Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 118 (2021)

Efficient multivariate low-degree tests via interactive oracle proofs of proximity for polynomial codes

Daniel Augot*[1,2] Sarah Bordage†[2,1] Jade Nardi‡[1,2]

October 5, 2022

Abstract

We consider the proximity testing problem for error-correcting codes which consist in evaluations of multivariate polynomials either of bounded individual degree or bounded total degree. Namely, given an oracle function $f\colon L^m \to \mathbb{F}_q$, where $L \subset \mathbb{F}_q$, a verifier distinguishes whether f is the evaluation of a low-degree polynomial or is far (in relative Hamming distance) from being one, by making only a few queries to f. This topic has been studied in the context of locally testable codes, interactive proofs, probalistically checkable proofs, and interactive oracle proofs. We present the first interactive oracle proofs of proximity (IOPP) for tensor products of Reed-Solomon codes (evaluation of polynomials with bounds on individual degrees) and for Reed-Muller codes (evaluation of polynomials with a bound on the total degree) that simultaneously achieve logarithmic query complexity, logarithmic verification time, linear oracle proof length and linear prover running time.

Such low-degree polynomials play a central role in constructions of probabilistic proof systems and succinct non-interactive arguments of knowledge with zero-knowledge. For these applications, highly-efficient multivariate low-degree tests are desired, but prior probabilistic proofs of proximity required super-linear proving time. In contrast, for multivariate codes of length N, our constructions admit a prover running in time linear in N and a verifier which is logarithmic in N.

Our constructions are directly inspired by the IOPP for Reed-Solomon codes of [Ben-Sasson *et al.*, ICALP 2018] named "FRI protocol". Compared to the FRI protocol, our IOPP for tensor products of Reed-Solomon codes achieves the same efficiency parameters. As for Reed-Muller codes, for fixed constant number of variables m, the concrete efficiency of our IOPP for Reed-Muller codes compares well, all things equal.

*Daniel.Augot@inria.fr
†sarah.bordage@lix.polytechnique.fr
‡jade.nardi@inria.fr

ISSN 1433-8092

1 Introduction

1.1 Low-degree tests and proofs of proximity

Let $\mathbb{F}_q$ be a finite field of size q. Any function $f\colon \mathbb{F}_q^m \to \mathbb{F}_q$ can be written as a polynomial of individual degrees at most $q-1$, hence a polynomial of total degree $\leqslant m(q-1)$. The problem of *low-degree testing* can be formulated as follows. Given a proximity parameter $\delta \in (0,1)$ and oracle access to a function $f\colon \mathbb{F}_q^m \to \mathbb{F}_q$ (as a table of values), check with a few queries whether f is a polynomial function of low degree compared to q, or δ-far in relative Hamming distance from being low-degree. The main focus of this paper is the problem of low-degree testing applied to a function $f\colon L^m \to \mathbb{F}_q$ with $L \subset \mathbb{F}_q$. Multivariate low-degree tests fall into two flavours, depending on whether one requires a bound on the total degree or the individual degree. In the former case, the low-degree test can be considered as a proximity test to a Reed-Muller code. In the latter case, it corresponds to a proximity test to the m-wise tensor product of a Reed-Solomon code. See Section 2 for formal definitions of those codes.

Low-degree tests have been the subject of a substantial body of research during the past four decades. Indeed, design and better analysis of low-degree tests have gone hand in hand with the construction of efficient probabilistically checkable proofs (PCPs), interactive proofs (IPs) and locally testable codes (LTCs). One motivation for designing probabilistic proof systems with low communication complexity, fast generation and sublinear verification is the application to verifiable computation. In [BBHR18], the authors point out that a subsequent bottleneck of PCP-based proof systems is that of computing solutions to the low-degree testing problem for multivariate polynomials. A few years ago, [BCS16, RRR16] introduced interactive oracle proofs (IOPs), which generalize both PCPs, IPs and interactive PCPs [KR08] and open a new large design space. On the contrary of known PCPs constructions, it turns out that the IOP model enables the design of proofs systems that are efficient enough for practical applications of zero-knowledge proofs and schemes for delegated computation. Indeed, highly-efficient IOPs lead to efficient succinct transparent non-interactive arguments [AHIV17, BCG+17, BBHR18, BBHR19, KPV19, BCR+19, BCG20] with real-world deployments [BBHR19, Sta21]. Interactive oracle proofs of proximity (IOPP) are the natural generalization of probabilistically checkable proofs of proximity (PCPP) [DR04, BGH+04] to the IOP model. Several of the aforementioned constructions crucially rely on a *prover-efficient* IOPP for Reed-Solomon codes (see Definition 1) which the authors of [BBHR18] named FRI protocol. Improved soundness analysis of the FRI protocal appear in subsequent works [BKS18, BGKS20, BCI+20]. While multivariate low degree tests have been extensively studied in the PCPP model, they have not been the subject of any direct construction in the IOPP model.

1.2 Interactive oracle proof of proximity for a code

In this work, we will consider linear codes C with evaluation domain D of size $n = |D|$ and alphabet $\mathbb{F}_q$ (i.e., $C \subseteq \mathbb{F}_q^D$). An IOPP $(\mathcal{P}, \mathcal{V})$ for a code C is a pair of probabilistic algorithms, $\mathcal{P}$ is designated as *prover* and $\mathcal{V}$ as *verifier*.

The IOPP $(\mathcal{P}, \mathcal{V})$ has round complexity $r(n)$ if the prover and the verifier interact over at most $r(n)$ rounds. At each round, the verifier sends a message to the prover, and the prover answers with an oracle. We denote by $\langle \mathcal{P} \leftrightarrow \mathcal{V} \rangle \in \{\mathsf{accept}, \mathsf{reject}\}$ the output of $\mathcal{V}$ after interacting with $\mathcal{P}$. The notation $\mathcal{V}^f(C)$ means that f is given as an oracle input to $\mathcal{V}$, while $\mathcal{P}(C, f)$ means that the prover has acess to full codeword. Both know the code C.

Definition 1 (IOPP for a code C)**.** *We say that a pair of probabilistic algorithms $(\mathcal{P}, \mathcal{V})$ is an IOPP system for a code C with* soundness error $s\colon (0,1] \to [0,1]$ *if the following two conditions hold:*

Perfect completeness: *If $f \in C$, then* $\Pr[\langle \mathcal{P}(C,f) \leftrightarrow \mathcal{V}^f(C)\rangle = \textsf{accept}] = 1$.

Soundness: *For any function $f \in \mathbb{F}_q^D$ such that $\delta := \Delta(f, C) > 0$ and any unbounded malicious prover $\mathcal{P}^*$,* $\Pr[\langle \mathcal{P}^* \leftrightarrow \mathcal{V}^f(C)\rangle = \textsf{accept}] \leqslant s(\delta)$.

The IOPP is *public-coin* if verifier's messages are generated by public randomness and, in particular, queries can be performed after the end of the interaction with the prover. Throughout this paper, we will consider arithmetic complexities, and we assume each arithmetic operation performed in $\mathbb{F}_q$ takes constant time. Relevant measures for an IOPP system are the following. The alphabet of the IOPP we consider will be a finite field $\mathbb{F}_q$. The total number of field elements of all the oracles built by the prover during the interaction is the proof length $l(n)$ of the IOPP. The query complexity $q(n)$ is the total number of symbols queried by the verifier to both the purported codeword f and the oracles sent by the prover during the interaction. The prover complexity $t_p(n)$ is the time needed to generate prover messages. The verifier complexity $t_v(n)$ is the time spent by the verifier to make her decision when queries and query-answers are given as inputs.

1.3 Contributions and outline

As mentioned above, the focus of the present paper is to tackle the low-degree testing problem for an oracle function $f\colon L^m \to \mathbb{F}_q$ and a degree $d < |L|$. Specifically, we propose two direct constructions: the first is an IOPP for the tensor product of Reed-Solomon codes, the second an IOPP for Reed-Muller codes. The alphabets $\mathbb{F}_q$ which we consider admit either smooth multiplicative subgroups or smooth affine subspaces, where smooth means that the size of the set is a power of a small fixed integer.

Our two IOPPs are generalizations of the FRI protocol [BBHR18] to the multivariate case. We construct an IOP of Proximity for tensor products of Reed-Solomon codes which has the same efficiency parameters than the FRI protocol, namely a strictly linear-time prover and a strictly logarithmic-time verifier (with respect to the block length of the code). In particular, query complexity is logarithmic in the degree bound d. Previous low-degree tests required the verifier to query a number of field elements linear in d. Our IOP of Proximity for Reed-Muller codes share the asymptotic complexities when the number of variables m is a constant independent of the block length.

Since our constructions are explicit, all efficiency measures of the two IOPPs are explicitly presented. These parameters match the IOPP for Reed-Solomon codes of [BBHR18], from which they are inspired (see Figure 3 and Figure 4). Concerning applications to IOP constructions, having a constant number of variables m can be relevant. Indeed, linear-size IOPs have already been constructed from m-wise tensor product codes [BCG20] and m were a fixed integer there. For Reed-Muller codes and unlike previous works, we are able to consider a support L^m where $L \subset \mathbb{F}_q$ can be much smaller than $\mathbb{F}_q$. We think that allowing smaller support might give more flexibility in the design of proof systems.

The organization of the paper is the following. Basic definitions and notations are given in Section 2. In Section 3, we define generic folding operators, which allow to reduce the initial proximity testing problem to a constant-size problem by a divide-and-conquer procedure. Then, a generic construction of an IOPP based on such folding operators is presented. The main purpose of Section 3 is to provide once and for all a unified soundness analysis of IOPP constructions which are based on properties of folding operators. This soundness analysis can be applied to the two explicit constructions of IOPPs we give in the present work, and generalizes the analyses of [BBHR18, BN20]. Section 4 provides technical lemmas about decomposition of multivariate polynomials and multivariate interpolation complexities. In Section 5, we study a special case of worst-case to average-case reduction of distance for linear subspaces, which will be used in our soundness analyses. In the last sections of the paper, we instantiate the generic construction of Section 3. Section 6 gives an algebraic setting shared by IOPs of Proximity for polynomial codes. In Sections 7 and 8, we provide IOPs of Proximity for tensor products of Reed-Solomon codes. The approach pursued in Section 8 leads to improvements over the one of Section 7. Our IOP of Proximity for Reed-Muller codes is constructed in Section 9, and is similar in spirit with the one of Section 7.

1.4 Related work and comparisons

Proximity problem for tensor product of Reed-Solomon codes Low-degree tests for bounded individual degree appear in numerous constructions of probabilistic proof systems [BFL90, BFLS91, PS94, FHS94, ALM+98, RS97, FGL+96, BS08] and play a central role in constructing short PCPs [PS94, BS08, Mie09]. The common idea of such tests is to rely on the following characterization. A function $f\colon \mathbb{F}_q^m \to \mathbb{F}_q$ is a m-variate polynomial function of individual degrees at most d if and only if, for any k-dimensional axis-parallel affine subspace S of $\mathbb{F}_q^m$, the restriction of f to S is a k-variate polynomial of individual degree d.

Ben-Sasson and Sudan [BS08] constructed a PCPP for the tensor product of RS codes by relying on their PCPP for Reed-Solomon codes. The PCPP to test a function $f\colon L^m \to \mathbb{F}$ is composed by a PCPP for Reed-Solomon codes (RS-PCPP) for each restrictions of f to an axis-parallel line. Therefore, the prover needs to compute $m\,|L|^{m-1}$ RS-PCPP, which yields prover complexity and proof length less than $m|L|^m \log^{1.5}|L|$. Both verifier complexity and query complexity are polylogarithmic in $|L|$. Our IOPP for the tensor of RS codes outperforms on all these parameters.

In the IOP model, there is no IOPP specifically tailored for tensor product of Reed-Solomon codes. Ron-Zewi and Rothblum [RR20] proposed an IOPP for any language computable in $\mathrm{poly}(n^m)$ time and bounded space. In particular, this gives a linear-size IOPP for Reed-Muller codes and tensor product of Reed-Solomon codes with polynomial prover complexity and sublinear verifier complexity.

However, there are a couple of IOPP constructions for m-wise tensor product of a generic linear code C. Indeed, axis-parallel tests enable local testability of repeated tensor products of any linear code [BS06, Vid15, CMS17]. Ben-Sasson *et al.* [BCG+17] suggested a 1-round IOPP system for tensor product codes $C^{\otimes m}$, where C is an arbitrary linear code and $m \geqslant 3$. Through interactive proof composition, Ben-Sasson *et al.* combine the robust local tester of [BS06, Vid15, CMS17] for tensor product codes with the Mie's PCP of Proximity for non-deterministic languages [Mie09]. The IOPP system constructed there has sublinear proof length and constant query complexity, which is significantly better than our protocol. However, for fixed $m > 3$, the verifier in [BCG+17] runs in time which is polylogarithmic in the length n of the base code C, whereas our verifier decision complexity is strictly logarithmic in n. Besides, and as opposed as our work, the IOPP system of [BCG+17] assume the proximity parameter to be smaller than half the minimum distance of the tensor code. Our construction is arguably much simpler to implement, as we do not rely on an heavy PCPP for NTIME, like Mie's one [Mie09].

Recently, Bootle, Chiesa and Groth [BCG20] showed how to construct a m-round IOPP for tensor codes $C^{\otimes m}$, where C is an arbitrary linear code of length n and dimension k. Their construction also relies on a folding operation (inspired by the FRI protocol of [BBHR19]) but takes a different approach than ours due to their need to work with linear-time encodable codes. In particular, performing the folding operation defined in [BCG20] requires to run an encoding algorithm for the m-wise tensor code $C^{\otimes m}$. When considering C a Reed-Solomon code, best known encoding algorithms run in time at least quasi-linear in n. In contrast, our IOPP does not rely on any encoding

Scheme	Prover	Verifier	Query	Length	Rounds
[BS08, BCGT13]	$O(mn^m \log^{1.5} n)$	$\mathrm{polylog}(n)$	$\mathrm{polylog}(n)$	$O(mn^m \log^{1.5} n)$	0
[BCG+17]*	$o(n^m)$	$\mathrm{poly}(m + \log n)$	$O(1)$	$o(n^m)$	1
[RR20]	$\mathrm{poly}(n^m)$	$(n^m)^\varepsilon$	$O(1)$	$< n^m$	$O(1)$
[BCG20]	$O(mn^m \log n)$	$O(nm \log n)$	$O(nm)$	$O(n^m)$	m
Ours	$< 8n^m$	$< 8m \log n$	$< 2 \log n$	$< n^m$	$< m \log n$

*: restricted to $m \geqslant 3$ and δ smaller than half the minimum distance of the tensor code.

Figure 1: Partial comparison of IOPs of Proximity solving the problem of proximity testing for tensor product of RS codes of length n^m. Soundness is omitted since it is difficult to provide and compare uniformly. The construction in the first line is a PCP of Proximity.

procedure of neither the tensor code, nor the base code.

Proximity problem for Reed-Muller codes A substantial body of research studies low total degree test [GLR+91, RS92, RS96, RS97, AS03, BSVW03, MR08] with evaluations over the entire domain $\mathbb{F}_q^m$. For this setting, considering restrictions of f to affine subspaces of fixed dimension is quite natural. Indeed, if $f\colon \mathbb{F}_q^m \to \mathbb{F}_q$ has total degree at most d then all its restrictions to u-dimensional affine subspaces are u-variate polynomials of degree at most d.

