{\mathrm{CGS}} \colon \mathbb{F}^K \to \mathbb{F}^N$ with block length $N = q^{O(\sqrt{q})} \cdot K^{1+O(1/\sqrt{q})}$ and constant relative distance, that is a q-query RLCC with constant correction radius $\tau_{\mathrm{cor}} = \Omega(1)$, and constant soundness $\epsilon = \Omega(1)$.*

3.4 Canonical PCPs of proximity

Next we define the notions of *probabilistically checkable proofs of proximity*, and the variants that we will need in this paper.

Definition 3.9 (PCP of proximity)**.** *A q-query* **PCP of proximity (PCPP) verifier** *for a language $L \subseteq \Sigma^*$ with soundness ϵ_{PCPP} with respect the to proximity parameter ρ, is a polynomial-time randomized algorithm V that receives oracle access to an input $x \in \Sigma^n$ and a proof π. The verifier makes at most q queries to $x \circ \pi$ and has the following properties:*

Completeness: *For every $x \in L$ there exists a proof π such that* $\Pr[V^{x,\pi} = ACCEPT] = 1$.

Soundness: *If x is ρ-far from L, then for every proof π it holds that* $\Pr[V^{x,\pi} = ACCEPT] \leq \epsilon_{PCPP}$.

A *canonical PCPP (cPCPP)* is a PCPP in which every instance in the language has a *canonical* accepting proof. Formally, a canonical PCPP is defined as follows.

Definition 3.10 (Canonical PCPP)**.** *A q-query* **canonical PCPP verifier** *for a language $L \subseteq \Sigma^*$ with soundness ϵ_{PCPP} with respect to proximity parameter ρ, is a polynomial-time randomized algorithm V that gets oracle access to an input $x \in \Sigma^n$ and a proof π. The verifier makes at most q queries to $x \circ \pi$, and satisfies the following conditions:*

Canonical completeness: *For every $w \in L$ there exists a* unique *(canonical) proof $\pi(w)$ for which* $\Pr[V^{w,\pi(w)} = ACCEPT] = 1$.

Canonical soundness: *For every $x \in \Sigma^n$ and proof π such that*

$$\delta(x, \pi) \triangleq \min_{w \in L} \left\{ \max \left(\frac{\mathrm{dist}(x, w)}{n} \, , \, \frac{\mathrm{dist}(\pi, \pi(w))}{len(n)} \right) \right\} > \rho \ , \tag{3}$$

it holds that $\Pr[V^{x,\pi} = ACCEPT] \leq \epsilon_{PCPP}$.

The following result on canonical PCPPs was proved in [DGG18] and [Par20].

Theorem 3.11 ([DGG18; Par20])**.** *Let $\rho > 0$ be a proximity parameter. For every language in $L \in$ **P** there exists a polynomial $len \colon \mathbb{N} \to \mathbb{N}$ and a canonical PCPP verifier for L satisfying the following properties.*

1. *For all $x \in L$ of length $|x| = n$ the length of the canonical proof $\pi(x)$ is $|\pi(x)| = len(n)$.*
2. *The query complexity of the PCPP verifier is $q = O(1/\rho)$.*
3. *The PCPP verifier for L has perfect completeness and soundness $\epsilon = 1/2$ for proximity parameter ρ (with respect to the uniform distance measure).*

Next, we define the stronger notion of *correctable canonical PCPPs (ccPCPP)*, originally defined in [CGS20]. A ccPCPP system is a canonical PCPP system that in addition to allowing the verifier to be able to locally verify the validity of the given proof, it also admits a local correction algorithm that locally corrects potentially corrupted symbols of the canonical proof. Formally, the ccPCPP is defined as follows.

Definition 3.12 (Correctable canonical PCPP)**.** *A language $L \subseteq \Sigma^*$ is said to admit a ccPCPP with query complexity q and soundness ϵ_{PCPP} with respect to the proximity parameter ρ, and correcting soundness ϵ for correcting radius τ_{cor} if it satisfies the following conditions.*

1. *L admits a q-query canonical PCPP verifier for L satisfying the conditions in Definition 3.10 with soundness ϵ_{PCPP} with respect to the proximity parameter ρ.*
2. *The code $\Pi_L = \{w \circ \pi(w) : w \in L\}$ is a $(\tau_{\mathrm{cor}}, \epsilon)$-RLCC with query complexity q, where $\pi(w)$ is the canonical proof for $w \in L$ from Definition 3.10.*

4 Consistency test using random walk on the Reed-Muller code

Below we define the notion of *consistency test using random walk (*CTRW*)*. This notion has been originally defined in [CGS20] for tensor powers of general codes. In this paper we focus on CTRW for the Reed-Muller code.

Informally speaking, a consistency test using random walk for Reed-Muller code $\mathsf{RM} = \mathsf{RM}_{\mathbb{F}}(m,d)$ is a randomized algorithm that gets a word $f \colon \mathbb{F}^m \to \mathbb{F}$, which is close to some codeword $Q^* \in \mathsf{RM}$, and an index $\vec{x} \in \mathbb{F}^m$ as an input, and its goal is to check whether $f(\vec{x}) = Q^*(\vec{x})$. In other words, it checks whether the value of f at $\vec{x}$ is consistent with the close codeword Q^*. Below we formally describe the random process.

Definition 4.1 (Consistency test using $\mathbb{H}$-plane-line random walk on $\mathsf{RM}_{\mathbb{F}}(m,d)$)**.** *Let $\mathbb{H}$ be a field, and let $\mathbb{F}$ be a field extension of $\mathbb{H}$. Let $\mathsf{RM} = \mathsf{RM}_{\mathbb{F}}(m,d)$ be the Reed-Muller code. An r-steps* **consistency test using $\mathbb{H}$-plane-line random walk** *on RM is a randomized algorithm that gets as input the evaluation table of some $f \colon \mathbb{F}^m \to \mathbb{F}$ and a coordinate $\vec{x} \in \mathbb{F}^m$, and works as in Algorithm 1.*

We say that CTRW has perfect completeness and (τ, ρ, ϵ)-robust soundness if it satisfies the following guarantees.
Perfect completeness: *If $f \in \mathsf{RM}$, then $\Pr[\mathsf{CTRW}^f(\vec{x}) = ACCEPT] = 1$ for all $\vec{x} \in \mathbb{F}^m$.*
(τ, ρ, ϵ)**-robust soundness:** *If f is τ-close to some $Q^* \in \mathsf{RM}$, but $f(\vec{x}) \neq Q^*(\vec{x})$, then*

$$\Pr[\mathrm{dist}_{\vec{x}}(f_{|\mathcal{P}_0}, \mathsf{RM}_{|\mathcal{P}_0}) \geq \rho \vee \exists i \in [r] \textit{ such that } \mathrm{dist}_{\ell_i}(f_{|\mathcal{P}_i}, \mathsf{RM}_{|\mathcal{P}_i}) \geq \rho] \geq \epsilon \ .$$

Here $\mathrm{dist}_{\vec{x}}$ and dist_{ℓ_i} are as in Definition 3.1.

Remark 4.2. Note that the soundness condition above is equivalent to checking that

$$\Pr[\mathrm{dist}(f^{(\vec{x})}_{|\mathcal{P}_0}, \mathsf{RM}^{(\vec{x})}_{|\mathcal{P}_0}) \geq \rho \vee \exists i \in [r] \text{ such that } \mathrm{dist}(f^{(\ell_i)}_{|\mathcal{P}_i}, \mathsf{RM}^{(\ell_i)}_{|\mathcal{P}_i}) \geq \rho] \geq \epsilon \ .$$

Next, we show that the Reed-Muller code admits an m-steps consistency test using $\mathbb{H}$-plane-line random walk with constant robust soundness.

Theorem 4.3. *For integer parameters $d, m \geq 2$, let $\mathbb{H}$ be a prime field, and let $\mathbb{F}$ be field extension of $\mathbb{H}$ of degree $[\mathbb{F} : \mathbb{H}] = m$ such that $|\mathbb{F}| \geq 2md$. Denote the size of $\mathbb{F}$ by $n = |\mathbb{F}|$. Let $\mathsf{RM} = \mathsf{RM}_{\mathbb{F}}(m,d)$ be the Reed-Muller code over the field $\mathbb{F}$, so that the distance of the code is $\delta_{\mathsf{RM}} \geq 1 - 1/2m \geq 3/4$. Then, for any $\tau \leq \delta_{\mathsf{RM}}/2$ and $\rho \leq \delta_{\mathsf{RM}}/8$ the m-steps consistency test using $\mathbb{H}$-plane-line random walk on RM has perfect completeness and $(\tau, \rho, \epsilon)-$robust soundness, with $\epsilon = \left(1 - \frac{4}{|\mathbb{F}|}\right)^m - \frac{\tau + \frac{2}{|\mathbb{H}|}}{\delta_{\mathsf{RM}} - 2\rho}$.*