For example, the "line-versus-point" test of Rubinfeld and Sudan [RS96] consists in checking the restriction of the function f to a randomly chosen line in $\mathbb{F}_q^m$. Analyses [RS96, AS03, ALM+98] showed that if the test accepts a function f with probability δ, then f agrees with a degree-d polynomial on $\simeq \delta$ fraction of points. The verifier queries $O(d^3)$ field elements to achieve constant soundness error. The original low-degree test of [RS96] can be reformulated in terms of a PCPP if we consider that an auxiliary oracle is given in addition to f. Such oracle proof is supposed to contain the restrictions of f to every line, represented as the $d + 1$ coefficients of a univariate polynomial. Then, the number of queries of the PCPP is only two, but symbols of the oracle proof belong are in a large alphabet $\mathbb{F}_q^d$. Similarly, restrictions to affine subspaces of higher dimensions have also been considered, such as the plane-versus-plane test [RS97, MR08] and cube-versus-cube test [BDN17]. The number of field elements needed to be queried is at least linear in d.

Most results apply to polynomials over fields that are larger than the degree bound d. The local testability of Reed-Muller codes when the degree is larger than the field size has been studied in [AKK+03, AKK+05, JPRZ04, KR04]. Aformentioned results show that generalized Reed-Muller codes are locally testable, and query complexity increases as the size of the field decreases.

Note however all the above constructions do not apply to the setting we consider where the function f has domain L^m where L is strictly contained in $\mathbb{F}_q$. Indeed, in such case, the notion of affine subspace does not exist.

By working in the IOPP model, we are able to construct a low-degree test for total degree with strictly linear oracle proof length which can be generated in linear time and admit logarithmic query complexity and verification time. As mentioned above, previous works require the verifier to make a number of queries which is at least linear in d. Moreover, the size of the oracle proof [RS92] is polynomial in q^m. In order to further reduce the proof size, constructions using a smaller subset of lines have been investigated [GS02, BSVW03, MR08]. However, such constructions do not achieve a strictly linear oracle proof length, but only proofs of almost linear size. Needlessly to say that proof length is a lower bound on prover running time.

2 Definitions and notations

2.1 Notations

Throughout this paper, we denote by $\mathbb{F}_q$ the finite field of size q and and by $\mathbb{F}_q^\times$ the multiplicative group of $\mathbb{F}_q$. The multiplicative subgroup generated by an element $\omega \in \mathbb{F}_q^\times$ will be denoted $\langle \omega \rangle$. The set of functions with domain D and values in $\mathbb{F}_q$ is denoted by $\mathbb{F}_q^D$.

We use the notation $[a \mathinner{..} b]$ for the set of integers $\{a, a+1, \dots, b\}$. Let $m \geqslant 1$ be an integer. Vectors are written in bold, and for two tuples $\boldsymbol{x} = (x_1, \dots, x_m)$ and $\boldsymbol{u} = (u_1, \dots, u_m)$, $\boldsymbol{x}^{\boldsymbol{u}}$ refers to $\boldsymbol{x}^{\boldsymbol{u}} := x_1^{u_1} \cdots x_k^{u_k}$. We use the notation $\boldsymbol{X} = (X_1, \dots, X_m)$, and $\mathbb{F}_q[\boldsymbol{X}]$ refers to the ring of polynomials in the indeterminates $X_1, \dots, X_m$. For a multivariate polynomial $P \in \mathbb{F}_q[\boldsymbol{X}]$, we denote by $\deg P$ the total degree of P and $\deg_{X_j} P$ the individual degree of P with respect to the indeterminate X_j.

The Hamming weight $w_H(\boldsymbol{u})$ of a vector $\boldsymbol{u} \in \mathbb{F}_q^n$ is the number of non-zero symbols of $\boldsymbol{u}$. We denote by $\Delta \colon \mathbb{F}_q^n \times \mathbb{F}_q^n \to [0,1]$ the relative Hamming distance over $\mathbb{F}_q$; namely for $\boldsymbol{u}, \boldsymbol{u'} \in \mathbb{F}_q^n$, $\Delta(\boldsymbol{u}, \boldsymbol{u'})$ equals the ratio of coordinates in which they differ. A code is any subset of $\mathbb{F}_q^n$, and a linear code is a $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$. Given $\boldsymbol{u} \in \mathbb{F}_q^n$ and a code $C \subseteq \mathbb{F}_q^n$, we define $\Delta(\boldsymbol{u}, C)$ to be the minimal distance between $\boldsymbol{u}$ and any codeword of C. If $\Delta(\boldsymbol{u}, C) > \delta$, we say that $\boldsymbol{u}$ is δ-far from C, otherwise $\boldsymbol{u}$ is δ-close to C. We will consider *evaluation codes*. In this setting, we view codewords as functions in $\mathbb{F}_q^D$, and for $f \in C$ and $x \in D$, $f(x)$ naturally denotes the x-entry of the codeword f. Henceforth, the term code will always refer to a linear code.

2.2 Tensor product of Reed-Solomon codes

Given two linear codes $C_1 \subseteq \mathbb{F}_q^{n_1}$ and $C_2 \subseteq \mathbb{F}_q^{n_2}$, a matrix $M \in \mathbb{F}^{n_2 \times n_1}$ belongs to the tensor product code $C_2 \otimes C_1$ if and only if each row of M belongs to C_1 and each column of M belongs to C_2. For $m \geqslant 1$ and a code $C \subseteq \mathbb{F}_q^n$, we write $C^{\otimes m}$ for the m-wise tensor product of C, where $C^{\otimes m}$ is inductively defined by $C^1 = C$ and $C^{\otimes m} = C^{\otimes m-1} \otimes C$ for $m > 1$.

Definition 2 (Reed-Solomon code)**.** *Given $L \subseteq \mathbb{F}_q$ and $k \leqslant |L|$, we denote by $\mathsf{RS}[\mathbb{F}_q, L, k]$ the Reed-Solomon (RS) code over alphabet $\mathbb{F}_q$ defined by*

$$\mathsf{RS}[\mathbb{F}_q, L, k] := \left\{ f \in \mathbb{F}_q^L \mid \exists P \in \mathbb{F}_q[X], \deg P < k \ \textit{s.t.}\ \forall x \in L,\ f(x) = P(x) \right\}.$$

The code $\mathsf{RS}[\mathbb{F}_q, L, k]$ is a linear code of blocklength $|L|$, dimension k, rate $\rho = \frac{k}{|L|}$ and relative minimum distance $\lambda = 1 - \frac{k-1}{|L|}$.

The tensor product of Reed-Solomon codes admits the following aternative definition.

Definition 3 (Tensor product of Reed-Solomon code)**.** *Given $L \subset \mathbb{F}_q$, and $m, k \geqslant 1$, such that $k \leqslant |L|$, we denote by $(\mathsf{RS}[\mathbb{F}_q, L, k])^{\otimes m}$ the m-wise tensor product of the code $\mathsf{RS}[\mathbb{F}_q, L, k]$. Equivalently, the $(\mathsf{RS}[\mathbb{F}_q, L, k])^{\otimes m}$ can be defined as follows*

$$(\mathsf{RS}[\mathbb{F}_q, L, k])^{\otimes m} := \left\{ f \in \mathbb{F}_q^{L^m} \mid \exists P \in \mathbb{F}_q[\boldsymbol{X}], \deg_{X_i} P < k, i \in [1 \mathinner{..} m],\ \textit{such that} \ \forall x \in L,\ f(\boldsymbol{x}) = P(\boldsymbol{x}) \right\}. \quad (1)$$

The tensor product code $(\mathsf{RS}[\mathbb{F}_q, L, k])^{\otimes m}$ has length $|L|^m$, dimension k^m, rate $\left(\frac{k}{|L|}\right)^m$ and relative distance $\left(1 - \frac{k-1}{|L|}\right)^m$.

2.3 Short Reed-Muller codes

Reed-Muller codes consist of evaluation of multivariate polynomials with coefficients in $\mathbb{F}_q$ of bounded total degree. The classical definition of (generalized) Reed-Muller codes involves evaluations over the whole finite field. We introduce here codes whose support is $L^m \subset \mathbb{F}_q^m$, where L may be much smaller than $\mathbb{F}_q$. This is an easy generalization, and we call these codes *short Reed-Muller codes.*

Definition 4 (Short Reed-Muller code)**.** *A* short Reed-Muller code *with support $L^m \subset \mathbb{F}_q^m$ is defined as follows*

$$\mathsf{SRM}\left[\mathbb{F}_q, L, m, k\right] := \left\{ f \in \mathbb{F}_q^{L^m} \mid \exists P \in \mathbb{F}_q[\boldsymbol{X}],\, \deg P < k \ \textit{s.t.}\ \forall \boldsymbol{x} \in L^m,\, f(\boldsymbol{x}) = P(\boldsymbol{x}) \right\}.$$

If $k \leqslant |L|$, the evaluation map from the space of multivariate polynomials of total degree less than k to the space of functions $\mathbb{F}_q^{L^m}$ is injective, thus the dimension of $\mathsf{SRM}\left[\mathbb{F}_q, L, m, k\right]$ is $\binom{m+k-1}{m}$. A bound on the minimum distance of $\mathsf{SRM}\left[\mathbb{F}_q, L, m, k\right]$ follows from the Schwartz-Zippel lemma [Zip79, Sch80], which states that any non-zero multivariate polynomial $P \in \mathbb{F}_q[\boldsymbol{X}]$ of total degree less than q cannot vanish in more than $\frac{\deg P}{|L|}$ fraction of L^m. The code $\mathsf{SRM}\left[\mathbb{F}_q, L, m, k\right]$ has length $|L^m|$, rate $\binom{m+k-1}{m} |L|^{-m}$ and relative distance at least $1 - \frac{k-1}{|L|}$.

Remark 1. *The setting where the support $L^m \subset \mathbb{F}_q^m$ with $|L| \ll |\mathbb{F}_q|$ is not commonly encountered in coding theory. We introduce the non-standard term* short Reed-Muller codes *to emphasize this fact. Notice that, strictly speaking, short Reed-Muller codes correspond to* punctured *codes, and not* shortened *codes.*

3 Generic interactive oracle proof of proximity based on folding operators

We formulate an abstract framework to construct proximity tests for codes in the IOP model from distance-preserving folding operators from the FRI protocol [BBHR18].

Let $\mathbb{F}$ be some finite field. Let us consider an $\mathbb{F}$-linear code $C \subset \Sigma^D$, where Σ is an $\mathbb{F}$-linear space not necessarily equal to $\mathbb{F}$, and D is some evaluation domain.

3.1 Folding operators

In this section, we assume that one has defined a finite sequence of codes $(C_i)_{0 \leqslant i \leqslant r}$ for some integer r, where $C_0 := C$ and each code $C_i \subset \Sigma^{D_i}$. We will assume that the evaluations domains $(D_i)_{0 \leqslant i \leqslant r}$ satisfy the following. For each $i \in [0\,..\,r-1]$, assume there exist an integer l_i and a map $\pi_i \colon D_i \to D_{i+1}$ such that π_i is l_i-to-1 from D_i to $\pi_i(D_i) = D_{i+1}$. In particular, $|D_{i+1}| = \frac{|D_i|}{l_i}$. For any $y \in D_{i+1}$, we will denote $S_y := \pi_i^{-1}(\{y\})$ the set of the l_i preimages of y by the function π_i.

Moreover, suppose that for each $i \in [0\,..\,r-1]$, one can define a family of *folding operators* $\mathsf{Fold}\left[\cdot, \boldsymbol{p}\right] : \Sigma^{D_i} \to \Sigma^{D_{i+1}}$ parameterized by $\boldsymbol{p} \in \mathbb{F}^t$ for some positive integer t. These operators are designed to "compress" functions evaluated over D_i into functions over D_{i+1} and feature nice properties with respect to the codes C_i and C_{i+1}.

Definition 5 (Folding operator)**.** *A* folding operator *for the code C_i is a map $\boldsymbol{\mathsf{Fold}}\left[\cdot, \cdot\right] : \Sigma^{D_i} \times \mathbb{F}^t \to \Sigma^{D_{i+1}}$ satisfying the following properties.*

1. *(Completeness) For any $\boldsymbol{p} \in \mathbb{F}^t$, $\boldsymbol{\mathsf{Fold}}\left[C_i, \boldsymbol{p}\right] \subseteq C_{i+1}$.*

2. *(Locality) For any function $f\colon D_i \to \Sigma$, $\boldsymbol{p} \in \mathbb{F}^t$ and $y \in D_{i+1}$, one can compute $\textbf{\textsf{Fold}}[f, \boldsymbol{p}](y)$ by making l_i queries to the function f.*

To ensure soundness of the IOPP based on folding, we will also require that a folding operator preserves the relative distance. Namely, if a function $f\colon D_i \to \Sigma$ is far from the code C_i, we expect the folding of the function f to be far from the code C_{i+1} with high probability over $\boldsymbol{p} \in \mathbb{F}^t$. We formulate this distance preservation property in terms of relative weighted agreements.

Definition 6 (Weighted agreement)**.** *For any weight function $\phi\colon D \to [0,1]$, we define the ϕ-*agreement *of $u, v \in \Sigma^D$, denoted $\mu_\phi(u,v)$, as follows:*

$$\mu_\phi(u,v) := \frac{1}{|D|} \sum_{\substack{x \in D \\ u(x)=v(x)}} \phi(x).$$

*Moreover, given $C \subset \Sigma^D$ and $u \in \Sigma^D$, we define the ϕ-*agreement of u with C*, denoted $\mu_\phi(u,C)$, as*

$$\mu_\phi(u,C) := \max_{v \in C} \mu_\phi(u,v).$$

Observe that, if a weight function $\phi\colon D \to [0,1]$ is constant equal to 1, then μ_ϕ is the standard notion of relative agreement, *i.e.* for any $u, v \in \Sigma^D$ and any subset $S \in \Sigma^D$, we have

$$\mu_\phi(u,v) = \frac{1}{|D|} |\{x \in D \mid u(x) = v(x)\}| = 1 - \Delta(u,v),$$

and

$$\mu_\phi(u,S) = 1 - \Delta(u,S).$$

Consequently, we have:

Fact 1. *For any weight function $\phi\colon D \to [0,1]$, any $u, v \in \Sigma^D$ and any $S \subset \Sigma^D$, we have*

$$\mu_\phi(u,v) \leqslant 1 - \Delta(u,v) \qquad \text{and} \qquad \mu_\phi(u,S) \leqslant 1 - \Delta(u,S).$$

Definition 7 (Distance preservation)**.** *Let us consider a function $\gamma\colon (0,1) \times [0,1] \to [0,1]$ which is strictly increasing with respect to the second variable. Let $i \in [0 \,..\, r-1]$, and denote by λ_{i+1} the minimum relative distance of C_{i+1}. We say that a folding operator $\textbf{\textsf{Fold}}[\cdot,\cdot]$ satisfies* distance preservation *if, for any weight functions $\phi_i\colon D_i \to [0,1]$ and $\phi_{i+1}\colon D_{i+1} \to [0,1]$ satisfying*

$$\forall y \in D_{i+1}, \qquad \phi_{i+1}(y) \geqslant \frac{1}{l_i} \sum_{x \in \pi_i^{-1}(\{y\})} \phi_i(x), \tag{2}$$

any $\varepsilon \in (0,1)$, any $\delta \in (0, \gamma(\varepsilon, \lambda_{i+1}))$ and any function $f\colon D_i \to \Sigma$ of ϕ_i-agreement

$$\mu_{\phi_i}(f, C_i) < 1 - \delta,$$

we have

$$\Pr_{\boldsymbol{p} \in \mathbb{F}^t}\left[\mu_{\phi_{i+1}}(\textbf{\textsf{Fold}}[f,\boldsymbol{p}], C_{i+1}) > 1 - \delta + \varepsilon\right] < \eta,$$

for some $\eta \in (0,1)$.

The reason why we consider weighted agreements instead of the standard relative Hamming distance is that it will facilitate tracking inconsistencies between the oracles actually sent by a malicious prover and the expected prover's messages (prescribed by the protocol) during soundness analysis. The weight functions ϕ_i, ϕ_{i+1} are left undefined in 7 since weights will be assigned to elements of the supports D_i, D_{i+1} depending on a prover's strategy.

3.2 Generic IOPP construction

Now we describe a generic way of constructing a public-coin IOPP to test proximity to a code $C \subseteq \Sigma^D$ using folding operators.

Taking $C_0 = C$ and $D_0 = D$, we consider a sequence of codes $(C_i)_{0 \leqslant i \leqslant r}$ with a family of folding operators defined as per Section 3.1. As in the FRI protocol [BBHR18], our protocol is divided into two phases. The interactive phase is referred to as COMMIT phase, while the non-interactive one is named QUERY phase.

The COMMIT phase is an interaction over r rounds between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. At each round i, the verifier samples a random element $\boldsymbol{p}_i \in \mathbb{F}^t$. The prover answers with an oracle function $f_{i+1}\colon D_i \to \Sigma$, expected to be equal to $\textsf{Fold}\,[f_i, \boldsymbol{p}_i]$.

During the QUERY phase, the task of the verifier $\mathcal{V}$ is to check that each pair of oracle functions (f_i, f_{i+1}) is consistent. The standard idea is to test whether the equality

$$f_{i+1}(y_{i+1}) = \textsf{Fold}\,[f_i, \boldsymbol{p}_i]\,(y_{i+1}) \tag{3}$$

holds at a random point $y_{i+1} \in D_{i+1}$. Thanks to the local property of the folding operator, the verifier $\mathcal{V}$ can perform such a test by querying l_i entries of f_i and one entry of f_{i+1}. As in [BBHR18], we call this step of verification a *round consistency test*. More specifically, the verifier begins by sampling uniformly at random $y_0 \in D_0$ and once this is done, all the locations of the round consistency tests below the current **query test** are determined. Indeed, for each i, $\mathcal{V}$ defines $y_{i+1} := \pi_i(y_i)$ to be the point where Equation (3) is checked. Through this process, and as in the FRI protocol, the round consistency tests are correlated in order to improve soundness. Such a **query test** can be seen as a *global* consistency test. As a final test, the verifier checks that $f_r \in C_r$ and rejects if it is not the case.

Remark 2. *Depending on the evaluation codes considered, it may be convenient to adapt the final round as follows in order to avoid the cost of a membership test to C_r. During the last round of the COMMIT phase, instead of sending a codeword $f_r \in C_r$, an honest $\mathcal{P}$ may "unencode" f_r, meaning he retrieves a word w_r from the messages space of C_r whose encoding leads to $f_r \in C_r$. The prover $\mathcal{P}$ sends k_r message symbols to represent w_r, where k_r refers to the message length of the code C_r. In that case, the verifier no longer needs to run a membership test to the code C_r during the QUERY phase. The verifier $\mathcal{V}$ can re-encode w_r, interpreting f_r to be the the encoding of w_r. This variant of the protocol is the one presented in the FRI protocol [BBHR18] for Reed-Solomon codes (in that case, w_r is the coefficients of a polynomial of bounded degree).*

Notice that in some cases, the verifier does not need to encode w_r, e.g. *when the function f_r is expected to be the evaluation of a constant polynomial function.*

Theorem 2. *Let $(C_i)_{0 \leqslant i \leqslant r}$ be a sequence of codes such that there exists a family of folding operators for each code C_i satisfying Definitions 5 and 7. The r-rounds IOPP system $(\mathcal{P}, \mathcal{V})$ for the code $C = C_0$ of Figure 2 is public-coin and fulfills the following properties:*

Perfect completeness: *If $f \in C$ and if the oracles $f_1, \dots f_r$ are computed by an honest prover $\mathcal{P}$, then $\mathcal{V}$ outputs accept with probability 1.*

Soundness: *Assume $f\colon D \to \Sigma$ is δ-far from C. For any $\varepsilon \in (0,1)$ and any unbounded prover $\mathcal{P}^*$, the verifier $\mathcal{V}$ outputs accept after α repetitions of the QUERY phase with probability at most*

$$r\eta + (1 - \min(\delta, \gamma(\varepsilon, \lambda)) + r\varepsilon)^{\alpha},$$

where λ denotes the smallest relative minimum distance of the codes C_i, $i \in [0\,..\,r]$ and $\gamma(\cdot,\cdot)$ is the function defined in 7.

Input common to Prover and Verifier:

- a sequence of codes $(C_i)_{0 \leqslant i \leqslant r}$ such that $C_i \subset \Sigma_i^D$.

COMMIT Phase
(interactive)

Prover's input:

- $f = f_0 \colon D_0 \to \mathbb{F}_q$.

Protocol:

1. For each round i from 0 to $r - 1$:
   (a) **Verifier** $\mathcal{V}$ picks uniformly at random an element $\boldsymbol{p}_i \in \mathbb{F}_q^t$;
   (b) **Verifier** $\mathcal{V}$ sends $\boldsymbol{p}_i$ to **Prover** $\mathcal{P}$;
   (c) An honest **Prover** $\mathcal{P}$ computes $\mathsf{Fold}\left[f_i, \boldsymbol{p}_i\right] : D_{i+1} \to \Sigma$

Prover's output:

- a sequence of oracle functions $f_0 \in \Sigma^{D_1}, \ldots, f_r \in \Sigma^{D_r}$.

QUERY Phase
(run by $\mathcal{V}$ only)

Verifier's input:

- $\boldsymbol{p}_0, \ldots \boldsymbol{p}_{r-1}$ the challenges sent during steps 1b of the COMMIT phase,
- oracle access to the **Prover**'s output functions $f_0 \in \Sigma^{D_1}, \ldots, f_r \in \Sigma^{D_r}$,
- a repetition parameter $\alpha \in \mathbb{N}$.

Output: acccept or reject.

Protocol:

1. Repeat α times the following **query test**:
   (a) Sample $y_0 \in D_0$ uniformly at random;
   (b) For $i = 0$ to $r - 1$:
      i. Define $y_{i+1} \in D_{i+1}$ as $y_{i+1} = \pi_i(\boldsymbol{y}_i)$;
      ii. Query f_i on $S_{y_{i+1}} = \pi_i^{-1}\left(\{y_{i+1}\}\right)$ to compute $\mathsf{Fold}\left[f_i, \boldsymbol{p}_i\right](y_{i+1})$;
      iii. Query $f_{i+1}(y_{i+1})$;
      iv. If $f_{i+1}(y_{i+1}) \neq \mathsf{Fold}\left[f_i, \boldsymbol{p}_i\right](y_{i+1})$, outputs **reject** *(Round consistency check)* ;
2. Outputs **accept** if and only if $f_r \in C_r$ (*Final test*).

Figure 2: IOPP $(\mathcal{P}, \mathcal{V})$ for a code $C = C_0$ based on folding operators

Proof. *(Perfect completeness)* Assume that $f_0 \in C_0$. An honest prover who follows the prescription of the COMMIT phase will make the round consistency tests pass with probability 1 for all rounds i. By completeness of the folding operator for every round i, we have $f_r \in C_r$. Therefore, the final test also passes. Thus, the verifier always accepts.

(Soundness) Our analysis relies on techniques of proofs from [BGKS20]. A similar analysis appears in [BN20]. We perform our analysis for $\alpha = 1$ repetition of the **query test**. We observe that the soundness error for $\alpha > 1$ directly follows from this case. Let $(f_i)_{1 \leqslant i \leqslant r}$ be the output of the COMMIT phase and $(y_i)_{1 \leqslant i \leqslant r}$ be the query points selected for the QUERY phase. The verifier accepts if both

1. for all $i \in [0 \,..\, r-1]$, $f_{i+1}(y_{i+1}) = \mathbf{Fold}\left[f_i, \boldsymbol{p}_i\right](y_{i+1})$,
2. $f_r \in C_r$.

Observe that if $f_r \notin C_r$, the verifier rejects with probability 1, therefore we continue the analysis assuming $f_r \in C_r$. Since the soundness analysis is quite technical, we divide it into several steps to improve readability.

Step 1 – Coloring the graph induced by prover's oracles. Set G the $(r+1)$-layered graph with vertex set $D_0 \sqcup D_1 \sqcup \cdots \sqcup D_r$. The edges of G consist in the couples $(y_i, y_{i+1}) \in D_i \times D_{i+1}$ such that $\pi_i(y_i) = y_{i+1}$. For any edge of G, the vertex y_{i+1} is called the *parent* of y_i. Vertices sharing the same parent are said to be *siblings*. For any vertex within the last layer $y_r \in D_r$, we denote by $G|_{y_r}$ the subgraph of G corresponding to the complete tree with root y_r. Therefore the trees $G|_{y_r}$ are disjoint.

A query test starts by selecting a leaf $y_0 \in D_0$, which belongs to a unique tree $G|_{y_r}$ for a certain $y_r \in D_r$. The verifier queries one set of siblings at each layer $i \in [0 \,..\, r-1]$ of $G|_{y_r}$, whose union forms a subset of vertices of G that we call the *path from y_0 to y_r*. Note that a path to y_r does not include y_r.

We now color the vertices of G (except those in the last layer) according to their success in passing the round consistency test. For $i \in [0 \,..\, r-1]$, a vertex $y_i \in D_i$ is colored green if

$$f_{i+1}(\pi_i(y_i)) = \mathbf{Fold}\left[f_i, \boldsymbol{p}_i\right](\pi_i(y_i))$$

and colored red otherwise. Notice siblings have the same color. The verifier outputs accept if and only if every vertex along the queried path from y_0 to y_r is green.

Step 2 – Defininining the function of weights. Define $\psi_0 D_0 \to [0,1]$ such that $\psi_0(x) = 1$ if and only if $x \in D_0$ is green. For all $i \in [1, r-1]$, define function

$$\psi_i \colon D_i \to [0,1]$$

such that $\psi_i(x)$ is equal to the fraction of leaves $x_0 \in D_0$ for which the path from x_0 to $x \in D_i$ contains only green vertices.

Step 3 – Rejection probability in terms of weighted agreement. By construction, the probability $\mathsf{err}_{\text{query}}$ that the verifier accepts during the QUERY phase is given by

$$\mathsf{err}_{\text{query}} = \frac{1}{|D_r|} \sum_{x \in D_r} \psi_r(x).$$

For $i \in [0 \mathinner{..} r-1]$, let us set

$$\mu_{f_i} := \mu_{\psi_i}(f_i, C_i), \tag{4}$$

where the ψ-agreement μ_ψ is defined in Definition 6. Since $f_r \in C_r$, observe that

$$\mathsf{err}_{\text{query}} = \mu_{f_r}. \tag{5}$$

Step 4 – Relating agreement of f_i with the one of the folding of f_{i-1}. For $i \in [0 \mathinner{..} r-1]$, we define $E_{i+1} \subseteq D_{i+1}$ to be the set of coordinates where f_{i+1} differs from $\mathsf{Fold}[f_i, \boldsymbol{p}_i]$, i.e.

$$E_{i+1} := \{y \in D_{i+1} \mid \forall x \in S_y, x \text{ is } \mathsf{red}\}.$$

Let us fix $i \in [0 \mathinner{..} r-1]$. We aim to show that

$$\mu_{\psi_{i+1}}(\mathsf{Fold}[f_i, \boldsymbol{p}_i], C_{i+1}) \geqslant \mu_{\psi_{i+1}}(f_{i+1}, C_{i+1}).$$

Let $v \in C_{i+1}$ such that

$$\mu_{\psi_{i+1}}(f_{i+1}, v) = \mu_{\psi_{i+1}}(f_{i+1}, C_{i+1})$$

(breaking ties arbitrarily). Since for any $y \in E_{i+1}$, $\psi_{i+1}(y) = 0$, we can write

$$\mu_{\psi_{i+1}}(\mathsf{Fold}[f_i, \boldsymbol{p}_i], v) = \frac{1}{|D_{i+1}|} \sum_{\substack{y \in D_{i+1} \setminus E_{i+1} \\ \mathsf{Fold}[f_i, \boldsymbol{p}_i](y) = v(y)}} \psi_{i+1}(y)$$

and

$$\mu_{\psi_{i+1}}(f_{i+1}, v) = \frac{1}{|D_{i+1}|} \sum_{\substack{y \in D_{i+1} \setminus E_{i+1} \\ f_{i+1}(y) = v(y)}} \psi_{i+1}(y).$$

But $\mathsf{Fold}[f_i, \boldsymbol{p}_i]$ and f_{i+1} coincide on the set $D_{i+1} \setminus E_{i+1}$, hence

$$\mu_{\psi_{i+1}}(\mathsf{Fold}[f_i, \boldsymbol{p}_i], v) = \mu_{\psi_{i+1}}(f_{i+1}, v).$$

Moreover, we have

$$\mu_{\psi_{i+1}}(\mathsf{Fold}[f_i, \boldsymbol{p}_i], C_{i+1}) \geqslant \mu_{\psi_{i+1}}(\mathsf{Fold}[f_i, \boldsymbol{p}_i], v)$$

by definition of the ψ_{i+1}-agreement. Thus,

$$\mu_{\psi_{i+1}}(\mathsf{Fold}[f_i, \boldsymbol{p}_i], C_{i+1}) \geqslant \mu_{\psi_{i+1}}(f_{i+1}, C_{i+1}). \tag{6}$$

Step 5 – Controlling the weighted agreement after folding. Let $\varepsilon \in (0,1)$ and

$$\delta_i < \min(1 - \mu_{f_i}, \gamma(\varepsilon, \lambda_i)).$$

Observe that

$$\psi_{i+1}(y) = \begin{cases} 0 & \text{if } y \in E_{i+1}, \\ \dfrac{1}{l_i} \displaystyle\sum_{x \in S_y} \psi_i(x) & \text{if } y \in D_{i+1} \setminus E_{i+1}. \end{cases}$$

Thus, the functions ψ_i satisfy (2):

$$\forall y \in D_{i+1}, \psi_{i+1}(y) \geqslant \frac{1}{l_i} \sum_{x \in S_y} \psi_i(x).$$

Since the folding operators satisfy distance preservation (7), we have for all $i \in [0 \mathinner{..} r-1]$

$$\Pr_{\boldsymbol{p}_i \in \mathbb{F}^t}\left[\mu_{\psi_{i+1}}(\textsf{Fold}\left[f_i, \boldsymbol{p}_i\right], C_{i+1}) > 1 - \delta_i + \varepsilon\right] \leqslant \eta,$$

which yields

$$\Pr_{\boldsymbol{p}_i \in \mathbb{F}^t}\left[\mu_{\psi_{i+1}}(\textsf{Fold}\left[f_i, \boldsymbol{p}_i\right], C_{i+1}) > \max\left(\mu_{f_i}, 1 - \gamma(\varepsilon, \lambda_i)\right) + \varepsilon\right] \leqslant \eta,$$

where μ_{f_i} is the notation introduced in (4).