Algorithm 1: $\mathbb{H}$-plane-line CTRW for the $m-$dimensional Reed-Muller code

```
    Input: $f\colon \mathbb{F}^m \to \mathbb{F}$, $\vec{x} \in \mathbb{F}^m$
1   Pick $\vec{h}_0, \vec{h}'_0 \in \mathbb{H}^m$ uniformly at random, and let $\vec{x}_0 = \vec{x}$
2   Let $\mathcal{P}_0 = \mathcal{P}_{\vec{x}_0,\vec{h}_0,\vec{h}'_0}$ be a random $\mathbb{H}$-plane passing through $\vec{x}$
3   for $i = 1$ to $r$ do
4       Sample $s_{i-1}, s'_{i-1} \in \mathbb{F}$ uniformly and independently
5       Let $\vec{x}_i = \vec{x}_{i-1} + s_{i-1} \cdot \vec{h}_{i-1} + s'_{i-1} \cdot \vec{h}'_{i-1}$ be a uniformly random point in $\mathcal{P}_{i-1}$
6       Sample $t_{i-1}, t'_{i-1} \in \mathbb{F}$ uniformly and independently, and let $\vec{h}_i = t_{i-1} \cdot \vec{h}_{i-1} + t'_{i-1} \cdot \vec{h}'_{i-1}$
7       Let $\ell_i = \ell_{\vec{x}_i,\vec{h}_i} = \{\vec{x}_i + t \cdot \vec{h}_i : t \in \mathbb{F}\}$ be a random line in $\mathcal{P}_{i-1}$
8       Pick $\vec{h}'_i \in \mathbb{H}^m$ uniformly at random
9       Let $\mathcal{P}_i = \mathcal{P}_{\vec{x}_i,\vec{h}_i,\vec{h}'_i}$
10  if $f_{|\mathcal{P}_i}$ is an evaluation of a polynomial of total degree at most $d$ for all $0 \le i \le r$ then
11      return ACCEPT
12  else
13      return REJECT
```

Proof. Consider an r-steps consistency test using random walk on RM as in Algorithm 1. By construction, it is clear that whenever $f \in \mathsf{RM}$, the algorithm accepts. It remains to prove the robust soundness of the algorithm. Assume that f is $\tau-$close to some $Q^* \in \mathsf{RM}$. Note that since RM is a linear code, without loss of generality, we may assume that Q^* is all-zeros codeword. Indeed, if f is τ-close to some non-zero codeword Q^*, then we can consider the word $f' = f - Q^*$, which is τ-close to the all-zeros codeword, and behavior of the algorithm on both of these cases are the same. Hence, from now on we will assume that $f(\vec{x}) \neq 0$, and f is $\tau-$close to the all-zeros codeword. Below, we show that when running Algorithm 1 on such f, then for any choice of $\mathcal{P}_0$ we have either

$$\operatorname{dist}_{\vec{x}}(f_{|\mathcal{P}_0}, \mathsf{RM}_{|\mathcal{P}_0}) \geq \rho \tag{4}$$

or

$$\Pr[\exists i \in [r] \text{ s.t. } \operatorname{dist}_{\ell_i}(f_{|\mathcal{P}_i}, \mathsf{RM}_{|\mathcal{P}_i}) \geq \rho] \geq \left(1 - \frac{4}{|\mathbb{F}|}\right)^r - \frac{\tau + \frac{2}{|\mathbb{H}|}}{\delta_{\mathsf{RM}} - 2\rho} \ . \tag{5}$$

It is clear that each of Eq. (4) and Eq. (5) proves Theorem 4.3.

Clearly, if $\operatorname{dist}_{\vec{x}}(f_{|\mathcal{P}_0}, \mathsf{RM}_{|\mathcal{P}_0}) \geq \rho$, then we are done. Hence, let us assume that $\operatorname{dist}_{\vec{x}}(f_{|\mathcal{P}_0}, \mathsf{RM}_{|\mathcal{P}_0}) < \rho$. In particular, since $f(\vec{x}) \neq 0$, $\rho \leq \delta_{\mathsf{RM}}/8$, and $\operatorname{dist}_{\vec{x}}(f_{|\mathcal{P}_0}, \mathsf{RM}_{|\mathcal{P}_0}) < \rho$, it follows that $f_{|\mathcal{P}_0}$ is 2ρ-close to some *non-zero* codeword of $\mathsf{RM}_{|\mathcal{P}_0}$, and hence $f_{|\mathcal{P}_0}$ contains at least $(\delta_{\mathsf{RM}} - 2\rho)n^2$ non-zero entries. For the rest of the proof we focus on proving Eq. (5) assuming that $f_{|\mathcal{P}_0}$ contains at least $(\delta_{\mathsf{RM}} - 2\rho)n^2$ non-zero entries.

In order to prove it, we introduce the events E_i and F_i.

Definition 4.4. *For $i \in [r]$ denote by E_i the event that $f_{|\ell_i}$ has at least $2\rho n$ non-zeros, and $f_{|\mathcal{P}_i}$ has less than $(\delta_{\mathsf{RM}} - 2\rho)n^2$ non-zeros. For $i \in [r]$ denote by F_i the event that $f_{|\ell_i}$ has at least $2\rho n$ non-zeros, and $f_{|\mathcal{P}_i}$ has at least $(\delta_{\mathsf{RM}} - 2\rho)n^2$ non-zeros.*

The following are the key observations about the event E_i

Observation 4.5. *If E_i holds and $\rho \leq \delta_{\mathsf{RM}}/4$, then*

1. $\mathrm{dist}_{\ell_i}(f_{|\mathcal{P}_i}, \mathbf{0}) \geq \rho$*, since* $f_{|\ell_i}$ *has at least* $2\rho n$ *non-zeros.*

2. $\mathrm{dist}_{\ell_i}(f_{|\mathcal{P}_i}, Q) \geq \rho$ *for all* $Q \in \mathsf{RM} \setminus \{0\}$*, since* $f_{|\mathcal{P}_i}$ *has less than* $2\rho n^2$ *non-zeros.*

In particular, if E_i *holds, then* $\mathrm{dist}_{\ell_i}(f_{|\mathcal{P}_i}, \mathsf{RM}_{|\mathcal{P}_i}) \geq \rho$*.*

For each $i \in [r]$ denote

$$\epsilon_i = \Pr[(\wedge_{j=0}^{i-1} F_j) \bigwedge E_i] \quad .$$

Observe that the events corresponding to ϵ_i's are disjoint, and hence

$$\Pr[\exists i \in [r] \text{ s.t. } \mathrm{dist}_{\ell_i}(f_{|\mathcal{P}_i}, \mathsf{RM}_{|\mathcal{P}_i}) \geq \rho] \geq \sum_{i=1}^{r} \epsilon_i \quad .$$

The following two lemmas are the key steps in the proof of Theorem 4.3.

Lemma 4.6. *For a uniformly random point* $\vec{z} \in \mathcal{P}_m$ *we have* $\Pr[f(\vec{z}) \neq 0] \leq \tau + \frac{2}{|\mathbb{H}|}$.

Lemma 4.7. *If* $\mathrm{dist}_{\vec{x}}(f_{|\mathcal{P}_0}, \mathsf{RM}_{|\mathcal{P}_0}) < \rho$*, then* $\Pr[(\wedge_{i=1}^{r} F_i)] > (1 - \frac{4}{|\mathbb{F}|})^r - \sum_{i=1}^{r} \epsilon_i$ *for all* $r \geq 1$*.*

We postpone the proofs of the lemmas for now, and proceed with the proof of Theorem 4.3 assuming the lemmas.

Note that if we choose a uniformly random $\vec{z} \in \mathcal{P}_m$, then

$$\begin{aligned}\Pr[f(\vec{z}) \neq 0] \geq \Pr[(\wedge_{i=1}^{m} F_i) \wedge f(\vec{z}) \neq 0] &= \Pr[(\wedge_{i=1}^{m} F_i)] \cdot \Pr[f(\vec{z}) \neq 0 \mid \wedge_{i=1}^{m} F_i] \\ &\geq \Pr[(\wedge_{i=1}^{m} F_i)] \cdot (\delta_{\mathsf{RM}} - 2\rho) \quad ,\end{aligned}$$

where the last inequality is by noting that if we condition on $\wedge_{i=1}^{m} F_i$, then $f_{|\mathcal{P}_m}$ has at least $(\delta_{\mathsf{RM}} - 2\rho)n^2$ non-zeros, and hence $\Pr[f(\vec{z}) \neq 0 \mid \wedge_{i=1}^{m} F_i] \geq (\delta_{\mathsf{RM}} - 2\rho)$. Therefore, by Lemma 4.6 and Lemma 4.7 it follows that

$$\tau + \frac{2}{|\mathbb{H}|} \geq \Pr[f(\vec{z}) \neq 0] \geq \Pr[(\wedge_{i=1}^{m} F_i)] \cdot (\delta_{\mathsf{RM}} - 2\rho) \geq \left(\left(1 - \frac{4}{|\mathbb{F}|}\right)^m - \sum_{i=1}^{m} \epsilon_i\right) \cdot (\delta_{\mathsf{RM}} - 2\rho) \quad , \qquad (6)$$

and hence

$$\Pr[\exists i \in [m] \text{ s.t. } \mathrm{dist}_{\ell_i}(f_{|\mathcal{P}_i}, \mathsf{RM}_{|\mathcal{P}_i}) \geq \rho] \geq \sum_{i=1}^{m} \epsilon_i \geq \left(1 - \frac{4}{|\mathbb{F}|}\right)^m - \frac{\tau + \frac{2}{|\mathbb{H}|}}{\delta_{\mathsf{RM}} - 2\rho} \quad .$$

This completes the proof of Theorem 4.3. □

We now return to the proof of Lemma 4.6.