Step 6 – Controlling the weighted agreement of f_r by the one of f_0. Let $\lambda = \min_i(\lambda_i)$. As the function $\gamma(\varepsilon, \cdot)$ is strictly increasing, we have

$$\Pr_{\boldsymbol{p}_i \in \mathbb{F}^t}\left[\mu_{\psi_{i+1}}(\textsf{Fold}\left[f_i, \boldsymbol{p}_i\right], C_{i+1}) > \max\left(\mu_{f_i}, 1 - \gamma(\varepsilon, \lambda)\right) + \varepsilon\right] \leqslant \eta.$$

Recalling (6), we deduce that

$$\Pr_{\boldsymbol{p}_i \in \mathbb{F}^t}\left[\mu_{f_{i+1}} > \max\left(\mu_{f_i}, 1 - \gamma(\varepsilon, \lambda)\right) + \varepsilon\right] \leqslant \eta.$$

By a union bound, the event that for all $i \in [0 \mathinner{..} r-1]$,

$$\mu_{f_{i+1}} \leqslant \max\left(\mu_{f_i}, 1 - \gamma(\varepsilon, \lambda)\right) + \varepsilon$$

occurs with probability at least $1 - r\eta$. If this event occurs, then

$$\mu_{f_r} \leqslant \max\left(\mu_{f_0}, 1 - \gamma(\varepsilon, \lambda)\right) + r\varepsilon.$$

Therefore

$$\Pr_{\boldsymbol{p}_0, \cdots, \boldsymbol{p}_{r-1} \in \mathbb{F}^t}\left[\mu_{f_r} \leqslant \max\left(\mu_{f_0}, 1 - \gamma(\varepsilon, \lambda)\right) + r\varepsilon\right] \geqslant 1 - r\eta.$$

Final step – Putting everything together. Recalling Theorem 1, we have

$$\mu_{f_0} \leqslant 1 - \Delta(f_0, C_0) < 1 - \delta.$$

Set $\textsf{err}_{\text{commit}} := r\eta$. We deduce that with probability at least $1 - \textsf{err}_{\text{commit}}$ over the randomness of the verifier during the $\textsf{COMMIT}$ phase, the verifier accepts with probability at most

$$\begin{aligned} \textsf{err}_{\text{query}} = \mu_{f_r} &\leqslant \max(\mu_{f_0}, 1 - \gamma(\varepsilon, \lambda)) + r\varepsilon \\ &< 1 - \min(\delta, \gamma(\varepsilon, \lambda)) + r\varepsilon. \end{aligned}$$

Remark 3. *The same proof holds for the variant of the protocol described in Remark 2, which results in no change in Theorem 2.*

4 Preliminaries about multivariate polynomials

4.1 Low-degree extensions

Proposition 1 (Low-degree extension ([BFLS91])). *Let $H_1, \ldots, H_m \subseteq \mathbb{F}_q$ and let $f\colon H_1 \times \cdots \times H_m \to \mathbb{F}_q$ be a function. Then there exists a unique polynomial $\widehat{f}$ in m variables over $\mathbb{F}_q$ such that:*

1. $\widehat{f}$ *has degree* $\deg_{X_i} \widehat{f} < |H_i|$ *in its* i*-th variable,*

2. $\widehat{f}$ *agrees with* f *on* $H_1 \times \cdots \times H_m$.

The polynomial $\widehat{f}$ *is referred to as* the low-degree extension *of the function* f *(with respect to* $\mathbb{F}_q$ *and* $H_1, \ldots, H_m$*).*

Proof. For $H \subset \mathbb{F}_q$ and $h \in H$, denote $L_{H,h}(X) := \prod_{k \in H \setminus \{h\}} \frac{X-k}{h-k}$ the Lagrange polynomial. The existence follows from the observation that the polynomial defined by

$$\sum_{\boldsymbol{h} \in H_1 \times \cdots \times H_m} f(\boldsymbol{h}) \prod_{j=1}^{m} L_{H_j, h_j}(X_j)$$

has degree less than $|H_j|$ in each variable and agrees with f on $H_1 \times \cdots \times H_m$. An easy induction on m leads to uniqueness. □

The arithmetic complexity of solving the interpolation problem of computing the coefficients of the low-degree extension of a function $f\colon H_1 \times \cdots \times H_m \to \mathbb{F}_q$ appears in [Pan94] for general subsets $H_1, \ldots, H_m \subset \mathbb{F}_q$. In our work, we will be specifically interested in the cost of interpolating and evaluating low-degree extensions of a function defined on a grid of size 2^m.

Definition 8. *A* multilinear polynomial *is a multivariate polynomial whose degree in each variable is at most one.*

Lemma 1 (Multilinear interpolation ([Pan94]))**.** *Let* $H_1, \ldots, H_m \subset \mathbb{F}_q$ *of size 2 and let* $f\colon H_1 \times \cdots \times H_m \to \mathbb{F}_q$ *be a function. The low-degree extension of* f *is a multilinear polynomial* $\widehat{f} \in \mathbb{F}_q[\boldsymbol{X}]$*. The number of operations required to interpolate* $\widehat{f}$ *is at most* $5m2^{m-1}$ *arithmetic operations.*

Lemma 2 (Efficient multilinear extension)**.** *Let* $H_1, \ldots, H_m \subset \mathbb{F}_q$ *of size 2 and let* $f\colon H_1 \times \cdots \times H_m \to \mathbb{F}_q$ *be a function. The low-degree extension of* f *is a multilinear polynomial* $\widehat{f} \in \mathbb{F}_q[\boldsymbol{X}]$ *and, given* $\boldsymbol{p} \in \mathbb{F}_q^m$*, evaluating* $\widehat{f}$ *at* $\boldsymbol{p}$ *can be done in less than* $4(2^m + m)$ *arithmetic operations.*

Proof. For any $\boldsymbol{h} = (h_1, \ldots, h_m) \in H_1 \times \cdots \times H_m$, define $L_{\boldsymbol{h}}(\boldsymbol{X}) := \prod_{j=1}^{m} L_{H_j, h_j}(X_j)$. For any $\boldsymbol{p} = (p_1, \ldots, p_m) \in \mathbb{F}_q^m$, we have

$$\widehat{f}(\boldsymbol{p}) = \sum_{\boldsymbol{h} \in H_1 \times \cdots \times H_m} f(\boldsymbol{h}) L_{\boldsymbol{h}}(\boldsymbol{p}). \tag{7}$$

As suggested by [VSBW13] regarding multilinear extensions over the boolean hypercube, we observe that $(L_{\boldsymbol{h}}(\boldsymbol{p}))_{\boldsymbol{h} \in H_1 \times \cdots \times H_m}$ can be computed in linear time and linear space using dynamic programming.

Notice that for all $k \in [1 \mathinner{..} m]$,

$$\prod_{j=1}^{k} L_{H_j, h_j}(p_j) = L_{H_k, h_k}(p_k) \prod_{j=1}^{k-1} L_{H_j, h_j}(p_j)$$

and $\deg L_{H_k, h_k} = 1$. Given a table of values containing $\prod_{j=1}^{k-1} L_{H_j, h_j}(p_j)$ for all $(h_1, \ldots, h_{k-1}) \in H_1 \times \cdots \times H_{k-1}$, one can get the values $\prod_{j=1}^{k} L_{H_j, h_j}(p_j)$ for all $(h_1, \ldots, h_k) \in H_1 \times \cdots \times H_k$ by computing the couple of values $\left(L_{H_k, h_k}(p_k)\right)_{h_k \in H_k}$ using 2 additions and 2 divisions, and multiplying both of them by all the 2^{k-1} precomputed values. In sum, this step requires $2^k + 4$ operations. Thus, computing $L_{\boldsymbol{h}}(\boldsymbol{p})$ for all $\boldsymbol{h} \in H_1 \times \cdots \times H_m$ takes $\sum_{j=1}^{m} \left(2^j + 4\right) < 2 \cdot 2^m + 4m$ arithmetic operations. Finally, given the table of values of f and $(L_{\boldsymbol{h}}(\boldsymbol{p}))_{\boldsymbol{h} \in H_1 \times \cdots \times H_m}$, computing the right-hand side of (7) takes 2^m multiplications and $(2^m - 1)$ additions. □

4.2 Multivariate polynomial decomposition

One efficient way to build folding operators on codes formed by evaluations of polynomials relies on some ingenious decompositions, as in [BS08, BBHR18]. This section gathers all the technical results about such decompositions and their properties.

Lemma 3. *Let R be an integral domain, and let $q \in R[X]$ be a monic polynomial of degree l. For every $f \in R[X]$ there exists a unique sequence of polynomials $(f_u(X))_{0 \leqslant u \leqslant \lfloor \frac{\deg f}{l} \rfloor}$ such that*

$$f(X) = \sum_{u=0}^{\lfloor \deg f/l \rfloor} f_u(X) q(X)^u.$$

Furthermore, $\deg f_u < l$*, for* $u \in [0 \,..\, \lfloor \deg f/l \rfloor]$.

Proof. As in [BS08, Proposition 6.3], we consider the Euclidean division of $f(X)$ by $(Y - q(X))$ in the polynomial ring $R[Y][X]$, i.e. with respect to the X variable. Polynomial division by a monic polynomial over an integral domain shares the same properties as polynomial division over a field. There exists a unique pair of polynomials $A,\ B \in R[X][Y]$ such that

$$f(X) = (Y - q(X))A(X,Y) + B(X,Y)$$

such that $\deg_X B < \deg q$. Writing $B(X,Y) = \sum f_u(X)Y^u$, with $\deg f_u < \deg q$, and evaluating the above identity at $Y = q(X)$ gives $f(X) = \sum f_u(X)q(X)^u$ as required, with appropriate degree bounds. The uniqueness of the decomposition follows from the one of the remainder B in the Euclidean division, as any other decomposition $\sum_{u=0} f'_u(X)q(X)^u$ with the same degree bounds would induce another remainder $\sum_{u=0} f'_u(X)Y^u \neq B$. $\square$

Lemma 4. *Let R be an integral domain, and let $q \in R[X]$ be a monic polynomial of degree l. For every $f \in R[\boldsymbol{X}]$ there exists a unique sequence $(f_{\boldsymbol{u}})_{\boldsymbol{u} \in \boldsymbol{U}}$ of polynomials in $R[\boldsymbol{X}]$ such that*

$$f(\boldsymbol{X}) = \sum_{\boldsymbol{u}=(u_1,\ldots,u_m)\in\boldsymbol{U}} f_{\boldsymbol{u}}(X_1,\ldots,X_m)q(X_1)^{u_1}\cdots q(X_m)^{u_m}, \tag{8}$$

where $\boldsymbol{U} = \left[0 \,..\, \lfloor \deg_{X_1} f/l \rfloor\right] \times \cdots \times \left[0 \,..\, \lfloor \deg_{X_m} f/l \rfloor\right]$ and $\deg_{X_i} f_{\boldsymbol{u}}(\boldsymbol{X}) < l$ for $i \in [1 \,..\, m]$ and $\boldsymbol{u} \in \boldsymbol{U}$.

Proof. The proof is done by induction on the number m of indeterminates, the case $m = 1$ being established in Lemma 3. Suppose the result holds for $m-1$ indeterminates and consider $f(\boldsymbol{X})$ as a polynomial in $R[X_1][X_2,\ldots,X_m]$. Since $R[X_1]$ is an integral domain, we can apply the induction hypothesis, and there exists a unique sequence $(f_{\boldsymbol{u}'}(X_1,X_2,\ldots,X_m))_{\boldsymbol{u}'\in\boldsymbol{U}'} \in R[X_1][X_2,\ldots,X_m]$ such that

$$f(X_1,X_2,\ldots,X_m) = \sum_{(u_2,\ldots,u_m)\in\boldsymbol{U}'} f_{u_2,\ldots,u_m}(X_1,X_2,\ldots,X_m)q(X_2)^{u_2}\cdots q(X_m)^{u_m}$$

where $\boldsymbol{U}' = \left[0 \,..\, \lfloor \deg_{X_2} f/l \rfloor\right] \times \cdots \times \left[0 \,..\, \lfloor \deg_{X_m} f/l \rfloor\right]$ and, for each $i \in [2 \,..\, m]$:

$$\deg_{X_i} f_{u_2,\ldots,u_m}(X_1,X_2,\ldots,X_m) < l.$$

Writing

$$f_{u_2,\ldots,u_m} = \sum_{0\leqslant u_2,\ldots,u_m<l} g_{u_2,\ldots,u_m}(X_1)X_2^{u_2}\cdots X_m^{u_m}$$

and applying Lemma 3 to each polynomial $g_{u_2,\ldots,u_m} \in R[X_1]$, we obtain a unique sequence

$$(g_{u_1,u_2,\ldots,u_m}(X_1))_{0 \leqslant u_1 \leqslant \lfloor (\deg_{X_1} f/l \rfloor}$$

of polynomials in $R[X_1]$ such that

$$g_{u_2,\ldots,u_m}(X_1) = \sum_{u_1=0}^{\lfloor (\deg_{X_1} f/l \rfloor} g_{u_1,u_2,\ldots,u_m}(X_1) q(X_1)^{u_1}$$

and $\deg g_{u_1,u_2,\ldots,u_m}(X_1) < l$. This gives

$$f_{u_2,\ldots,u_m} = \sum_{0 \leqslant u_2,\ldots,u_m < l} \sum_{u_1=0}^{\lfloor (\deg_{X_1} f/l \rfloor} g_{u_1,u_2,\ldots,u_m}(X_1) X_2^{u_2} \cdots X_m^{u_m} q(X_1)^{u_1},$$

which leads to the expected decomposition after collecting terms. $\square$

Proposition 2 (Multivariate decomposition)**.** *Let R be an integral domain, and let $q \in R[X]$ be a monic polynomial of degree l. For every $f \in R[\boldsymbol{X}]$ there exists a unique sequence $(g_{\boldsymbol{e}})_{\boldsymbol{e} \in [0\,..\,l-1]^m}$ of polynomials in $R[\boldsymbol{X}]$ such that*

$$f(\boldsymbol{X}) = \sum_{\boldsymbol{e} \in [0\,..\,l-1]^m} \boldsymbol{X}^{\boldsymbol{e}} g_{\boldsymbol{e}}\left(q(X_1), \ldots, q(X_m)\right), \tag{9}$$

and

- *for all $\boldsymbol{e} \in [0\,..\,l-1]^m$ and $j \in [1\,..\,m]$, $\deg_{X_j} g_{\boldsymbol{e}} \leqslant \left\lfloor \frac{\deg_{X_j} f}{l} \right\rfloor$,*
- *for all $\boldsymbol{e} \in [0\,..\,l-1]^m$, $\deg g_{\boldsymbol{e}} \leqslant \left\lfloor \frac{\deg f - w_H(\boldsymbol{e})}{l} \right\rfloor$.*

Proof. We use the notation of Lemma 4. For each $\boldsymbol{u} \in \boldsymbol{U}$, writing each polynomial $f_{\boldsymbol{u}}$ as $f_{\boldsymbol{u}}(\boldsymbol{X}) = \sum_{\boldsymbol{e} \in [0\,..\,l-1]^m} a_{\boldsymbol{u},\boldsymbol{e}} \boldsymbol{X}^{\boldsymbol{e}}$, Equation (8) becomes

$$\begin{aligned} f(\boldsymbol{X}) &= \sum_{\boldsymbol{u} \in \boldsymbol{U}} \sum_{\boldsymbol{e} \in [0\,..\,l-1]^m} a_{\boldsymbol{u},\boldsymbol{e}} \boldsymbol{X}^{\boldsymbol{e}} q(X_1)^{u_1} \cdots q(X_m)^{u_m}, \\ &= \sum_{\boldsymbol{e} \in [0\,..\,l-1]^m} \boldsymbol{X}^{\boldsymbol{e}} \sum_{\boldsymbol{u} \in \boldsymbol{U}} a_{\boldsymbol{u},\boldsymbol{e}} q(X_1)^{u_1} \cdots q(X_m)^{u_m}. \end{aligned}$$

For each $\boldsymbol{e} \in [0\,..\,l-1]^m$, define $g_{\boldsymbol{e}}(\boldsymbol{X}) = \sum_{\boldsymbol{u} \in \boldsymbol{U}} a_{\boldsymbol{u},\boldsymbol{e}} \boldsymbol{X}^{\boldsymbol{u}}$. We thus get the decomposition of Equation (9). The bounds for individual degrees of each $g_{\boldsymbol{e}}$ comes from the definition of $\boldsymbol{U}$. Moreover, we have $\deg f = \max_{\boldsymbol{e}} \{\deg(\boldsymbol{X}^{\boldsymbol{e}} g_{\boldsymbol{e}}(q(X_1), \ldots, q(X_m)))\}$, thus $\deg f \geqslant w_H(\boldsymbol{e}) + l \deg g_{\boldsymbol{e}}$.

The uniqueness of the sequence of polynomials $(g_{\boldsymbol{e}})_{\boldsymbol{e}}$ follows from the one of the sequence of polynomials $(f_{\boldsymbol{u}})_{\boldsymbol{u}}$. $\square$

5 Distance preservation for random multilinear combinations

In this section, we study a special case worst-case to average-case reduction of distance for linear subspaces. Several works looked at this question [RVW13, AHIV17, BKS18, BGKS20] for general linear subspaces, but we are interested in the following specific context. For $\boldsymbol{u} = (u_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m} \subset \mathbb{F}_q^D$, and $\boldsymbol{p} \in \mathbb{F}_q^m$, we consider the set

$$S_{\boldsymbol{u}} := \left\{ \sum_{\boldsymbol{e}\in\{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} u_{\boldsymbol{e}} \mid \boldsymbol{p} \in \mathbb{F}_q^m \right\}$$

of multilinear combinations of elements of $\boldsymbol{u}$. Given a linear code $C \subset \mathbb{F}_q^D$, we estimate the average-distance to C of an element $u' \in S_{\boldsymbol{u}}$ compared to the maximum distance to C of a member $u_{\boldsymbol{e}}$ from $\boldsymbol{u}$.

5.1 Hamming distance version

Proposition 3. *Let m be a positive integer. Let $C \subset \mathbb{F}_q^D$ be a linear code of relative distance $\lambda = \Delta(C)$. Let $\varepsilon, \delta > 0$ such that $\varepsilon < 1/3$ and*

$$\delta < 1 - (1 - \lambda + \varepsilon)^{1/3}. \tag{10}$$

Let $\boldsymbol{u} = (u_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m}$ such that

$$\Pr_{\boldsymbol{p}\in\mathbb{F}_q^m}\left[\Delta\left(\sum_{\boldsymbol{e}\in\{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} u_{\boldsymbol{e}}, C\right) < \delta\right] \geqslant \frac{2m}{\varepsilon^2 q}. \tag{11}$$

Then there exist $T \subset D$ and a family $\boldsymbol{v} = (v_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m} \in C^{2^m}$ such that

- *$|T| \geqslant (1 - \delta - m\varepsilon)\,|D|$,*
- *for each $\boldsymbol{e} \in \{0,1\}^m$, $u_{\boldsymbol{e}|T} = v_{\boldsymbol{e}|T}$.*

Proof. We proceed by induction on the number of variables m. The case $m = 1$ is dealt with in [BGKS20, Lemma 3.2]. Let us assume that the proposition is true for $m-1$ and prove that it also holds for m. For $\boldsymbol{p} \in \mathbb{F}_q^m$, we write $\boldsymbol{p} = (\tilde{\boldsymbol{p}}, p_m)$, with $\tilde{\boldsymbol{p}} \in \mathbb{F}_q^{m-1}$ and $p_m \in \mathbb{F}_q$. Similarly, for $\boldsymbol{e} \in \{0,1\}^m$, we write $\boldsymbol{e} = (\tilde{\boldsymbol{e}}, e_m)$, with $\tilde{\boldsymbol{e}} \in \{0,1\}^{m-1}$ and $e_m \in \{0,1\}$. Equation (11) gives

$$\Pr_{p_m\in\mathbb{F}_q}\left[\Pr_{\tilde{\boldsymbol{p}}\in\mathbb{F}_q^{m-1}}\left[\Delta\left(\sum_{\tilde{\boldsymbol{e}}\in\{0,1\}^{m-1}} \tilde{\boldsymbol{p}}^{\tilde{\boldsymbol{e}}}\left(u_{(\tilde{\boldsymbol{e}},0)} + p_m u_{(\tilde{\boldsymbol{e}},1)}\right), C\right) < \delta\right] \geqslant \frac{2(m-1)}{\varepsilon^2 q}\right] \geqslant \frac{2}{\varepsilon^2 q}.$$

For any $z \in \mathbb{F}_q$, we write $u_{\tilde{\boldsymbol{e}},z} = u_{(\tilde{\boldsymbol{e}},0)} + z u_{(\tilde{\boldsymbol{e}},1)}$. Let A be the set

$$A = \left\{ z \in \mathbb{F}_q;\ \Pr_{\tilde{\boldsymbol{p}}\in\mathbb{F}_q^{m-1}}\left[\Delta\left(\sum_{\tilde{\boldsymbol{e}}\in\{0,1\}^{m-1}} \tilde{\boldsymbol{p}}^{\tilde{\boldsymbol{e}}} u_{\tilde{\boldsymbol{e}},z}, C\right) < \delta\right] \geqslant \frac{2(m-1)}{\varepsilon^2 q} \right\}.$$

By assumption, $|A| \geqslant 2/\varepsilon^2$. Moreover the inductive hypothesis implies that for each $z \in A$, there exist $T_z \subset D$ and $v_{\tilde{\boldsymbol{e}},z} \in C$ such that

$$|T_z| \geqslant (1 - \delta - (m-1)\varepsilon)\,|D| \ \text{ and } u_{\tilde{\boldsymbol{e}},z|T_z} = v_{\tilde{\boldsymbol{e}},z|T_z} \quad \text{for all } \tilde{\boldsymbol{e}} \in \{0,1\}^{m-1}.$$

We are now in a position where we can mimic the proof of [BGKS20].

Let us prove there exists a large subset $A' \subset A$ such that for all $\tilde{\boldsymbol{e}} \in \{0,1\}^{m-1}$ and for all $z \in A'$, $v_{\tilde{\mathbf{e}},z}$ depends linearly on z, *i.e.* there exists some $v_{(\tilde{\mathbf{e}},0)}, v_{(\tilde{\mathbf{e}},1)} \in C$ such that $v_{\tilde{\mathbf{e}},z} = v_{(\tilde{\mathbf{e}},0)} + zv_{(\tilde{\mathbf{e}},1)}$.

For z_0, z_1, z_2, picked uniformly and independently in A and y picked uniformly from D, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{z_0,z_1,z_2}\left[\frac{|T_{z_0} \cap T_{z_1} \cap T_{z_2}|}{|D|}\right] &= \mathop{\mathbf{E}}_{y,z_0,z_1,z_2}\left[\mathbf{1}_{y \in T_{z_0} \cap T_{z_1} \cap T_{z_2}}\right] \\
&= \mathop{\mathbf{E}}_{y}\left[\mathop{\mathbf{E}}_{z}\left[\mathbf{1}_{y \in T_z}\right]^3\right] \\
&\geqslant \mathop{\mathbf{E}}_{y,z}\left[\mathbf{1}_{y \in T_z}\right]^3 \\
&\geqslant (1-\delta)^3 \\
&> 1-\lambda+\varepsilon.
\end{aligned}$$

From this, one obtains:

$$\Pr_{z_0,z_1,z_2}\left[|T_{z_0} \cap T_{z_1} \cap T_{z_2}| \geqslant (1-\lambda)|D|\right] \geqslant \varepsilon.$$

The probability of z_0, z_1, z_2 being all distinct is at least $1 - \frac{3}{|A|}$, which is greater than $1 - \frac{\varepsilon}{2}$ since $|A| \geqslant \frac{2}{\varepsilon^2} > \frac{6}{\varepsilon}$. Thus, we get

$$\Pr_{z_0,z_1,z_2}\left[z_0, z_1, z_2 \text{ are all distinct and } |T_{z_0} \cap T_{z_1} \cap T_{z_2}| \geqslant (1-\lambda)|D|\right] \geqslant \varepsilon/2.$$

Consequently, there are distinct z_1 and z_2 such that

$$\Pr_{z_0}\left[|T_{z_0} \cap T_{z_1} \cap T_{z_2}| \geqslant (1-\lambda)|D|\right] \geqslant \varepsilon/2.$$

Fix $z_0 \in \mathbb{F}_q$ such that $|T_{z_0} \cap T_{z_1} \cap T_{z_2}| \geqslant (1-\lambda)|D|$ and set $S = T_{z_0} \cap T_{z_1} \cap T_{z_2}$. For each $\tilde{\boldsymbol{e}} \in \{0,1\}^{m-1}$, the vectors

$$(z_0, u_{\tilde{\mathbf{e}},z_0}), (z_1, u_{\tilde{\mathbf{e}},z_1}), (z_2, u_{\tilde{\mathbf{e}},z_2})$$

are collinear. Then their restrictions to S, $\left(z_i, u_{\tilde{\mathbf{e}},z_i|S}\right)$, which coincide with $\left(z_i, v_{\tilde{\mathbf{e}},z_i|S}\right)$ by definition of S, are also collinear. Since S is an information set of C, we can linearly map the vectors $v_{\tilde{\mathbf{e}},z_i|S}$ to elements $v_{\tilde{\mathbf{e}},z_i}$ of the code C, which preserves collinearity. Therefore, the vectors $v_{\tilde{\mathbf{e}},z_i}$ $(z = z_0, z_1, z_2)$ all belong to a same line

$$\left\{v_{(\tilde{\mathbf{e}},0)} + zv_{(\tilde{\mathbf{e}},1)}; z \in \mathbb{F}_q\right\} \subset \mathbb{F}_q^D \text{ where } v_{(\tilde{\mathbf{e}},0)}, v_{(\tilde{\mathbf{e}},1)} \in C.$$

Set $A' = \left\{z \in A \mid v_{\tilde{\mathbf{e}},z} = v_{(\tilde{\mathbf{e}},0)} + zv_{(\tilde{\mathbf{e}},1)}\right\}$. Then we have $|A'| \geqslant \frac{\varepsilon}{2}|A| \geqslant \frac{1}{\varepsilon}$. Now consider the set

$$T = \left\{x \in D | \forall \tilde{\boldsymbol{e}} \in \{0,1\}^{m-1},\ u_{(\tilde{\mathbf{e}},0)}(x) = v_{(\tilde{\mathbf{e}},0)}(x) \text{ and } u_{(\tilde{\mathbf{e}},1)}(x) = v_{(\tilde{\mathbf{e}},1)}(x)\right\}.$$

For any $x \in D \backslash T$, there exists at most one $z \in \mathbb{F}_q$ such that, for all $\tilde{\boldsymbol{e}} \in \{0,1\}^{m-1}$,

$$u_{(\tilde{\mathbf{e}},0)}(x) + zu_{(\tilde{\mathbf{e}},1)}(x) = v_{(\tilde{\mathbf{e}},0)}(x) + zv_{(\tilde{\mathbf{e}},1)}(x).$$

For any $z \in A'$, for any $\tilde{\boldsymbol{e}} \in \{0,1\}^{m-1}$, we have

$$1 - \frac{|T_z|}{|D|} \geqslant \Delta_D(u_{\tilde{\mathbf{e}},z}, v_{\tilde{\mathbf{e}},z}).$$

We thus also have

$$\begin{aligned}
1-\frac{|T_z|}{|D|} &\geqslant \underset{z\in A'}{\mathbf{E}}\left[\Delta_D(u_{\tilde{\boldsymbol{e}},z}, v_{\tilde{\boldsymbol{e}},z})\right] \\
&\geqslant \frac{|D\backslash x|}{|D|}\left(1-\frac{1}{|A'|}\right) \\
&\geqslant \left(1-\frac{|T|}{|D|}\right)(1-\varepsilon) \\
&\geqslant 1-\frac{|T|}{|D|}-\varepsilon
\end{aligned}$$

Using $|T_z| \geqslant (1-\delta-(m-1)\varepsilon)\,|D|$, and rearranging, we get $|T| \geqslant (1-\delta-m\varepsilon)\,|D|$. □

5.2 Weighted agreement version

For soundness analysis, we need a variant of Proposition 3 stated in terms of weighted agreement. This technical result will be used to prove distance preservation properties in Section 7 and Section 9.

Proposition 4. *Let m be a positive integer. Let $C \subset \mathbb{F}_q^D$ be a linear code of distance $\lambda = \Delta(C)$. Let $\varepsilon, \delta > 0$ such that $\varepsilon < 1/3$ and*

$$\delta < 1-(1-\lambda+\varepsilon)^{1/3}.$$

For any weight function $\phi\colon D \to [0,1]$ and any $\boldsymbol{u} = (u_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m}$ satisfying

$$\Pr_{\boldsymbol{p}\in\mathbb{F}_q^m}\left[\mu_\phi\left(\sum_{\boldsymbol{e}\in\{0,1\}^m}\boldsymbol{p}^{\boldsymbol{e}}u_{\boldsymbol{e}}, C\right) > 1-\delta\right] \geqslant \frac{2m}{\varepsilon^2 q}, \tag{12}$$

there exist $T \subset D$ and a family $\boldsymbol{v} = (v_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m} \in C^{2^m}$ such that

- $\sum_{x\in T}\phi(x) \geqslant (1-\delta-m\varepsilon)\,|D|$,
- *for each $\boldsymbol{e} \in \{0,1\}^m$, $u_{\boldsymbol{e}|T} = v_{\boldsymbol{e}|T}$.*

Before proving Proposition 4, we first state a variant of [BGKS20, Lemma 3.2]. The proof of Lemma 5 is relatively straigthforward, based on the original proof of [BGKS20, Lemma 3.2]. We provide it in Appendix A for completeness.

Lemma 5 ([BGKS20])**.** *Let $C \subset \mathbb{F}_q^D$ be a linear code of distance $\lambda = \Delta(C)$. Let $\varepsilon, \delta > 0$ such that $\varepsilon < 1/3$ and*

$$\delta < 1-(1-\lambda+\varepsilon)^{1/3}.$$

For any weight function $\phi\colon D \to [0,1]$ and any functions $u_0, u_1 \in \mathbb{F}_q^D$ satisfying

$$\Pr_{z\in\mathbb{F}_q}\left[\mu_\phi(u_0+zu_1, C) > 1-\delta\right] \geqslant \frac{2}{\varepsilon^2 q}, \tag{13}$$

there exist $T \subset D$ and $v_0, v_1 \in C$, such that

- $\sum_{x\in T}\phi(x) \geqslant (1-\delta-\varepsilon)\,|D|$,
- *for each $i \in \{0,1\}$, $u_{i|T} = v_{i|T}$.*

Proof of Proposition 4. As for Proposition 3, we proceed by induction on m. The case $m = 1$ is treated by Lemma 5. Let us assume that the statement is true for $m-1$.

Observe that if the function $\phi\colon D \to [0,1]$ is constant equal to 1, then $\mu_\phi(u,v) = 1 - \Delta(u,v)$. Therefore, for any weight function $\phi\colon D \to [0,1]$ and any $u, v \in \mathbb{F}_q^D$, $\mu_\phi(u,v) \leqslant 1 - \Delta(u,v)$. Consequently, $\mu_\phi(u,C) \leqslant 1 - \Delta(u,C)$.

Thus we get from (12):

$$\left\{ \boldsymbol{p} \in \mathbb{F}_q^m \mid \mu_\phi\left(\sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} u_{\boldsymbol{e}}, C \right) > 1 - \delta \right\} \subseteq \left\{ \boldsymbol{p} \in \mathbb{F}_q^m \mid \Delta\left(\sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} u_{\boldsymbol{e}}, C \right) < \delta \right\}.$$

The latter set has size at least $\frac{2m}{\varepsilon^2} q^{m-1}$. Then, the proof follows the proof of Proposition 3, until we get a set $A' \subset A$ of size at least $1/\varepsilon$ and $v_{(\boldsymbol{a},0)}, v_{(\boldsymbol{a},1)} \in C$ such that for all $\boldsymbol{a} \in \{0,1\}^{m-1}$, for all $z \in A'$, $v_{\boldsymbol{a},z} = v_{(\boldsymbol{a},0)} + z v_{(\boldsymbol{a},1)}$.