Proof of Lemma 4.6. Fix $\vec{x}_0$ in Algorithm 1, and consider the independent choices of $\{s_i, s'_i \in \mathbb{F}\}_{i=0}^{m}$, $\{t_i, t'_i \in \mathbb{F}\}_{i=0}^{m}$, and $\{\vec{h}'_i \in \mathbb{H}^m\}_{i=1}^{m}$.

Note first that for all $i \in [m]$ we have

$$\vec{h}_i = (\prod_{j=0}^{i-1} t_j) \cdot \vec{h}_0 + \sum_{u=0}^{i-1} (t'_u \cdot \prod_{j=u+1}^{i-1} t_j) \cdot \vec{h}'_u \quad .$$

We prove this by induction. Indeed, by Algorithm 1, we have $\vec{h}_1 = t_0 \cdot \vec{h}_0 + t'_0 \cdot \vec{h}'_0$. For the induction step, assume that the equation holds for some $i \in [m-1]$. Then for $i+1$ we have

$$\begin{aligned}
\vec{h}_{i+1} &= t_i \cdot \vec{h}_i + t'_i \cdot \vec{h}'_i \\
&= t_i \cdot \left((\prod_{j=0}^{i-1} t_j) \cdot \vec{h}_0 + \sum_{u=0}^{i-1} (t'_u \cdot \prod_{j=u+1}^{i-1} t_j) \cdot \vec{h}'_u \right) + t'_i \cdot \vec{h}'_i \\
&= (\prod_{j=0}^{i} t_j) \cdot \vec{h}_0 + t_i \left(\sum_{u=0}^{i-1} (t'_u \cdot \prod_{j=u+1}^{i-1} t_j) \cdot \vec{h}'_u \right) + t'_i \cdot \vec{h}'_i \\
&= (\prod_{j=0}^{i} t_j) \cdot \vec{h}_0 + \left(\sum_{u=0}^{i-1} (t'_u \cdot \prod_{j=u+1}^{i} t_j) \cdot \vec{h}'_u \right) + t'_i \cdot \vec{h}'_i \\
&= (\prod_{j=0}^{i} t_j) \cdot \vec{h}_0 + \sum_{u=0}^{i} (t'_u \cdot \prod_{j=u+1}^{i} t_j) \cdot \vec{h}'_u \ ,
\end{aligned}$$

which concludes the induction step. Note that the first equation comes from the definition in Algorithm 1 and second equation follows from the induction hypothesis.

Also, for all $i \in [m]$ we have

$$\vec{x}_i = \vec{x}_0 + \sum_{j=0}^{i-1} s_j \vec{h}_j + \sum_{j=0}^{i-1} s'_j \vec{h}'_j \ .$$

Again, we prove this by induction. By Algorithm 1, we have $\vec{x}_1 = \vec{x}_0 + s_0 \cdot \vec{h}_0 + s'_0 \cdot \vec{h}'_0$. For the induction step, if we assume that the equation holds for some $i \in [m-1]$, then for $i+1$ we have

$$\begin{aligned}
\vec{x}_{i+1} &= \vec{x}_i + s_i \cdot \vec{h}_i + s'_i \cdot \vec{h}'_i \\
&= \left(\vec{x}_0 + \sum_{j=0}^{i-1} s_j \vec{h}_j + \sum_{j=0}^{i-1} s'_j \vec{h}'_j \right) + s_i \cdot \vec{h}_i + s'_i \cdot \vec{h}'_i \\
&= \vec{x}_0 + \sum_{j=0}^{i} s_j \vec{h}_j + \sum_{j=0}^{i} s'_j \vec{h}'_j
\end{aligned}$$

This, completes the induction step. Note that the first equation comes from the definition in Algorithm 1 and second equation follows from the induction hypothesis.

Let $\mathcal{P}_m = \mathcal{P}_{\vec{x}_m, \vec{h}_m, \vec{h}'_m}$ Note that we can sample $\vec{z} \in \mathcal{P}_m$ uniformly by choosing $s_m, s'_m \in \mathbb{F}$, and letting

$\vec{z} = \vec{x}_m + s_m \cdot \vec{h}_m + s'_m \vec{h}'_m$. Therefore,

$$\begin{aligned}
\vec{z} &= \left(\vec{x}_0 + \sum_{i=0}^{m} s_i \vec{h}_i\right) + \left(\sum_{r=0}^{m} s'_r \vec{h}'_r\right) \\
&= \left(\vec{x}_0 + s_0\vec{h}_0 + \sum_{i=1}^{m} s_i \left((\prod_{j=0}^{i-1} t_j)\cdot \vec{h}_0 + \sum_{r=0}^{i-1}(t'_r \cdot \prod_{j=r+1}^{i-1} t_j)\cdot \vec{h}'_r\right)\right) + \left(\sum_{r=0}^{m} s'_r \vec{h}'_r\right) \\
&= \left(\vec{x}_0 + (s_0 + \sum_{i=1}^{m} s_i (\prod_{j=0}^{i-1} t_j))\cdot \vec{h}_0\right) + \sum_{i=1}^{m} s_i \left(\sum_{r=0}^{i-1}(t'_r \cdot \prod_{j=r+1}^{i-1} t_j)\cdot \vec{h}'_r\right) + \left(\sum_{r=0}^{m} s'_r \vec{h}'_r\right) \\
&= \left(\vec{x}_0 + (s_0 + \sum_{i=1}^{m} s_i (\prod_{j=0}^{i-1} t_j))\cdot \vec{h}_0\right) + \sum_{r=0}^{m} \left(\sum_{i=r+1}^{m} (s_i t'_r \cdot \prod_{j=r+1}^{i-1} t_j) + s'_r\right) \vec{h}'_r \ .
\end{aligned}$$

Next, we fix all random choices except for $\{s'_r\}$ and $\{\vec{h}'_r\}$, and apply Claim 3.5. Let A be the set of indices $\vec{x}$ such that $f(\vec{x}) \neq 0$. Since f is τ-close to all-zeros codeword, it immediately follows that $|A| = \tau \cdot |\mathbb{F}^m|$. Since each s'_r is chosen uniformly at random from $\mathbb{F}$, and each $\vec{h}'_r$ is chosen uniformly at random from $\mathbb{H}^m$, by applying Claim 3.5 with respect to A, we have

$$\begin{aligned}
\Pr[f(\vec{z}) \neq 0] &= \Pr\left[\left(\vec{x}_0 + (s_0 + \sum_{i=1}^{m} s_i (\prod_{j=0}^{i-1} t_j))\cdot \vec{h}_0\right) + \sum_{r=0}^{m} \left(\sum_{i=r+1}^{m} (s_i t'_r \cdot \prod_{j=r+1}^{i-1} t_j) + s'_r\right) \vec{h}'_r \in A\right] \\
&\leq \tau + \frac{2}{\mathbb{H}} \ ,
\end{aligned}$$

which completes the proof of Lemma 4.6. ☐

Next we prove Lemma 4.7.

Proof of Lemma 4.7. We lower-bound the value of $\Pr[(\wedge_{i=1}^{r} F_i)]$ by peeling off one F_i at a time. Observe that for every $i \in [r]$ we have

$$\begin{aligned}
\Pr[(\wedge_{j=1}^{i-1} F_j) \wedge \ell_i \text{ has at least } 2\rho n \text{ non-zeros}] &= \Pr[(\wedge_{j=1}^{i-1} F_j) \wedge F_i] + \Pr[(\wedge_{j=1}^{i-1} F_j) \wedge E_i] \\
&= \Pr[(\wedge_{j=1}^{i} F_j)] + \epsilon_i \ . \qquad (7)
\end{aligned}$$

We will use the following claim.

Claim 4.8. *For all $i \in [r]$ if $\rho \leq \delta_{\mathsf{RM}}/8$, then* $\Pr[\ell_i$ *has at least* $2\rho n$ *non-zeros* $\mid \wedge_{j=1}^{i-1} F_j] \geq \left(1 - \frac{4}{|\mathbb{F}|}\right)$.

Proof. The proof is rather immediate from Lemma 3.4. Let $\mathcal{P}_{i-1} = \mathcal{P}_{\vec{x}_{i-1}, h_{i-1}, h'_{i-1}}$ be the plane chosen by Algorithm 1 in the iteration $i-1$. Note that conditioning on $(\wedge_{j=1}^{i-1} F_j)$ implies that $f_{|\mathcal{P}_{i-1}}$ has at least $(\delta_{\mathsf{RM}} - 2\rho)n^2$ non-zeros.[1]

[1]This follows only from conditioning on F_{i-1}, and the other F_j's are irrelevant.

Since ℓ_i is a uniformly random line in $\mathcal{P}_{i-1}$, by Lemma 3.4 it follows that

$$\begin{aligned}\Pr[f_{|\ell_i} \text{ has less than } 2\rho n \text{ non-zeros}] &\leq \frac{1}{|\mathbb{F}|} \cdot \frac{\delta_{\mathsf{RM}} - 2\rho}{(\delta_{\mathsf{RM}} - 2\rho - 2\rho)^2} \\ &\leq \frac{4}{|\mathbb{F}|} \ ,\end{aligned}$$

where the last inequality is by the assumption that $\rho \leq \delta_{\mathsf{RM}}/8$ and $\delta_{\mathsf{RM}} \geq 3/4$. This completes the proof of Claim 4.8. □

By applying Claim 4.8 we get

$$\begin{aligned}\Pr[(\wedge_{j=1}^{i-1} F_j) \wedge \ell_i \text{ has at least } 2\rho n \text{ non-zeros}] &= \Pr[(\wedge_{j=1}^{i-1} F_j)] \cdot \Pr[\ell_i \text{ has at least } 2\rho n \text{ non-zeros} \mid \wedge_{j=1}^{i-1} F_j] \\ &\geq \Pr[(\wedge_{j=1}^{i-1} F_j)] \cdot \left(1 - \frac{4}{|\mathbb{F}|}\right) \ . \qquad (8)\end{aligned}$$

Combining Eq. (7) with Eq. (8) together we get

$$\Pr[(\wedge_{j=1}^{i} F_j)] \geq \Pr[(\wedge_{j=1}^{i-1} F_j)] \cdot \left(1 - \frac{4}{|\mathbb{F}|}\right) - \epsilon_i \ . \qquad (9)$$

By exactly the same argument, using the assumption that $f_{|\mathcal{P}_0}$ contains at least $(\delta_{\mathsf{RM}} - 2\rho)n^2$ non-zeros, it follows that

$$\Pr[F_1] \geq 1 - \frac{4}{|\mathbb{F}|} - \epsilon_1 \ . \qquad (10)$$

The rest of the proof follows by induction, peeling off one F_i at a time, and applying Eq. (9).

$$\begin{aligned}\Pr[(\wedge_{i=1}^{r} F_i)] &\geq \Pr[(\wedge_{i=1}^{r-1} F_i)] \cdot \left(1 - \frac{4}{|\mathbb{F}|}\right) - \epsilon_r \\ &\geq \left(\Pr[(\wedge_{i=1}^{r-2} F_i)] \cdot \left(1 - \frac{4}{|\mathbb{F}|}\right) - \epsilon_{r-1}\right)\left(1 - \frac{4}{|\mathbb{F}|}\right) - \epsilon_r \\ &= \Pr[(\wedge_{i=1}^{r-2} F_i)] \cdot \left(1 - \frac{4}{|\mathbb{F}|}\right)^2 - \left(1 - \frac{4}{|\mathbb{F}|}\right)\epsilon_{r-1} - \epsilon_r \\ &\geq \cdots \\ &\geq \Pr[F_1] \cdot \left(1 - \frac{4}{|\mathbb{F}|}\right)^{r-1} - \sum_{i=1}^{r}\left(1 - \frac{4}{|\mathbb{F}|}\right)^{r-i} \cdot \epsilon_i \\ &\geq \left(1 - \frac{4}{|\mathbb{F}|} - \epsilon_1\right) \cdot \left(1 - \frac{4}{|\mathbb{F}|}\right)^{r-1} - \sum_{i=1}^{r}\left(1 - \frac{4}{|\mathbb{F}|}\right)^{r-i} \cdot \epsilon_i \\ &= (1 - \frac{4}{|\mathbb{F}|})^r - \sum_{i=1}^{r}\left(1 - \frac{4}{|\mathbb{F}|}\right)^{r-i} \cdot \epsilon_i \\ &> (1 - \frac{4}{|\mathbb{F}|})^r - \sum_{i=1}^{r}\epsilon_i \ .\end{aligned}$$

We get that $\Pr[(\wedge_{i=1}^{r} F_i)] > (1 - \frac{4}{|\mathbb{F}|})^r - \sum_{i=1}^{r}\epsilon_i$, which concludes Lemma 4.7. □

5 PCPs of proximity

In this section we explain how to construct PCPP systems for the languages $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}$ and $\mathsf{RM}^{(\ell)}_{|\mathcal{P}}$ defined in Eqs. (1) and (2). Note that since all planes in $\mathbb{F}^m$ are isomorphic, we may think of each $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}$ and $\mathsf{RM}^{(\ell)}_{|\mathcal{P}}$ as the language $\mathsf{RM}_{\mathbb{F}}(2,d)$ of bivariate polynomials of total degree at most d, concatenated with repetitions of their values at $\vec{x}$ and ℓ respectively. Following Notation 2.1 and Notation 2.2, we make the following definition.

Definition 5.1. *Let $\mathbb{F}$ be a field of size $|\mathbb{F}| = n$, and let $f\colon \mathbb{F}^2 \to \mathbb{F}$ be an $\mathbb{F}$-valued function. Let $\vec{x} \in \mathbb{F}^2$ be a point in $\mathbb{F}^2$, $\ell \subseteq \mathbb{F}^2$ be the line $\ell = \ell_{\vec{x}_1,\vec{x}_2} = \{\vec{x}_1 + t \cdot \vec{x}_2 \colon t \in \mathbb{F}\}$ for some $\vec{x}_1, \vec{x}_2 \in \mathbb{F}^2$.*

- *Define $f^{(\vec{x})}$ to be the concatenation of f with n^2 repetitions of the value of f at the point $\vec{x}$, i.e., $f^{(\vec{x})} = f \circ (f(\vec{x}))^{n^2}$.*
- *Define $f^{(\ell)}$ to be the concatenation of f with n repetitions of the restriction of f to line ℓ, i.e., $f^{(\ell)} = f \circ (f_{|\ell})^n$.*

Define $\mathsf{RM}^{(\vec{x})} = \mathsf{RM}^{(\vec{x})}_{\mathbb{F}}(2,d) = \{Q^{(\vec{x})}\colon Q \in \mathsf{RM}_{\mathbb{F}}(2,d)\}$ and $\mathsf{RM}^{(\ell)} = \mathsf{RM}^{(\ell)}_{\mathbb{F}}(2,d) = \{Q^{(\ell)}\colon Q \in \mathsf{RM}_{\mathbb{F}}(2,d)\}$.

Note that given an oracle access to f we can query every coordinate of $f^{(\vec{x})}$ and $f^{(\ell)}$ by querying one coordinate of f. In particular, any PCPP system for $f^{(\vec{x})}$ can be emulated when given access to f without increasing the query complexity of the proof system.

The following observation is immediate from Definition 5.1.

Observation 5.2. *Let $\mathbb{F}$ be a field of size $|\mathbb{F}| = n$, Let $\vec{x} \in \mathbb{F}^2$ be a point in $\mathbb{F}^2$, $\ell \subseteq \mathbb{F}^2$ be the line $\ell = \ell_{\vec{x}_1,\vec{x}_2} = \{\vec{x}_1 + t \cdot \vec{x}_2 \colon t \in \mathbb{F}\}$ for some $\vec{x}_1, \vec{x}_2 \in \mathbb{F}^2$. Then for any $f\colon \mathbb{F}^2 \to \mathbb{F}$ we have*

$$\mathrm{dist}_{\vec{x}}(f, \mathsf{RM}) = \mathrm{dist}(f^{(\vec{x})}, \mathsf{RM}^{(\vec{x})}) \quad \textit{and} \quad \mathrm{dist}_{\ell}(f, \mathsf{RM}) = \mathrm{dist}(f^{(\ell)}, \mathsf{RM}^{(\ell)}) \ .$$

It is clear that the languages $\mathsf{RM}^{(\ell)}$ and $\mathsf{RM}^{(\vec{x})}$ can be solved in polynomial time. Therefore, by Theorem 6.1 in [CGS20], these languages admit a ccPCPP with the appropriate parameters.

Theorem 5.3 (Canonical PCPP for RM)**.** *Let $\mathbb{F}$ be a finite field and let $d \in \mathbb{N}$ be a parameter. Let L be either $\mathsf{RM}^{(\ell)}_{\mathbb{F}}(2,d)$ or $\mathsf{RM}^{(\vec{x})}_{\mathbb{F}}(2,d)$. Then L admits a ccPCPP with the following parameters*

1. *The ccPCPP verifier has perfect completeness and soundness $\epsilon_{PCPP} = 0.5$ for any proximity parameter $\rho > 0$.*
2. *The query complexity of the verifier is $q = O(1/\rho)$.*
3. *The length of canonical proof $\pi(f)$ for $f \in L$ of length n is $len(n) = \mathrm{poly}(n)$.*
4. *The language $\Pi_L = \{f \circ \pi(f) : f \in L\}$ is a $(\tau_{\mathrm{cor}}, \epsilon_{inRLCC})$-RLCC with query complexity $q = O(1)$, constant correction radius $\tau_{\mathrm{cor}} = \Omega(1)$, and constant soundness $\epsilon = \Omega(1)$.*

Informally speaking, in order to prove Theorem 5.3 [CGS20] start with a cPCPP system from Theorem 3.11, and for every $w \in L$, and a canonical proof $\pi(w)$, define a correctable proof $\pi^*(w)$ by encoding $w \circ \pi(w)$ using a systematic RLCC with constant distance and polynomial block length (e.g., the one from [GRR18] or [CGS20]). Since the RLCC is systematic, the encoding is of the form $w \circ \pi(w) \circ \pi'(w)$ for some string $\pi'(w)$ of length $\mathrm{poly}(|w|)$. Then, define the canonical proof to be $\pi^*(w) = \pi(w) \circ \pi'(w)$. It is rather straightforward to define a verifier that will satisfy the requirements of Theorem 5.3. We omit the details from here, and refer the interested reader to Theorem 6.1 in [CGS20].