Let T be the set

$$T = \left\{ x \in D |\text{ for all } \boldsymbol{a} \in \{0,1\}^{m-1},\ u_{(\boldsymbol{a},0)}(x) = v_{(\boldsymbol{a},0)}(x) \text{ and } u_{(\boldsymbol{a},1)}(x) = v_{(\boldsymbol{a},1)}(x) \right\}.$$

For all $\boldsymbol{a} \in \{0,1\}^{m-1}$, for all $z \in A'$, $\Delta(u_{(\boldsymbol{a},0)} + z u_{(\boldsymbol{a},1)}, v_{(\boldsymbol{a},0)} + z v_{(\boldsymbol{a},1)}) < \delta + (m-1)\varepsilon$. Still noting $u_{\boldsymbol{a},z} = u_{\boldsymbol{a},0} + z u_{\boldsymbol{a},1}$ and $v_{\boldsymbol{a},z} = v_{\boldsymbol{a},0} + z v_{\boldsymbol{a},1}$, we get $\mu_\phi(u_{\boldsymbol{a},z}, v_{\boldsymbol{a},z}) > 1 - \delta - (m-1)\varepsilon$. We have:

$$\begin{aligned} 1 - \delta - (m-1)\varepsilon &< \frac{1}{|A'|} \sum_{z \in A'} \mu_\phi(u_{\boldsymbol{a},z}, v_{\boldsymbol{a},z}) \\ &< \frac{1}{|A'|\,|D|} \sum_{z \in A'} \sum_{x \in D} \left(\phi(x) \cdot \mathbf{1}_{u_{\boldsymbol{a},z}(x) = v_{\boldsymbol{a},z}(x)} \right) \\ &< \frac{1}{|D|} \sum_{x \in D} \phi(x) \cdot \left(\frac{1}{|A'|} \sum_{z \in A'} \mathbf{1}_{u_{\boldsymbol{a},z}(x) = v_{\boldsymbol{a},z}(x)} \right). \end{aligned}$$

For $x \in D \backslash T$, there is at most one element $z \in \mathbb{F}_q$ such that $u_{(\boldsymbol{a},0)}(x) + z u_{(\boldsymbol{a},1)}(x) = v_{(\boldsymbol{a},0)}(x) + z v_{(\boldsymbol{a},1)}(x)$. Thus, we get

$$\begin{aligned} 1 - \delta - (m-1)\varepsilon &< \frac{1}{|D|} \sum_{x \in T} \phi(x) + \frac{1}{|D|} \sum_{x \in D \backslash T} \phi(x) \frac{1}{|A'|} \\ &< \frac{1}{|D|} \sum_{x \in T} \phi(x) + \varepsilon. \end{aligned}$$

Rearranging, we have $\sum_{x \in T} \phi(x) > (1 - \delta - m\varepsilon)\,|D|$. □

6 Sequence of evaluation domains defined by two-to-one maps

In this section, we provide a common notation for two different settings, depending on the algebraic nature of the evaluation domain L. The first one will be prime fields which admit a 2-smooth multiplicative subgroup. The second one will be fields of characteristic two. These two settings also appear in [BS08, BBHR18] in the context of proximity testing to Reed-Solomon codes.

6.1 Case of a smooth multiplicative group

Let us assume that $\mathbb{F}_q$ is a prime field and $q-1$ is divisible by a power of two, i.e. $q = a \cdot 2^n + 1$ for some positive integers a and n. We will consider $L_0 \subset \mathbb{F}_q$ a cyclic multiplicative group of order 2^n. For any integer r, we define a sequence of evaluation sets $(L_i)_{0 \leqslant i \leqslant r}$ as: $L_{i+1} := q_i(L_i)$ where $q_i(X) = X^2$. Let $A_i \subset L_i$ a multiplicative subgroup of L_i of size 2, each multiplicative coset of A_i is mapped to a single element of L_{i+1} by the map $x \mapsto q_i(x)$.

6.2 Case of an affine subspace in characteristic 2

If $\mathbb{F}_q$ has characteristic two, we consider $L_0 \subset \mathbb{F}_q$ an affine subspace over $\mathbb{F}_2$ of dimension n. Let $A_i \subset L_0$ be an $\mathbb{F}_2$-affine subspace with $\dim A_i = 1$. Define $q_i(X) := \prod_{a \in A_i} (X - a)$. Then $q_i(X)$ is a so-called *subspace polynomial*, also known as *linearized polynomials* when A_i is a vector space. It has the form $X^2 + \alpha X + \beta$ for $\alpha, \beta \in \mathbb{F}_q$, and each additive coset of A_i is mapped to a single element of L_{i+1} by the map $x \mapsto q_i(x)$, and $\dim L_{i+1} = \dim L_i - \dim A_i = \dim L_i - 1$. For more on affine and linearized polynomials, see [LN97, Section 3.4].

6.3 Common properties

In both cases, we have that $|L_{i+1}| = \frac{1}{2}|L_i| = \frac{1}{2^i}|L_0|$. Moreover, the map $\pi_i \colon L_i^m \to L_{i+1}^m$ defined by $\pi_i(\boldsymbol{x}) := (q_i(x_1), \ldots, q_i(x_m))$ is 2^m-to-1 on its domain.

A crucial ingredient of the constructions presented in the two next sections will be the following fact: if $f \colon L_i^m \to \mathbb{F}_q$ corresponds to the evaluation of a polynomial $\widehat{f} \in \mathbb{F}_q[\boldsymbol{X}]$ of bounded degree, then Proposition 2 gives a decomposition of f in terms of functions $(g_{\boldsymbol{e}} \circ \pi_i)_{\boldsymbol{e} \in \{0,1\}^m}$ where $g_{\boldsymbol{e}} : L_{i+1}^m \to \mathbb{F}_q$ is the evaluation of a polynomial of half degree.

Remark 4. *The choice to consider degree-2 maps q_i is intended to simplify the exposition. Recall that Proposition 2 is stated for q_i of arbitrary degree l. After examining proofs of Sections 7 and 9, one can see that the generalization to maps of higher degree is also valid.*

7 A first IOP of Proximity for tensor products of Reed-Solomon codes

Based on the decomposition given in Proposition 2, we present a first construction for tensor prod uct codes, then we will show how efficiency parameters can be improved by increasing the number of rounds and defining the folding operators differently.

7.1 Sequence of codes

Let k be a power of two and set $r = \log_2 k$. As suggested in Section 6, depending on whether we work in case 6.1 or 6.2, consider $L \subset \mathbb{F}_q$ of size $|L| > k$ which is either a cyclic group of order a power of two, or an affine subspace over $\mathbb{F}_2$. We will use the notations introduced in Section 6 and will consider $L_0 = L, L_1, \ldots, L_r$ as defined there.

Set $k_0 := k$. For $0 < i \leqslant r$, define $k_{i+1} := \frac{k_i}{2}$. In particular, for all i, we have $k_i < |L_i|$. In the sequel, we denote by $(\mathsf{RS}_i^m)_{0 \leqslant i \leqslant r}$ where RS_i^m the sequence of tensor product of RS codes refers to the code $(\mathsf{RS}[\mathbb{F}_q, L_i, k_i])^{\otimes m}$, regardless we are in case 6.1 or 6.2.

Notice that, for all $i \in [0, r]$, we have $k_i < |L_i|$. Moreover, each code RS_i^m has same rate $R := \left(\frac{k}{|L|}\right)^m$. The relative distances of the codes $\mathsf{RS}_0^m, \ldots, \mathsf{RS}_r^m$ are greater than $\left(1 - \frac{k}{|L|}\right)^m$.

7.2 Folding operators

For each code RS_i^m, $0 \leqslant i < r$, we define a family of folding operators satisfying the distance preservation property. They will enable us to iteratively reduce the problem of proximity testing to a code RS_i^m to a problem of size 2^m times smaller, namely proximity testing to RS_{i+1}^m.

Definition 9 (Folding operators)**.** *Let $i \in [0, r-1]$. Let $f\colon L_i^m \to \mathbb{F}_q$ be an arbitrary function and let $\widehat{f}$ be its low-degree extension. Let $(\widehat{g}_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m}$ be the 2^m m-variate polynomials provided by Proposition 2 applied to $\widehat{f}$. We consider their evaluations on L_{i+1}^m, respectively denoted by $g_{\boldsymbol{e}}$. For any $\boldsymbol{p} \in \mathbb{F}_q^m$, we define the* folding of f $\textsf{\textbf{Fold}}[f, \boldsymbol{p}]$ *as the following function:*

$$\textsf{\textbf{Fold}}[f,\boldsymbol{p}] : \begin{cases} L_{i+1}^m & \to \quad \mathbb{F}_q, \\ \boldsymbol{y} & \mapsto \sum_{\boldsymbol{e}\in\{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} g_{\boldsymbol{e}}(\boldsymbol{y}). \end{cases} \tag{14}$$

First, we show that this defines a folding operator for the code RS_i^m as per Definition 5.

Lemma 6 (Completeness)**.** *For any $\boldsymbol{p} \in \mathbb{F}_q^m$, if $f \in \mathsf{RS}_i^m$, then $\textsf{\textbf{Fold}}[f,\boldsymbol{p}] \in \mathsf{RS}_{i+1}^m$.*

Proof. Proposition 2 shows that, for all $\boldsymbol{e} \in \{0,1\}^m$ and all $j \in [1, m]$, $\deg_{X_j} \widehat{g}_{\boldsymbol{e}} \leqslant \left\lfloor \frac{k_i - 1}{2} \right\rfloor$, which is strictly less than k_{i+1} since k_i is even. □

Lemma 7 (Locality)**.** *Let $f\colon L_i^m \to \mathbb{F}_q$ be an arbitrary function and let $\boldsymbol{p} \in \mathbb{F}_q^m$. The value of $\textsf{\textbf{Fold}}[f,\boldsymbol{p}]$ at any $\boldsymbol{y} \in L_{i+1}^m$ can be computed with exactly 2^m queries to f.*

Proof. Take $\boldsymbol{y} = (y_1, \ldots, y_m) \in L_{i+1}^m$. For each $j \in [1 \,..\, m]$, define $S_{y_j} \subset L_i$ the coset of A_i such that $q_i(S_{y_j}) = y_j$ (i.e. S_{y_j} is the set of roots of the polynomial $q_i(X) - y_j$). Set $S_{\boldsymbol{y}} = \prod_{j=1}^m S_{y_j}$ and consider $P_{f,\boldsymbol{y}} \in \mathbb{F}_q[\boldsymbol{X}]$ the unique low-degree extension of $f_{|S_{\boldsymbol{y}}}$.

Let us prove that for all $\boldsymbol{p} \in \mathbb{F}_q^m$, we have $P_{f,\boldsymbol{y}}(\boldsymbol{p}) = \textsf{Fold}[f,\boldsymbol{p}](\boldsymbol{y})$, which would induce that the value of $\textsf{Fold}[f,\boldsymbol{p}](\boldsymbol{y})$ can be computed by interpolating the set of points $\{(\boldsymbol{x}, f(\boldsymbol{x})),\ \boldsymbol{x} \in S_{\boldsymbol{y}}\}$ of size 2^m.

By Lemma 4, one can write

$$\widehat{f}(\boldsymbol{X}) = \sum_{\boldsymbol{u}\in U} \widehat{f}_{\boldsymbol{u}}(\boldsymbol{X}) q_i(X_1)^{u_1} \cdots q_i(X_m)^{u_m}$$

with for all $\boldsymbol{u} \in U$ and $j \in [1, m]$, $\deg_{X_j} f_{\boldsymbol{u}} < 2$. Since the polynomial $\widehat{f}(\boldsymbol{X})$ and $P_{f,\boldsymbol{y}}(\boldsymbol{X})$ agree on $S_{\boldsymbol{y}}$, we get that

$$\widehat{f}(\boldsymbol{X}) = P_{f,\boldsymbol{y}}(\boldsymbol{X}) \mod (q_i(X_1) - y_1, \ldots, q_i(X_m) - y_m).$$

By definition of the low-degree extension, $\deg_{X_j} P_{f,\boldsymbol{y}} < 2$ for all j, thus

$$P_{f,\boldsymbol{y}}(\boldsymbol{X}) = \sum_{\boldsymbol{u}\in U} \widehat{f}_{\boldsymbol{u}}(\boldsymbol{X}) \boldsymbol{y}^u.$$

For each $\boldsymbol{u} \in U$, write each polynomial $f_{\boldsymbol{u}}$ as $f_{\boldsymbol{u}}(\boldsymbol{X}) = \sum_{\boldsymbol{e}\in\{0,1\}^m} a_{\boldsymbol{u},\boldsymbol{e}} \boldsymbol{X}^{\boldsymbol{e}}$. Proof of Proposition 2 shows that each polynomial $\widehat{g}_{\boldsymbol{e}}$ is equal to $\sum_{\boldsymbol{u}\in U} a_{\boldsymbol{u},\boldsymbol{e}} \boldsymbol{X}^{\boldsymbol{u}}$. Therefore, for all $\boldsymbol{y} \in L_{i+1}^m$, we have

$$P_{f,\boldsymbol{y}}(\boldsymbol{X}) = \sum_{\boldsymbol{e}\in\{0,1\}^m} \boldsymbol{X}^{\boldsymbol{e}} \widehat{g}_{\boldsymbol{e}}(\boldsymbol{y}).$$

Finally, for all $\boldsymbol{p} \in \mathbb{F}_q^m$ and $\boldsymbol{y} \in L_{i+1}^m$, the evaluation of $\textsf{Fold}[f,\boldsymbol{p}]$ at $\boldsymbol{y}$ can be obtained by evaluating $P_{f,\boldsymbol{y}}$ at $\boldsymbol{p}$. □

Let us now show that Definition 9 satisfies distance preservation (Definition 7).

Proposition 5 (Distance preservation)**.** *Let $f_i \colon L_i^m \to \mathbb{F}_q$ be an arbitrary function. Let $\varepsilon \in \left(0, \frac{1}{3}\right)$ and $\delta < 1 - (1 - \lambda + \varepsilon)^{\frac{1}{3}}$. Let $\phi_i \colon L_i^m \to [0,1]$ and $\phi_{i+1} \colon L_{i+1}^m \to [0,1]$ be weight functions such that*

$$\forall \boldsymbol{y} \in L_{i+1}^m,\ \phi_{i+1}(\boldsymbol{y}) \leqslant \frac{1}{2^m} \sum_{\boldsymbol{x} \in S_{\boldsymbol{y}}} \phi_i(\boldsymbol{x}).$$

If $f \colon L_i^m \to \mathbb{F}_q$ has weighted agreement $\mu_{\phi_i}(f, \mathsf{RS}_i^m) < 1 - \delta$, then

$$\Pr_{\boldsymbol{p} \in \mathbb{F}_q^m} \left[\mu_{\phi_{i+1}}(\textsf{\textbf{\textit{Fold}}}\,[f, \boldsymbol{p}], \mathsf{RS}_{i+1}^m) > 1 - \delta + m\varepsilon\right] < \frac{2m}{\varepsilon^2 q}.$$

Proof. We proceed by contraposition and we assume

$$\Pr_{\boldsymbol{p} \in \mathbb{F}_q^m} \left[\mu_{\phi_{i+1}}(\textsf{\textbf{Fold}}\,[f, \boldsymbol{p}], \mathsf{RS}_{i+1}^m) > 1 - \delta + m\varepsilon\right] \geqslant \frac{2m}{\varepsilon^2 q}.$$

Applying Proposition 4 on $\textsf{\textbf{Fold}}\,[f, \boldsymbol{p}] = \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} g_{\boldsymbol{e}}$, we get that there exist $T \subset L_{i+1}^m$ and $(v_{\boldsymbol{e}})_{\boldsymbol{e} \in \{0,1\}^m}$, $v_{\boldsymbol{e}} \in \mathsf{RS}_{i+1}^m$, satisfying

- $\displaystyle\sum_{\boldsymbol{y} \in T} \phi_{i+1}(\boldsymbol{y}) \geqslant (1 - \delta) \left|L_{i+1}^m\right|$,
- for all $\boldsymbol{e} \in \{0,1\}^m$, $g_{\boldsymbol{e}\,|T} = v_{\boldsymbol{e}\,|T}$.