6 Composition theorem and the local correcting algorithm

In this section we present a composition theorem used to combine the CTRW for Reed-Muller codes with appropriate PCPPs. This composition theorem immediately implies the statement of Theorem 1, albeit for a large alphabet.

6.1 Composition theorem using CTRW

Below we prove that if $\mathsf{RM}_{\mathbb{F}}(m,d)$ admits an m-steps CTRW, then it can be composed with a PCPP system with appropriate parameters to obtain an RLCC with query complexity $O(m)$. The composition theorem that we present is a slightly modified version of the composition theorem presented in [CGS20]. The main difference compared to [CGS20] is that we consider two types of PCPP proofs for RM.

Theorem 6.1 (Composition theorem for Reed-Muller codes)**.** *Let $\mathbb{F}$ be a finite field of size $|\mathbb{F}| = n$, let $m, d \in \mathbb{N}$ be parameters, and let $\delta_{\mathsf{RM}} = 1 - \frac{d}{n}$. Suppose that $\mathbb{H}$ is a subfield of $\mathbb{F}$ such that $[\mathbb{F} : \mathbb{H}] = m$. Consider the following components.*

- *Reed-Muller code $\mathsf{RM}_{\mathbb{F}}(m,d) \colon \mathbb{F}^K \to \mathbb{F}^{n^m}$ that admits an m-steps $\mathbb{H}$-plane-line-CTRW with the following parameters.*
  1. *CTRW has perfect completeness and $(\tau, \rho, \epsilon_{RW})$-robust soundness.*
  2. *The total number of predicates (of both types) defined for the CTRW is at most B.*
- *Canonical PCPP systems for languages of the form $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}$ and $\mathsf{RM}^{(\ell)}_{|\mathcal{P}}$ with the following properties.*
  1. *For each $f^{(\vec{x})}_{|\mathcal{P}} \in \mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}$ the length of the canonical proof is at most $len(2n^2)$.*
  2. *For each $f^{(\ell)}_{|\mathcal{P}} \in \mathsf{RM}^{(\ell)}_{|\mathcal{P}}$ the length of the canonical proof is at most $len(2n^2)$.*
  3. *The verifier has query complexity q_{PCPP}, perfect completeness, soundness $\epsilon_{PCPP} < 1$ for proximity parameter ρ.*
  4. *The codes $\Pi_{\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}} = \{f^{(\vec{x})}_{|\mathcal{P}} \circ \pi(f^{(\vec{x})}_{|\mathcal{P}}) : f^{(\vec{x})}_{|\mathcal{P}} \in \mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}\}$ and $\Pi_{\mathsf{RM}^{(\ell)}_{|\mathcal{P}}} = \{f^{(\ell)}_{|\mathcal{P}} \circ \pi(f^{(\ell)}_{|\mathcal{P}}) : f^{(\ell)}_{|\mathcal{P}} \in \mathsf{RM}^{(\ell)}_{|\mathcal{P}}\}$ are $(2\rho, \epsilon_{inRLCC})$-RLCC with query complexity q_{PCPP}.*

Then, there exists a code $C_{comp} \colon \mathbb{F}^K \to \mathbb{F}^N$ with block length $N \leq n^m + 2B \cdot len(2n^2)$ and relative distance at least $\frac{1}{2}\left(1 - \frac{d}{n}\right)$.

The code C_{comp} is a $(\tau_{\mathrm{cor}}, \epsilon_{RLCC})$-RLCC with query complexity $q_{RLCC} = (m+3) \cdot q_{\mathrm{PCPP}}$, where the decoding radius of C_{comp} is $\tau_{\mathrm{cor}} = \tau/4$ and the soundness is

$$\epsilon_{RLCC} = \min\left(\frac{\epsilon_{RW} \cdot (1 - \epsilon_{PCPP}) \cdot \delta_{\mathsf{RM}}}{2}, \frac{\epsilon_{inRLCC}}{2}\right) .$$

Before proceeding with the proof of Theorem 6.1, we show how it implies Theorem 1.

Proof of Theorem 1. Given a sufficiently large parameter q_{RLCC} specifying the desired query complexity of an RLCC, let $m = \lfloor \frac{q_{RLCC}}{q_{\mathrm{PCPP}}} \rfloor - 1 \geq 2$, and let $d \geq 16^m$. Choose a prime field $\mathbb{H}$, such that $(4d)^{1/m} \leq |\mathbb{H}| \leq 2 \cdot (4d)^{1/m}$. Finally, we let $\mathbb{F}$ be the degree-m extension of $\mathbb{H}$, and let $n = |\mathbb{F}|$. In particular, by the choice of d we have $|\mathbb{H}| \geq 16$, and the relative distance of $\mathsf{RM}_{\mathbb{F}}(m,d)$ is $\delta_{\mathsf{RM}} = 1 - \frac{d}{n} = 1 - \frac{d}{|\mathbb{H}|^m} \geq 1 - \frac{d}{4d} \geq 3/4$.

By Theorem 4.3, the $\mathsf{RM}_{\mathbb{F}}(m,d)$ admits an m-steps $\mathbb{H}$-plane-line-CTRW with perfect completeness and $(\tau,\rho,\epsilon_{RW})$-robust soundness, with $\tau=\delta_{\mathsf{RM}}/2$, $\rho=\delta_{\mathsf{RM}}/8$, and soundness parameter

$$\epsilon_{RW}=\left(1-\frac{4}{|\mathbb{F}|}\right)^m-\frac{\tau+\frac{2}{|\mathbb{H}|}}{\delta_{\mathsf{RM}}-2\rho}>1-\frac{4m}{|\mathbb{F}|}-\frac{\delta/2+1/8}{3\delta/4}\geq 1-\frac{4m}{4\cdot 16^m}-\frac{3/8+1/8}{9/16}\geq\frac{1}{16}-\frac{m}{16^m}=\Omega(1)\enspace .$$

Furthermore, by a simple counting, the total number of predicates (of both types) defined for the CTRW is $B\leq 2n^m\cdot|\mathbb{H}|^{2m}\cdot n^2=2n^{m+4}$.

For the canonical PCPP component, by Theorem 5.3 the ccPCPP system has perfect completeness, constant soundness with respect to $\rho=\delta_{\mathsf{RM}}/8$, query complexity $q_{\text{PCPP}}=O(1/\rho)=O(1/\delta_{\mathsf{RM}})=O(1)$, and the length of the canonical proof is $len(2n^2)=\text{poly}(n)$.

Therefore, by Theorem 6.1 we obtain a q_{RLCC}-query $(\tau_{\text{cor}},\epsilon_{RLCC})$-RLCC $C_{comp}\colon\mathbb{F}^K\to\mathbb{F}^N$ with constant relative distance, $\tau_{\text{cor}}=\Omega(1)$, $\epsilon_{RLCC}=\Omega(1)$, where the message length of the code is $K=\binom{d+m}{m}\geq\left(\frac{d}{m}\right)^m$, and its block length is $N\leq n^m+2B\cdot len(2n^2)\leq n^m+4n^{m+4}\cdot\text{poly}(n)$. By plugging in the parameters we get

$$N=n^{m+O(1)}\leq(2^m\cdot 4d)^{m+O(1)}=(4m\cdot 2^m)^{m+O(1)}\cdot\left(\frac{d}{m}\right)^{m+O(1)}=2^{O(m^2)}\cdot K^{1+O(1/m)}\enspace ,$$

and relative distance is at least $\frac{1}{2}\left(1-\frac{d}{n}\right)\geq 3/8$. This completes the proof of Theorem 1. $\square$

The rest of this section is devoting to the proof of Theorem 6.1.

6.2 Constructing the composed code

Constructing the composed code C_{comp}: Given the components in the statement of Theorem 6.1, the composed code $C_{comp}\colon\mathbb{F}^K\to\mathbb{F}^N$ is obtained by concatenating several repetitions of the Reed-Muller encoding of the message with the canonical proofs of proximity.

Specifically, given a message $M\in\mathbb{F}^K$, we first let $Q_{\mathsf{RM}}=\mathsf{RM}(M)$ be encoding of M using $\mathsf{RM}(m,d)$. The final encoding $C_{comp}(M)$ consists of the following three parts:

$$C_{comp}(M)=\mathsf{RM}^{rep}\circ\Pi_{Point}\circ\Pi_{Line}\enspace ,$$

descried below.

1. RM^{rep} consists of $t=\lceil B\cdot len(2n^2)/n^m\rceil$ repetitions of Q_{RM}, where $t\geq 1$ is the minimal integer so that $t\cdot n^m\geq B\cdot len(2n^2)$. Although these repetitions look rather artificial, they make sure that the Reed-Muller part of the encoding will constitute a constant fraction of the codeword $C_{comp}(M)$.

2. Π_{Point} is the concatenation of proofs of proximity $\pi_{(\mathcal{P},\vec{x})}$ (as per the ccPCPPs in the hypothesis of the theorem) for each $\mathbb{H}$-plane $\mathcal{P}$ and for each point $\vec{x}\in\mathcal{P}$. That is, each such $\pi_{(\mathcal{P},\vec{x})}$ is the canonical proof for the assertion that $(Q_{\mathsf{RM}})^{(\vec{x})}_{|\mathcal{P}}\in\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}}$.

   Note that since $C_{comp}(M)$ contains many copies of Q_{RM}, each $\pi_{(\mathcal{P},\vec{x})}$ is expected to be the canonical proof for all the copies.

3. Π_{Line} is the concatenation of proofs of proximity $\pi_{(\mathcal{P},\ell)}$ (as per the ccPCPPs in the hypothesis of the theorem) for each $\mathbb{H}$-plane $\mathcal{P}$ and for each line $\ell\subseteq\mathcal{P}$. Each such $\pi_{(\mathcal{P},\ell)}$ is the canonical proof for the assertion that $(Q_{\mathsf{RM}})^{(\ell)}_{|\mathcal{P}}\in\mathsf{RM}^{(\ell)}_{|\mathcal{P}}$.

Again, since $C_{comp}(M)$ contains many copies of Q_{RM}, each $\pi_{(\mathcal{P},\ell)}$ is expected to be the canonical proof for all the copies.

Parameters of C_{comp}: Note that the total block length of the encoding is $N = t \cdot n^m + B \cdot len(2n^2) \leq n^m + 2B \cdot len(2n^2)$.

As for the relative distance of C_{comp}, if the relative distance of $\mathsf{RM}_{\mathbb{F}}(m,d)$ is δ_{RM}, then, the relative distance of the RM^{rep} part is also δ_{RM}. Furthermore, since the length of the RM^{rep} part is at least half of the total block length, it follows that the relative distance of C_{comp} is at least $\delta_{\mathsf{RM}}/2$.

6.3 Local correction algorithm for the Reed-Muller part

Below we present the local correcting algorithm for C_{comp} for the RM^{rep} part of the code. Given a word $w \in \mathbb{F}^N$ write $w = f^{rep} \circ \Pi_{Point} \circ \Pi_{Line}$, where f^{rep} is (expected to be) the t copies of some Reed-Muller codeword, and Π_{Point}, Π_{Line} are the proofs as described above.

Let $i \in [N]$ be the coordinate in the RM^{rep} part of the code, which corresponds to some $\vec{x} \in \mathbb{F}^m$ of (one of the copies of) the Reed-Muller encoding. The local correcting algorithm works as follows.

Algorithm 2: Local correcting algorithm for the RM^{rep} part

```
   Input: $w = f^{rep} \circ \Pi_{Point} \circ \Pi_{Line}, i \in [N]$
1  Let $\vec{x} \in \mathbb{F}^m$ be the index corresponding to the $i$'th coordinate of the RM encoding
2  Sample $r \in [t]$ uniformly at random, and let $f \colon \mathbb{F}^m \to \mathbb{F}$ be the substring of $f^{rep}$ corresponding to
     the $r$-th copy of the base codeword
3  Run the $m$-steps $\mathbb{H}$-Line-Plane-CTRW from Algorithm 1 on the input $(f, \vec{x})$
4  Let $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_m$ be the planes sampled by CTRW, and let $\ell_1, \ldots, \ell_m$ be the sampled lines
5  Run the cPCPP verifier on $\pi_{(\mathcal{P}_0,\vec{x})}$ to check that $f^{(\vec{x})}_{|\mathcal{P}_0}$ is $\rho$-close to $\mathsf{RM}^{(\vec{x})}_{|\mathcal{P}_0}$
6  for $j = 1$ to $m$ do
7      Run the cPCPP verifier on $\pi_{(\mathcal{P}_j,\ell_j)}$ to check that $f^{(\ell_j)}_{|\mathcal{P}_j}$ is $\rho$-close to $\mathsf{RM}^{(\ell_j)}_{|\mathcal{P}_j}$
8  if Step 5 accepts and all iterations of Step 7 accept then
9      return $f(\vec{x})$
10 else
11     return $\perp$
```

Query complexity: The total number of queries made in Algorithm 2 is clearly upper bounded by $(m+1) \cdot q_{\text{PCPP}}$ from Lines 5 and 7 of the algorithm.

Proof of correctness: By the description of the algorithm, it is clear that if the input is a non-corrupted codeword, i.e., $w \in C_{comp}$ then for any $\vec{x}^* \in \mathbb{F}^m$, the algorithm always returns the correct answer.

Now, assume that the input $w = f^{rep} \circ \Pi_{Point} \circ \Pi_{Line}$ is τ_{cor}-close to some codeword $W^* \in C_{comp}$, and suppose that the RM^{rep} part of W^* consists of t copies of some degree-d polynomial $Q^* \in \mathsf{RM}$. We will show that $\Pr[\mathcal{D}^w_{Algorithm\ 2}(i) \in \{Q^*(\vec{x}), \perp\}] \geq \epsilon_{RLCC}$.

Note that since w is τ_{cor}-close to W^*, and the length of f^{rep} is at least 1/2 of the total block length, it follows that f^{rep} is $2\tau_{\text{cor}}$-close to the t repetitions of Q^*. Denote W_{close} to be the event that $\text{dist}(f, Q^*) \leq 4\tau_{\text{cor}} = \tau$ for the random copy f in the RM^{rep} part sampled in Line 2 of the algorithm. Then, by Markov's inequality $\Pr[W_{\text{close}}] \geq 1/2$. Therefore,

$$\Pr[\mathcal{D}^w_{Algorithm\ 2}(i) \in \{Q^*(\vec{x}), \perp\}] \geq 1/2 \cdot \Pr[\mathcal{D}^w_{Algorithm\ 2}(i) \in \{Q^*(\vec{x}), \perp\} | W_{\text{close}}] \ .$$

From now on, let us condition on the event W_{close}, and focus on the term $\Pr[\mathcal{D}^w_{Algorithm\ 2}(i) \in \{Q^*(\vec{x}), \perp\} | W_{\text{close}}]$. Furthermore, let us fix the choice of f and consider the following two cases.

Case 1: $f(\vec{x}) = Q^*(\vec{x})$**.** Noting that $\mathcal{D}^w_{Algorithm\ 2}(i)$ always outputs either $f(\vec{x})$ or $\perp$ it follows that by conditioning on such f we have

$$\Pr[\mathcal{D}^w_{Algorithm\ 2}(i) \in \{Q^*(\vec{x}), \perp\} | W_{\text{close}}, f(\vec{x}) = Q^*(\vec{x})] = 1 \ .$$

Case 2: $f(\vec{x}) \neq Q^*(\vec{x})$**.** In this case since CTRW admits $(\tau, \rho, \epsilon_{RW})$-robust soundness, it follows that

$$\Pr[\text{dist}_{\vec{x}}(f_{|\mathcal{P}_0}, \mathsf{RM}_{|\mathcal{P}_0}) \geq \rho \vee \exists j \in [m] \text{ such that } \text{dist}_{\ell_j}(f_{|\mathcal{P}_j}, \mathsf{RM}_{|\mathcal{P}_j}) \geq \rho] \geq \epsilon_{RW} \ .$$

Therefore, when running the cPCPP verifier for which the local view is ρ-far from the corresponding predicate, the verifier will reject with probability at least $1 - \epsilon_{PCPP}$, and hence the decoder will output $\perp$ with the same probability. Therefore, we can lower bound the second term by

$$\Pr[\mathcal{D}^w_{Algorithm\ 2}(i) \in \{Q^*(\vec{x}), \perp\} | W_{\text{close}}, f(\vec{x}) \neq Q^*(\vec{x})] \geq \epsilon_{RW}(1 - \epsilon_{PCPP}) \ .$$

Putting all together, we conclude

$$\Pr[\mathcal{D}^w_{Algorithm\ 2}(i) \in \{Q^*(\vec{x}), \perp\} \geq \frac{\epsilon_{RW} \cdot (1 - \epsilon_{PCPP})}{2} \geq \epsilon_{RLCC} \ .$$

This completes the proof of correctness of the algorithm for the RM^{rep} part of the code.

6.4 Local correction algorithm for the proof part

Next, we present the correction algorithm for the cPCPP proofs part of the code. Let $w = f^{rep} \circ \Pi_{Point} \circ \Pi_{Line} \in \mathbb{F}^N$ be a given word, and let $i \in [N]$ be a coordinate in the Π_{Point} part of the proof. The correction algorithm works as follows. (If i belongs to the Π_{Line} part, the correction algorithm is analogous.)