For each $\boldsymbol{e} \in \{0,1\}^m$, let us consider $\widehat{v}_{\boldsymbol{e}} \in \mathbb{F}_q[\boldsymbol{Y}]$ the polynomial of individual degrees less than k_{i+1} associated with the codeword $v_{\boldsymbol{e}} \in \mathsf{RS}_{i+1}^m$.

Let R be the polynomial defined by

$$R(\boldsymbol{X}) \coloneqq \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{X}^{\boldsymbol{e}} \widehat{v}_{\boldsymbol{e}}(q_i(X_1), \ldots, q_i(X_m))$$

and v be the evaluation of R on L_i^m.

Since $k_{i+1} \leqslant k_i/2$, we have $\deg_{X_j} R \leqslant 1 + 2 \cdot (k_{i+1} - 1) < k_i$, hence $v \in \mathsf{RS}_i^m$. For all $\boldsymbol{y} \in T$ and $\boldsymbol{x} \in S_{\boldsymbol{y}}$, i.e. $\pi(\boldsymbol{x}) = \boldsymbol{y}$, $v(\boldsymbol{x}) = \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{x}^{\boldsymbol{e}} v_{\boldsymbol{e}}(\pi(\boldsymbol{x}))$ and

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{x}^{\boldsymbol{e}} \widehat{g}_{\boldsymbol{e}}(\boldsymbol{y}) = \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{x}^{\boldsymbol{e}} g_{\boldsymbol{e}}(\boldsymbol{y}) = \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{x}^{\boldsymbol{e}} v_{\boldsymbol{e}}(\boldsymbol{y}) = v(\boldsymbol{x}). \tag{15}$$

Thus v agrees with f on $S_T \coloneqq \bigsqcup_{\boldsymbol{y} \in T} S_{\boldsymbol{y}}$. Since $v \in \mathsf{RS}_i^m$, we have

$$\mu_{\phi_i}(f, \mathsf{RS}_i^m) \geqslant \frac{1}{|L_i^m|} \sum_{\boldsymbol{x} \in S_T} \phi_i(\boldsymbol{x}) = \frac{1}{|L_i^m|} \sum_{\boldsymbol{y} \in T} \sum_{\boldsymbol{x} \in S_{\boldsymbol{y}}} \phi_i(\boldsymbol{x}) \geqslant \frac{1}{\left|L_{i+1}^m\right|} \sum_{\boldsymbol{y} \in T} \phi_{i+1}(\boldsymbol{y}).$$

Eventually, we conclude that $\mu_{\phi_i}(f, \mathsf{RS}_i^m) \geqslant 1 - \delta$ by definition of T. This contradicts the hypothesis on f. $\square$

7.3 IOPP for tensor product of RS codes by dividing the length by 2^m

Given a sequence of codes $(\mathsf{RS}_i^m)_{0\leqslant i\leqslant r}$ as defined in Section 7.1 and a family of folding operators for each code RS_i^m (see Section 7.2), the generic construction described in Section 3.2 leads to a public-coin IOPP $(\mathcal{P}_{\mathsf{RS}^m}, \mathcal{V}_{\mathsf{RS}^m})$ for the code RS_0^m.

Notice that the last function f_r is supposed to be constant. Therefore, we use the variant of the protocol described in Remark 2. Specifically, instead of sending f_r during the COMMIT phase, the prover $\mathcal{P}_{\mathsf{RS}^m}$ sends a single field element $\beta \in \mathbb{F}_q$. The verifier $\mathcal{V}_{\mathsf{RS}^m}$ does not run a membership test to C_r but checks the equation $\beta = \textbf{Fold}\left[f_{r-1}, \boldsymbol{p}_{r-1}\right](\boldsymbol{y}_r)$.

The properties of the resulting IOPP system $(\mathcal{P}_{\mathsf{RS}^m}, \mathcal{V}_{\mathsf{RS}^m})$ are displayed in the following theorem.

Theorem 3. *Let k, m be positive integers such that $k > 1$ is a power of two. Let $L \subset \mathbb{F}_q^{\times}$ as described in Section 6 such that $k < |L|$. Then, the generic construction of Section 3.2 leads to public-coin IOPP system $(\mathcal{P}_{\mathsf{RS}^m}, \mathcal{V}_{\mathsf{RS}^m})$ for the tensor product code $(\mathsf{RS}\left[\mathbb{F}_q, L, k\right])^{\otimes m}$ of blocklength n^m with the following properties.*

1. **Round complexity** *is $r(n^m) < \log n$.*
2. **Query complexity** *is $q(n^m) < \alpha 2^m \log n + 1$ for α repetitions of the QUERY phase.*
3. **Proof length** *is $l(n^m) < \frac{n^m}{2^m-1}$.*
4. **Prover complexity** *is $t_p(n^m) < 4(m+2)n^m$.*
5. **Verifier decision complexity** *is $t_v(n^m) < 4\alpha(2^m+m)\log n$.*
6. **Perfect completeness:** *If $f \in (\mathsf{RS}\left[\mathbb{F}_q, L, k\right])^{\otimes m}$ and if the oracles $f_1, \dots f_r$ are computed by an honest prover $\mathcal{P}_{\mathsf{RS}^m}$, then $\mathcal{V}_{\mathsf{RS}^m}$ outputs **accept** with probability 1.*
7. **Soundness:** *Assume that $f\colon L^m \to \mathbb{F}_q$ is δ-far from $(\mathsf{RS}\left[\mathbb{F}_q, L, k\right])^{\otimes m}$. Denote $\lambda = \left(1 - \frac{k}{|L|}\right)$ and, for any $\varepsilon \in \left(0, \frac{1}{3}\right)$, set $\gamma(\lambda, \varepsilon) := 1 - (1 - \lambda + \varepsilon)^{1/3}$. Then, for any unbounded prover $\mathcal{P}^*$, the verifier $\mathcal{V}_{\mathsf{RS}^m}$ outputs accept after α repetitions of the QUERY phase with probability at most*
$$\frac{2m\log n}{\varepsilon^2 q} + (1 - \min(\delta, \gamma(\varepsilon, \lambda)) + \varepsilon m \log n)^{\alpha}.$$

Proof. We apply the construction of the public-coin IOPP system presented in Section 3.2 with the family of folding operators defined in Section 7.2. Completeness and soundness follow from Theorem 2. The number of round is $r = \log k < \log |L|$ by definition. For a single repetition of the query test, $\mathcal{V}_{\mathsf{RS}^m}$ queries each oracle f_i, $i \in [0\mathinner{..}r-1]$, at 2^m locations. The verifier retrieves β a single time, which yields the claimed query complexity.

The total proof length is
$$\sum_{i=1}^{r} |L_i^m| = \sum_{i=1}^{r} \frac{n^m}{2^{mi}} < \frac{n^m}{2^m - 1}.$$
We examine prover complexity. Let $f\colon L_i^m \to \mathbb{F}_q$ and $\boldsymbol{p} \in \mathbb{F}_q^m$. For each $\boldsymbol{y} \in L_{i+1}^m$, the prover evaluates the low-degree extension $P_{f,\boldsymbol{y}}(\boldsymbol{X})$ of $f_{|S_{\boldsymbol{y}}}$ at $\boldsymbol{p}$, where $S_{\boldsymbol{y}} = \pi_i^{-1}(\{\boldsymbol{y}\})$. It follows from Lemma 2 that the number of operations to evaluate $\textbf{Fold}\left[f, \boldsymbol{p}\right]$ on L_{i+1}^m is $4(2^m+m)\left|L_{i+1}^m\right|$. We deduce that the cost of honestly generating $\mathcal{P}_{\mathsf{RS}^m}$'s messages is
$$\sum_{i=1}^{r} 4(2^m+m)\left|L_{i+1}^m\right| < 4(2^m+m)\frac{n^m}{2^m-1} \leqslant 4(m+2)n^m.$$
We also deduce from Lemma 2 that the verifier complexity is less than $\alpha \sum_{i=1}^{r} 4(2^m+m)$. ☐

8 IOP of Proximity for tensor products of RS codes by folding with respect to each variable

The construction presented in this section essentially consists in applying the FRI protocol [BBHR18] to each variable. We use the possibilty of folding with respect to a single indeterminate, instead of folding along all the indeterminates at once. We call this *partial folding.*

Let us first present the idea for the case where $m = 2$ and L is a multiplicative subgroup of a field of odd characteristic. Given a function $f\colon L^2 \to \mathbb{F}_q$, and $\widehat{f}(X_1, X_2)$ its associated low-degree extension, we can decompose it as

$$\widehat{f}_1(X_1, X_2) = \widehat{g}_0(X_1^2, X_2) + X_1\widehat{g}_1(X_1^2, X_2).$$

For $a \in \mathbb{F}$, the notation $\mathsf{Fold}_{X_1}[f, a] : q(L) \times L \to \mathbb{F}_q$ will refer to the function whose low-degree extension is the polynomial

$$\widehat{g}_0(X_1, X_2) + z\widehat{g}_1(X_1, X_2).$$

For any $y \in q(L)$ and $a \in \mathbb{F}_q$, one can compute $\mathsf{Fold}_{X_1}[f, a](y)$ from exactly two entries of f. Such a folding operator $\mathsf{Fold}_{X_1}[\cdot, \cdot]$ will allow us to reduce a problem of proximity to a code $\mathsf{RS}[\mathbb{F}_q, L, k]^{\otimes 2}$ to a similar but smaller problem, which is associated to the code

$$\mathsf{RS}[\mathbb{F}_q, q(L), k/2] \otimes \mathsf{RS}[\mathbb{F}_q, L, k].$$

Remark 5. *We can also write*

$$\widehat{f}(X_1, X_2) = \widehat{h}_0(X_1, X_2^2) + X_2\widehat{h}_1(X_1, X_2^2),$$

and given $b \in \mathbb{F}_q$, we can define $\mathsf{Fold}_{X_2}[f, b] : L_1 \times q(L_2) \to \mathbb{F}_q$ whose low-degree extension is $\widehat{h}_0(X_1, X_2) + b\widehat{h}_1(X_1, X_2)$. A simple calculation shows that partial folding admits a "commutative property", namely for $f\colon L_1 \times L_2 \to \mathbb{F}_q$, and $\boldsymbol{z} = (a, b) \in \mathbb{F}^2$, we have

$$\mathsf{Fold}[f, \boldsymbol{z}] = \mathsf{Fold}_{X_2}[\mathsf{Fold}_{X_1}[f, a], b] = \mathsf{Fold}_{X_1}[\mathsf{Fold}_{X_2}[f, b], a]. \tag{16}$$

Let us now assume that we want to construct an IOPP for m-wise a tensor product of Reed-Solomon code $\mathsf{RS}[\mathbb{F}_q, L_0, k_0]^{\otimes m}$. The idea of the construction is to start by folding with respect to the first variable, until a sufficiently small degree with respect to X_1 is reached. We use $s = \log k_0$ rounds of interaction to reduce the problem of proximity for $\mathsf{RS}[\mathbb{F}_q, L_0, k_0]^{\otimes m}$ to a problem of proximity for

$$\mathsf{RS}[\mathbb{F}_q, L_s, k_s] \otimes \mathsf{RS}[\mathbb{F}_q, L_0, k_0]^{\otimes (m-1)}.$$

Then, we repeat the process with respect to the second variable, using $s = \log k_0$ rounds of interactions to reduce the proximity problem for $\mathsf{RS}[\mathbb{F}_q, L_s, k_s] \otimes \mathsf{RS}[\mathbb{F}_q, L_0, k_0]^{\otimes (m-1)}$ to a proximity problem for

$$\mathsf{RS}[\mathbb{F}_q, L_s, k_s]^{\otimes 2} \otimes \mathsf{RS}[\mathbb{F}_q, L_0, k_0]^{\otimes (m-2)}.$$

By repeating this process for the remaining indeterminates, namely after a total number of rounds $r := m \log k_0$, we are left with a trivial proximity problem for the code

$$\mathsf{RS}[\mathbb{F}_q, L_s, k_s]^{\otimes m}.$$

Compared to Section 7, this approach yields an IOPP for tensor product of Reed-Solomon codes that has the same efficiency parameters than the FRI protocol, which deals with the univariate case. In particular, even when considering that the number of variables m is not constant, we achieve strictly linear prover time and strictly logarithmic verification time.

In order not to overload notations, we present our IOPP construction for a m-wise tensor product of Reed-Solomon codes. One can readily verify that the construction we are going to present can be generalized to the case where the initial proximity testing problem is concerned with a tensor product of distinct Reed-Solomon codes, namely

$$\mathsf{RS}\left[\mathbb{F}_q, L^{(1)}, k^{(1)}\right] \otimes \mathsf{RS}\left[\mathbb{F}_q, L^{(2)}, k^{(2)}\right] \otimes \cdots \otimes \mathsf{RS}\left[\mathbb{F}_q, L^{(m)}, k^{(m)}\right]$$

with different degree bounds $\left(k^{(j)}\right)_{1 \leqslant j \leqslant m}$ and supports $\left(L^{(j)}\right)_{0 \leqslant j \leqslant m}$.

8.1 Sequence of codes with length divided by 2

We assume again that $L_0, \ldots, L_r \subset \mathbb{F}_q$ are defined as per Sections 6.1 or 6.2. Let k be a power of two and set $s := \log k$. As in Section 7.1, we set $L_0 := L$ and $k_0 := k$. For any $i \in [0 \,..\, s-1]$, $L_{i+1} = q_i(L_i)$, where q_i is defined in Section 6, and $k_{i+1} := \frac{k_i}{2}$. Letting $r := ms$, we define a sequence of $r+1$ codes $C_0, C_1, \ldots, C_r$ as follows. The first code C_0 is $C_0 := \mathsf{RS}\left[\mathbb{F}_q, L_0, k_0\right]^{\otimes m}$. For $j \in [1 \,..\, m]$ and $i \in [0 \,..\, s-1]$, the code $C_{(j-1)s+i}$ is

$$C_{(j-1)s+i} := \mathsf{RS}\left[\mathbb{F}_q, L_s, k_s\right]^{\otimes j-1} \otimes \mathsf{RS}\left[\mathbb{F}_q, L_i, k_i\right] \otimes \mathsf{RS}\left[\mathbb{F}_q, L_0, k_0\right]^{\otimes m-j},$$

where we use the convention that, for any $\mathbb{F}_q$-linear space V, $V^{\otimes 0} = \mathbb{F}_q$ and $V^{\otimes 1} = V$ (we have $\mathbb{F}_q \otimes V = V \otimes \mathbb{F}_q = V$).

8.2 Partial folding operators

Let us fix $j \in [1 \,..\, m]$ and $i \in [0 \,..\, s-1]$ for this subsection. We want to define a folding operator with respect to the j-th variable. Once again, we assume $\deg q_i = 2$ to simplify the exposition. One can readily verify that the arguments presented here can be carried over to the case $\deg q_i \geqslant 2$.

For $z \in \mathbb{F}_q$, we construct a folding operator

$$\mathsf{Fold}_j\left[\cdot, z\right] : \mathbb{F}^{L_s^{j-1} \times L_i \times L_0^{m-j}} \times \mathbb{F} \to \mathbb{F}^{L_s^{j-1} \times L_{i+1} \times L_0^{m-j}}$$

that will allows us to reduce the problem of proximity to the code $C_{(j-1)s+i}$ to a problem of half the size.