Algorithm 3: Local correction for the PCPP part Π

```
   Input: $w = f^{rep} \circ \Pi_{Point} \circ \Pi_{Line}, i \in [N]$
1  Sample $r \in [t]$ uniformly at random, and let $f\colon \mathbb{F}^m \to \mathbb{F}$ be the substring of $f^{rep}$ corresponding to
     the $r$-th copy of the base codeword
2  if $i$ is a coordinate in $\Pi_{Point}$ then
3      Let $\mathcal{P}^\star$ and $\vec{x}^\star \in \mathcal{P}^\star$ be the plane and the point such that $i$ is a coordinate of $\pi = \pi_{(\mathcal{P}^\star,\vec{x}^\star)}$
4      Run the cPCPP verifier to check that $\mathrm{dist}(f^{(\vec{x}^\star)}_{|\mathcal{P}^\star}, \mathsf{RM}^{(\vec{x}^\star)}_{|\mathcal{P}^\star}) \leq \rho$
5      if Step 4 rejects then
6          return $\perp$
7  else
8      Let $\mathcal{P}^\star$ and $\ell^\star \subseteq \mathcal{P}^\star$ be the plane and the line such that $i$ is a coordinate of $\pi = \pi_{(\mathcal{P}^\star,\ell^\star)}$
9      Run the cPCPP verifier to check that $\mathrm{dist}(f^{(\ell^\star)}_{|\mathcal{P}^\star}, \mathsf{RM}^{(\ell^\star)}_{|\mathcal{P}^\star}) \leq \rho$
10     if Step 9 rejects then
11         return $\perp$
12 Choose a uniformly random $\vec{x}_0 \in \mathcal{P}^\star$
13 Run the $m$-steps $\mathbb{H}$-Line-Plane-CTRW from Algorithm 1 on the input $(f, \vec{x}_0)$
14 Let $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_m$ be the planes sampled by CTRW, and let $\ell_1, \ldots, \ell_m$ be the sampled lines
15 Run the cPCPP verifier on $\pi_{(\mathcal{P}_0,\vec{x}_0)}$ to check that $f^{(\vec{x}_0)}_{|\mathcal{P}_0}$ is $\rho$-close to $\mathsf{RM}^{(\vec{x}_0)}_{|\mathcal{P}_0}$
16 for $j = 1$ to $m$ do
17     Run the cPCPP verifier on $\pi_{(\mathcal{P}_j,\ell_j)}$ to check that $f^{(\ell_j)}_{|\mathcal{P}_j}$ is $\rho$-close to $\mathsf{RM}^{(\ell_j)}_{|\mathcal{P}_j}$
18 if Steps 15 or 17 reject then
19     return $\perp$
20 if $i$ is a coordinate in $\Pi_{Point}$ then
21     Run the local corrector of the inner ccPCPP on $f^{(\vec{x}^\star)}_{|\mathcal{P}^\star} \circ \pi_{(\mathcal{P}^\star,\vec{x}^\star)}$ to correct $w_i$
22     return the value obtained in Step 21
23 else
24     Run the local corrector of the inner ccPCPP on $f^{(\ell^\star)}_{|\mathcal{P}^\star} \circ \pi_{(\mathcal{P}^\star,\ell^\star)}$ to correct $w_i$
25     return the value obtained in Step 24
```

Query complexity: The total number of queries is upper bounded by (i) q_{PCPP} queries in Step 4 or in Step 9, (ii) at most $(m+1) \cdot q_{\mathrm{PCPP}}$ queries in Steps 15 and 17, and (iii) at most q_{PCPP} queries in Step 21 or Step 24 . Therefore, the total query complexity is upper bounded by $(m+3) \cdot q_{\mathrm{PCPP}}$, as required.

Proof of correctness: By the description of the algorithm, it is clear that if $w \in C_{comp}$, then for any index $i \in [N]$ in the proof part, the algorithm always returns the correct answer w_i.

We assume from now on that the input $w \in \mathbb{F}^N$ is τ_{cor}-close to some codeword $W^* \in C_{comp}$, and suppose that the RM^{rep} part of W^* consists of t copies of some degree-d polynomial $Q^* \in \mathsf{RM}$. As in the previous part, since w is τ_{cor}-close to W^*, and the length of f^{rep} is at least 1/2 of the total block length, it follows that f^{rep} is $2\tau_{\mathrm{cor}}$-close to the t repetitions of Q^*. Therefore, for the random copy f in the RM^{rep} part sampled in Line 1 of the algorithm, we have $\Pr[\mathrm{dist}(f, Q^*) \leq 4\tau_{\mathrm{cor}} = \tau] \geq 1/2$. From now on let us condition on the event $\mathrm{dist}(f, Q^*) \le \tau$.

Let us assume that the coordinate $i \in N$ we wish to decode belongs to some $\pi_{(\mathcal{P}^\star, \vec{x}^\star)}$. The following claim completes the analysis of the correcting algorithm.

Claim 6.2. *If* $\Pr[\mathcal{D}^w_{Algorithm\ 3}(i) = \bot] < \epsilon_{RLCC}$*, then* $\Pr[\mathcal{D}^w_{Algorithm\ 3}(i) = W^*_i] > \frac{\epsilon_{inRLCC}}{2}$.

Proof. Since Algorithm 3 returns $\bot$ with probability less than ϵ_{RLCC} in Step 6, the PCPP verifier for $\pi_{(\mathcal{P}^\star, \vec{x}^\star)}$ in Step 4 accepts with probability at least $1 - \epsilon_{RLCC} > \epsilon_{PCPP}$. Thus, there is some bivariate degree-d polynomial $Q' \colon \mathcal{P}^\star \to \mathbb{F}$ (not necessarily equal to $Q^*_{|\mathcal{P}^\star}$) such that

1. $\mathrm{dist}(f^{(\vec{x}^\star)}_{|\mathcal{P}^\star}, Q'^{(\vec{x}^\star)}_{|\mathcal{P}^\star}) \le \rho$, and hence $\mathrm{dist}(f_{|\mathcal{P}^\star}, Q'_{|\mathcal{P}^\star}) \le 2\rho$,
2. and $\pi_{(\mathcal{P}^\star, \vec{x}^\star)}$ is ρ-close to the canonical proof $\pi(Q'^{(\vec{x}^\star)}_{|\mathcal{P}^\star})$.

Next, we use the assumption that Algorithm 3 returns $\bot$ with probability less than ϵ_{RLCC} in Steps 15 or 17. That is, when running $\mathbb{H}$-plane-line CTRW from a uniformly random $\vec{x}_0 \in \mathcal{P}^\star$, and then running the corresponding cPCPPs, with probability at least $1 - \epsilon_{RLCC}$ all cPCPP verifiers accept. For each $\vec{z} \in \mathcal{P}^\star$ let $p_{\vec{z}}$ be the probability that both Step 15 and Step 17 accept when starting from $\vec{z}$. Then $\mathbb{E}[p_{\vec{z}}] > 1 - \epsilon_{RLCC}$, and hence for at least $(1 - \delta_{\mathsf{RM}}/2)n^2$ starting points $\vec{z} \in \mathcal{P}^\star$ we have $p_{\vec{z}} \ge 1 - \frac{2\epsilon_{RLCC}}{\delta_{\mathsf{RM}}} \ge 1 - \epsilon_{RW}(1 - \epsilon_{PCPP})$. Therefore, by the analysis of the correcting algorithm for the RM^{rep} part, for more than $(1 - \delta_{\mathsf{RM}}/2)n^2$ starting points $\vec{z} \in \mathcal{P}^\star$ it holds that $f(\vec{z}) = Q^*(\vec{z})$. Indeed, by *case 2* of the analysis if $f(\vec{z}) \neq Q^*(\vec{z})$, then $p_{\vec{z}} < 1 - \epsilon_{RW}(1 - \epsilon_{PCPP})$. Therefore, $\mathrm{dist}(f_{|\mathcal{P}^\star}, Q^*_{|\mathcal{P}^\star}) < \delta_{\mathsf{RM}}/2$.

Combining with the conclusion from the previous step that $\mathrm{dist}(f_{|\mathcal{P}^\star}, Q'_{|\mathcal{P}^\star}) \le 2\rho$ it follows that $\mathrm{dist}(Q^*_{|\mathcal{P}^\star}, Q'_{|\mathcal{P}^\star}) < 2\rho + \delta_{\mathsf{RM}}/2 \le \delta_{\mathsf{RM}}$. Thus, since $\mathsf{RM}_{\mathbb{F}}(m, d)$ has distance δ_{RM} we conclude that $Q^*_{|\mathcal{P}} = Q'_{|\mathcal{P}}$.

So far we showed that if Algorithm 3 returns $\bot$ with probability less than ϵ_{RLCC} and f is τ-close to Q^* (which happens with probability at least $1/2$), then $f_{|\mathcal{P}^\star}$ is 2ρ-close to $Q^*_{|\mathcal{P}^\star}$, and $\pi_{(\mathcal{P}^\star, \vec{x}^\star)}$ is ρ-close to $\pi(Q^{*(\vec{x}^\star)}_{|\mathcal{P}^\star})$, the canonical proof of $Q^{*(\vec{x}^\star)}_{|\mathcal{P}^\star}$. Therefore, the local correction algorithm for the inner ccPCPP applied on $(f_{|\mathcal{P}^\star} \circ \pi_{(\mathcal{P}^\star, \vec{x}^\star)})$ in Step 21 returns either W^*_i or $\bot$ with probability at least ϵ_{inRLCC}. Therefore,

$$\begin{aligned}\Pr[\mathcal{D}^w_{Algorithm\ 3}(i) \in \{W^*_i, \bot\}] &\ge \Pr[\text{Step 22 returns } W^*_i \text{ or } \bot | \mathrm{dist}(f, Q^*) \le \tau] \cdot \Pr[\mathrm{dist}(f, Q^*) \le \tau] \\ &\ge \frac{\epsilon_{inRLCC}}{2} \quad ,\end{aligned}$$

as required. □

We proved correctness of the local correction algorithm assuming that the coordinate $i \in N$ we wish to decode belongs to some $\pi_{(\mathcal{P}^\star, \vec{x}^\star)}$. For the case when i belongs to $\pi_{(\mathcal{P}^\star, \ell^\star)}$, the analysis is exactly the same. This concludes the proof of Theorem 6.1.

7 Concluding remarks and open problems

In this paper we constructed an $O(q)$-query RLDC $C \colon \mathbb{F}^K \to \mathbb{F}^N$ with block length $N = q^{O(q^2)} \cdot K^{1+O(1/q)}$, assuming that the field is large enough, namely, assuming that $|\mathbb{F}| \ge c_q \cdot K^{1/q}$. Using standard techniques it is possible to obtain a binary RLDC with similar parameters. This can be done by concatenating our code with an arbitrary binary code with constant rate and constant relative distance. Indeed, this transformation appears in [CGS20, Appendix A], who showed how concatenating CTRW-based RLDC over large alphabet with a good binary code gives a binary RLDC that essentially inherits the block length and the query complexity of the RLDC over large alphabet. Below we provide the proof sketch, explaining how the concatenation works.

Proof sketch. Suppose that we want to construct a short binary RLCC. Let $C_{RLCC} : \mathbb{F}^K \to \mathbb{F}^N$ be the RLCC over some field $\mathbb{F}$ with the desired block length, and let $C_{bin} : \{0,1\}^{K'} \to \{0,1\}^{N'}$ be an error-correcting code with constant rate and constant distance. We also assume that field $\mathbb{F}$ is chosen so that $|\mathbb{F}| = 2^{K'}$. (To satisfy this condition, one can simply set $\mathbb{H}$ to be a field of characteristic 2.) This assumption will allow us to have a bijection between each symbol of $\mathbb{F}$ and binary string of length K'.

We construct the binary concatenated code $C_{concat} : \{0,1\}^{K \cdot K'} \to \{0,1\}^{N \cdot N'}$ as follows. Given a message $M \in \{0,1\}^{K \cdot K'}$, we first convert it to an string in $M' \in \mathbb{F}^K$ in the natural way. Then, we encode M' using C_{RLCC} to obtain a codeword $c^* \in C_{RLCC}$. Finally, we encode each symbol of c^* using C_{bin} to get the final codeword $c \in \{0,1\}^{N \cdot N'}$.

To prove that the concatenated code is an RLCC, Chiesa, Gur, and Shinkar proved in [CGS20, Theorem A.4] that if C_{RLCC} admits an r-steps CTRW with some soundness guarantees, then C_{concat} admits an r-steps CTRW with related soundness guarantees. The CTRW on the concatenated code C_{concat} emulates the CTRW on C_{RLCC} by sampling planes for the CTRW on the Reed-Muller code, and instead of reading the symbols from $\mathbb{F}$, it reads the binary encodings of all symbols belonging to these planes.

Indeed, it is not difficult to see that if C_{RLCC} admits an r-steps CTRW with some soundness guarantees, then so does the concatenated code. We omit the details, and refer the interested reader to Appendix A in [CGS20]. □

We conclude the paper with several open problems we leave for future research.

1. The most fundamental open problem regarding RLDCs/RLCCs is to understand the optimal trade-off between the query complexity of LDCs and their block length in the constant query regime. It is plausible that the lower bound of [GL20] can be improved to $K^{1+\Omega(1/q)}$, although we do not have any evidence for this.

2. As discussed in the intoduction, [Ben+06] asked whether it is possible to prove a separation between LDCs and RLDCs. Understanding the trade-off between the query complexity and the block length is one possible way to show such separation.

3. Another interesting open problem is to construct an RLDC/RLCC with constant rate and small query complexity. In particular, it is plausible that there exist $\text{polylog}(N)$-query RLDCs with $N = O(K)$.

4. Also, it would be interesting to construct RLDCs/RLCCs using high-dimensional expanders [KM17; DK17; Dik+18; KO18]. Since there are several definitions of high-dimensional expanders, it would be interesting to state the sufficient properties of high-dimensional expanders required for RLDCs. We believe this approach can be useful in constructing constant rate RLDCs with small query complexity.

References

[Ben+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. "Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding". In: *SIAM Journal on Computing* 36.4 (2006), pp. 889–974.

[CGS20] Alessandro Chiesa, Tom Gur, and Igor Shinkar. "Relaxed Locally Correctable Codes with Nearly-Linear Block Length and Constant Query Complexity". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. 2020, pp. 1395–1411.

[DGG18] Irit Dinur, Oded Goldreich, and Tom Gur. *Every set in P is strongly testable under a suitable encoding*. Tech. rep. Available at `https://eccc.weizmann.ac.il/report/2018/050`. 2018.

[DK17] I. Dinur and T. Kaufman. "High Dimensional Expanders Imply Agreement Expanders". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. 2017, pp. 974–985.

[DR04] Irit Dinur and Omer Reingold. "Assignment Testers: Towards a Combinatorial Proof of the PCP Theorem". In: *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*. FOCS 2004. 2004, pp. 155–164.

[Des+02] A. Deshpande, R. Jain, T. Kavitha, S. V. Lokam, and J. Radhakrishnan. "Better lower bounds for locally decodable codes". In: *Proceedings 17th IEEE Annual Conference on Computational Complexity*. 2002, pp. 184–193.

[Dik+18] Yotam Dikstein, Irit Dinur, Yuval Filmus, and Prahladh Harsha. "Boolean Function Analysis on High-Dimensional Expanders". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*. Leibniz International Proceedings in Informatics (LIPIcs). 2018, 38:1–38:20.

[Efr12] Klim Efremenko. "3-Query Locally Decodable Codes of Subexponential Length". In: *SIAM J. Comput.* 41.6 (2012), pp. 1694–1703.

[GL20] Tom Gur and Oded Lachish. "A Lower Bound for Relaxed Locally Decodable Codes". In: *31st ACM-SIAM Symposium on Discrete Algorithms*. SODA 2020. 2020.

[GRR18] Tom Gur, Govind Ramnarayan, and Ron D. Rothblum. "Relaxed Locally Correctable Codes". In: *9th Innovations in Theoretical Computer Science Conference*. ITCS '18. 2018, 27:1–27:11.

[Gol+02] O. Goldreich, H. Karloff, L. J. Schulman, and L. Trevisan. "Lower bounds for linear locally decodable codes and private information retrieval". In: *Proceedings 17th IEEE Annual Conference on Computational Complexity*. 2002, pp. 175–183.

[KM17] Tali Kaufman and David Mass. "High Dimensional Random Walks and Colorful Expansion". In: *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Leibniz International Proceedings in Informatics (LIPIcs). 2017, 4:1–4:27.

[KO18] Tali Kaufman and Izhar Oppenheim. "High Order Random Walks: Beyond Spectral Gap". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*. Leibniz International Proceedings in Informatics (LIPIcs). 2018, 47:1–47:17.

[KS17] Swastik Kopparty and Shubhangi Saraf. "Local Testing and Decoding of High-Rate Error-Correcting Codes". In: *Electronic Colloquium on Computational Complexity (ECCC)* (2017).

[KT00] Jonathan Katz and Luca Trevisan. "On the Efficiency of Local Decoding Procedures for Error-Correcting Codes". In: *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*. 2000, pp. 80–86.

[KW03] Iordanis Kerenidis and Ronald de Wolf. "Exponential lower bound for 2-query locally decodable codes via a quantum argument". In: *Journal of Computer and System Sciences*. 2003, pp. 106–115.

[Mul54] David E Muller. "Application of Boolean algebra to switching circuit design and to error detection". In: *Transactions of the IRE professional group on electronic computers* (1954).

[Oba02] Kenji Obata. “Optimal Lower Bounds for 2-Query Locally Decodable Linear Codes”. In: *Randomization and Approximation Techniques in Computer Science*. 2002, pp. 39–50.

[Par20] Orr Paradise. “Smooth and Strong PCPs”. In: *Proceedings of the 11th Innovations in Theoretical Computer Science Conference*. ITCS 2020. 2020.

[WW05] Stephanie Wehner and Ronald de Wolf. “Improved Lower Bounds for Locally Decodable Codes and Private Information Retrieval”. In: *Proceedings of the 32nd International Conference on Automata, Languages and Programming*. ICALP’05. 2005, 1424—1436.

[Woo07] David Woodruff. *New Lower Bounds for General Locally Decodable Codes*. Tech. rep. Available at `https://eccc.weizmann.ac.il/report/2007/006/`. 2007.

[Woo10] David P. Woodruff. “A Quadratic Lower Bound for Three-Query Linear Locally Decodable Codes over Any Field”. In: *Proceedings of the 14th International Workshop on Randomized Techniques in Computation*. RANDOM 10. 2010, pp. 766–779.

[Yek08] Sergey Yekhanin. “Towards 3-query locally decodable codes of subexponential length”. In: *J. ACM* 55.1 (2008), 1:1–1:16.

[Yek12] Sergey Yekhanin. “Locally Decodable Codes”. In: *Foundations and Trends in Theoretical Computer Science* 6.3 (2012), pp. 139–255.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il