For $\boldsymbol{X} = (X_1, X_2, \ldots, X_m)$, we denote by $\boldsymbol{X}_{\bar{j}}$ the tuple obtained by removing the j-th entry of $\boldsymbol{X}$, i.e. $\boldsymbol{X}_{\bar{j}} = (X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_m)$. Accordingly, we use the notation $\mathbb{F}_q[\boldsymbol{X}_{\bar{j}}]$ to refer to the polynomial ring $\mathbb{F}_q[X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_m]$. The following corollary is a straightforward generalization of 3.

Corollary 1. *Given a polynomial $\widehat{f} \in \mathbb{F}_q[\boldsymbol{X}_{\bar{j}}][X_j]$ and a monic polynomial $q \in \mathbb{F}_q[X]$ of degree 2, there are two polynomials $\widehat{g}_0, \widehat{g}_1 \in \mathbb{F}_q[\boldsymbol{X}_{\bar{j}}][X_j]$ such that*

$$\widehat{f}(\boldsymbol{X}_{\bar{j}})(X_j) = \widehat{g}_0(\boldsymbol{X}_{\bar{j}})(q_i(X_j)) + X_j \widehat{g}_1(\boldsymbol{X}_{\bar{j}})(q_i(X_j)), \tag{17}$$

and, for all $j' \in [1 \,..\, m]$,

$$\deg_{X_{j'}} \widehat{g}_0, \deg_{X_{j'}} \widehat{g}_1 \leqslant \begin{cases} \frac{\deg_{X_{j'}} \widehat{f}}{2} & \text{if } j' = j \\ \deg_{X_{j'}} \widehat{f} & \text{otherwise.} \end{cases}$$

Definition 10. *Given $q_i(X)$ defined in Section 6, we define $\pi_{j,i} \colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ as the function*

$$\pi_{j,i} \colon (x_1, \ldots, x_m) \mapsto (x_1, \ldots, x_{j-1}, q_i(x_j), x_{j+1}, \ldots, x_m).$$

Definition 11 (Folding with respect to the j-th variable)**.** *Let $f \colon L_s^{j-1} \times L_i \times L_0^{m-j} \to \mathbb{F}_q$ be an arbitrary function and $\widehat{f} \in \mathbb{F}_q[\boldsymbol{X}]$ its low-degree extension. Let g_0, g_1 be the evaluations on $L_s^{j-1} \times L_{i+1} \times L_0^{m-j}$ of the polynomials $\widehat{g_0}, \widehat{g_1} \in \mathbb{F}_q[\boldsymbol{X}]$ given by 1, respectively. For $z \in \mathbb{F}_q$, we define the j-th partial folding of f as the function*

$$\textbf{\textsf{Fold}}_j\left[f, z\right] : L_s^{j-1} \times L_{i+1} \times L_0^{m-j} \to \mathbb{F}_q$$

defined by $\textbf{\textsf{Fold}}_j\left[f, z\right] \coloneqq g_0 + z g_1$.

An immediate consequence of 11 is the following lemma.

Lemma 8 (Completeness)**.** *For any $z \in \mathbb{F}_q$, if $f \in C_{(j-1)s+i}$, then $\textbf{\textsf{Fold}}_j\left[f, z\right] \in C_{(j-1)s+i+1}$.*

Lemma 9 (Local computability)**.** *Let $f \colon L_s^{j-1} \times L_i \times L_0^{m-j} \to \mathbb{F}_q$ be an arbitrary function and let $z \in \mathbb{F}_q$. For any $\boldsymbol{y} \in L_s^{j-1} \times L_{i+1} \times L_0^{m-j}$, the value $\textbf{\textsf{Fold}}_j\left[f, z\right](\boldsymbol{y})$ can be computed with exactly 2 entries of f.*

Proof. Let $\boldsymbol{y} \in L_s^{j-1} \times L_{i+1} \times L_0^{m-j}$. Denote $S_{\boldsymbol{y}} \subset L_s^{j-1} \times L_i \times L_0^{m-j}$ the set

$$S_{\boldsymbol{y}} \coloneqq \pi_{j,i}^{-1}(\{\boldsymbol{y}\}).$$

Since $q_i(X)$ has two distinct roots, the set $S_{\boldsymbol{y}}$ has 2 elements. Let us consider $P_{f,\boldsymbol{y}} \in \mathbb{F}_q[X]$ the polynomial of degree less than 2 such that, for all $\boldsymbol{x} \in S_{\boldsymbol{y}}$,

$$P_{f,\boldsymbol{y}}(x_j) = f(\boldsymbol{x}).$$

We have that the polynomial equation

$$P_{f,\boldsymbol{y}}(X) = \widehat{g}_0(\boldsymbol{y}_{\bar{j}})(y_j) + X\widehat{g}_0(\boldsymbol{y}_{\bar{j}})(y_j)$$

holds, since both polynomials have degree less than 2 and agree on two distinct values. Recalling 11 we have, for all $z \in \mathbb{F}_q$,

$$P_{f,\boldsymbol{y}}(z) = \textbf{\textsf{Fold}}_j\left[f, z\right](\boldsymbol{y}).$$

In particular, the value $\textbf{\textsf{Fold}}_j\left[f, z\right](\boldsymbol{y})$ can be computed by interpolating the set of points

$$\{(x_j, f(\boldsymbol{x})) \mid \boldsymbol{x} \in S_{\boldsymbol{y}}\}$$

of size two.

Proposition 6 (Distance preservation)**.** *Let $j \in [1 \mathinner{..} m]$, $i \in [0 \mathinner{..} s-1]$ and $f \colon L_s^{j-1} \times L_i \times L_0^{m-j} \to \mathbb{F}_q$ be an arbitrary function. Let $\varepsilon \in \left(0, \frac{1}{3}\right)$ and*

$$\delta < 1 - (1 - \Delta(C_{(j-1)s+i+1}) + \varepsilon)^{\frac{1}{3}},$$

where $\Delta(C_{(j-1)s+i+1})$ is the relative distance of the code $\Delta(C_{(j-1)s+i+1})$. Let $\phi_i \colon L_s^{j-1} \times L_i \times L_0^{m-j} \to [0,1]$ and $\phi_{i+1} \colon L_s^{j-1} \times L_{i+1} \times L_0^{m-j} \to [0,1]$ be weight functions such that

$$\forall \boldsymbol{y} \in L_s^{j-1} \times L_{i+1} \times L_0^{m-j},\ \phi_{i+1}(\boldsymbol{y}) \leqslant \frac{1}{2} \sum_{\boldsymbol{x} \in S_{\boldsymbol{y}}} \phi_i(\boldsymbol{x}).$$

If $f \colon L_s^{j-1} \times L_i \times L_0^{m-j} \to \mathbb{F}_q$ has weighted agreement $\mu_{\phi_i}(f, C_{(j-1)s+i}) < 1 - \delta$, then

$$\Pr_{z \in \mathbb{F}_q}\left[\mu_{\phi_{i+1}}(\textbf{\textsf{Fold}}_j\left[f, z\right], C_{(j-1)s+i+1}) > 1 - \delta + \varepsilon\right] < \frac{2}{\varepsilon^2 q}.$$

Proof. We proceed by contraposition and we assume

$$\Pr_{z \in \mathbb{F}_q}\left[\mu_{\phi_{i+1}}(\mathsf{Fold}_j\left[f,z\right], C_{(j-1)s+i+1}) > 1-\delta+\varepsilon\right] \geqslant \frac{2}{\varepsilon^2 q}.$$

Applying 5 on $\mathsf{Fold}_j\left[f,z\right] = g_0 + zg_1$, we get that there exist

$$T \subset L_s^{j-1} \times L_{i+1} \times L_0^{m-j}$$

and $v_0, v_1 \in C_{(j-1)s+i+1}$ satisfying

- $\displaystyle\sum_{\boldsymbol{y} \in T} \phi_{i+1}(\boldsymbol{y}) \geqslant (1-\delta)\left|L_s^{j-1} \times L_{i+1} \times L_0^{m-j}\right|$,
- $g_{0\,|T} = v_{0\,|T}$ and $g_{1\,|T} = v_{1\,|T}$.

Let us consider $\widehat{v_0}, \widehat{v_1} \in \mathbb{F}_q[\boldsymbol{X}]$ the polynomial associated to the codewords $v_0, v_1 \in C_{(j-1)s+i+1}$ respectively. We have:

$$\deg_{X_{j'}} \widehat{v_0}, \deg_{X_{j'}} \widehat{v_1} \begin{cases} < k_s, & \text{for } 1 \leqslant j' < j, \\ < k_i, & \text{for } j' = j, \\ < k_0, & \text{for } j < j' \leqslant m. \end{cases}$$

Define $R \in \mathbb{F}_q[\boldsymbol{X}]$ as the polynomial which, when viewed as a polynomial in $\mathbb{F}_q[\boldsymbol{X}_{\bar{j}}][X_j]$, is equal to

$$R(\boldsymbol{X}_{\bar{j}})(X_j) := \widehat{v_0}(\boldsymbol{X}_{\bar{j}})(q_i(X_j)) + X_j \widehat{v_1}(\boldsymbol{X}_{\bar{j}})(q_i(X_j)).$$

Consider v the evaluation on $L_s^{j-1} \times L_i \times L_0^{m-j}$ of $R(\boldsymbol{X}) \in \mathbb{F}_q[\boldsymbol{X}]$. Since $k_{i+1} = k_i/2$, we have

$$\deg_{X_j} R \leqslant 1 + 2 \cdot (k_{i+1} - 1) < k_i.$$

Given the degrees in the remaining variables of $\widehat{v_0}(\boldsymbol{X}), \widehat{v_1}(\boldsymbol{X})$, we deduce that $v \in C_{(j-1)s+i}$. We have that v agrees with f on the preimages under the map $\pi_{j,i}$ of the elements of T. Indeed, for all $\boldsymbol{y} \in T$ and $\boldsymbol{x} \in \pi_{j,i}^{-1}(\{\boldsymbol{y}\})$, we have

$$\begin{aligned} f(\boldsymbol{x}) &= \widehat{g_0}(\pi_{j,i}(\boldsymbol{x})) + x_j \widehat{g_1}(\pi_{j,i}(\boldsymbol{x})) \\ &= g_0(\pi_{j,i}(\boldsymbol{x})) + x_j g_1(\pi_{j,i}(\boldsymbol{x})) \\ &= v_0(\pi_{j,i}(\boldsymbol{x})) + x_j v_1(\pi_{j,i}(\boldsymbol{x})) \\ &= v(\boldsymbol{x}). \end{aligned}$$

Thus v agrees with f on $S_T := \bigsqcup_{\boldsymbol{y} \in T} S_{\boldsymbol{y}}$. Since $v \in C_{(j-1)s+i}$, we have

$$\begin{aligned} \mu_{\phi_i}(f, C_{(j-1)s+i}) &\geqslant \frac{1}{\left|L_s^{j-1} \times L_i \times L_0^{m-j}\right|} \sum_{\boldsymbol{x} \in S_T} \phi_i(\boldsymbol{x}) \\ &= \frac{1}{\left|L_s^{j-1} \times L_i \times L_0^{m-j}\right|} \sum_{\boldsymbol{y} \in T} \sum_{\boldsymbol{x} \in S_{\boldsymbol{y}}} \phi_i(\boldsymbol{x}) \\ &\geqslant \frac{1}{\left|L_s^{j-1} \times L_{i+1} \times L_0^{m-j}\right|} \sum_{\boldsymbol{y} \in T} \phi_{i+1}(\boldsymbol{y}). \end{aligned}$$

Eventually, we conclude that $\mu_{\phi_i}(f, C_{(j-1)s+i}) \geqslant 1-\delta$ by definition of T. This contradicts the hypothesis on f.

8.3 Improved IOPP for tensor product of RS codes

Given a sequence of codes $(C_i)_{0 \leqslant i \leqslant r}$ as defined in Section 8.1 and a family of folding operators for each code C_i as in Section 8.2, Section 3 leads to a public-coin IOPP $(\mathcal{P}_{\mathsf{RS}^m}, \mathcal{V}_{\mathsf{RS}^m})$ for the code C_0. With our choices of parameters, the last function f_r is again supposed to be constant. Thus, we use the variant of the protocol described in Remark 2. For the last round of the COMMIT phase, the prover $\mathcal{P}_{\mathsf{RS}^m}$ sends a single field element $\beta \in \mathbb{F}_q$ and the verifier checks that

$$\beta = \mathbf{Fold}_j \left[f_{r-1}, z_{r-1}\right](\boldsymbol{y}_r).$$

This leads to the following theorem.

Theorem 4. *Let k, m be positive integers such that $k > 1$ is a power of two. Let $L \subset \mathbb{F}_q^\times$ be a set of size $|L| > k$ defined as per Section 6. The construction presented in Section 3 with the folding operators defined as per Defintion 11 leads to public-coin IOPP system $(\mathcal{P}_{\mathsf{RS}^m}, \mathcal{V}_{\mathsf{RS}^m})$ for the tensor product code $(\mathsf{RS}\left[\mathbb{F}_q, L, k\right])^{\otimes m}$ of blocklength n^m satisfying:*

1. **Round complexity** *is $r(n^m) = m \log k$.*
2. **Query complexity** *is $q(n^m) = 2\alpha m \log k + 1$ for s repetitions of the QUERY phase.*
3. **Proof length** *is $l(n^m) < n^m$.*
4. **Prover complexity** *is $t_p(n^m) < 8n^m$.*
5. **Verifier decision complexity** *is $t_v(n^m) \leqslant 8\alpha m \log k$.*
6. ***Completeness:*** *If $f \in (\mathsf{RS}\left[\mathbb{F}_q, L, k\right])^{\otimes m}$ and if the oracles $f_1, \dots f_r$ are computed by an honest prover $\mathcal{P}_{\mathsf{RS}^m}$, then $\mathcal{V}_{\mathsf{RS}^m}$ outputs **accept** with probability 1.*
7. ***Soundness:*** *Assume that $f \colon L^m \to \mathbb{F}_q$ is δ-far from $(\mathsf{RS}\left[\mathbb{F}_q, L, k\right])^{\otimes m}$. Define $\lambda := \left(1 - \frac{k}{|L|}\right)^m$ and, for any $\varepsilon \in \left(0, \frac{1}{3}\right)$, set $\gamma(\lambda, \varepsilon) := 1 - (1 - \lambda + \varepsilon)^{1/3}$. Then, for any unbounded prover $\mathcal{P}^*$, the verifier $\mathcal{V}_{\mathsf{RS}^m}$ outputs accept after α repetitions of the QUERY phase with probability at most*
$$\frac{2m \log k}{\varepsilon^2 q} + (1 - \min(\delta, \gamma(\varepsilon, \lambda)) + \varepsilon m \log k)^\alpha.$$

Proof. Completeness and soundness follows from Theorem 2 by observing that the relative distances of the codes $C_0, \dots, C_r$ are greater than $1 - \frac{k}{|L|}$.

By construction, the IOPP has $r = \log(k^m)$ rounds. During the QUERY phase, the verifier makes 2 queries to each function $f_0, \dots, f_{r-1}$. The verifier also queries the element β sent during the last round of the COMMIT phase. This gives the claimed query cmoplexity.

Let us denote by n_i the length of the code C_i for $i \in [0 \mathinner{..} r]$. The proof length is sum of the n_i's for $i \in [1 \mathinner{..} r]$. Since $n_{i+1} = n_i/2$, the sum of the first terms of a geometric sequence gives the claimed proof length.

For fixed $j \in [1 \mathinner{..} m], i \in [0 \mathinner{..} s-1]$ and $\boldsymbol{y} \in L_s^{j-1} \times q_i(L_i) \times L_0^{m-j}$, we compute the number c of field operations to perform in order to get the value $\mathbf{Fold}_j\left[f, z\right](\boldsymbol{y})$. Recall that $\mathbf{Fold}_j\left[f, z\right](\boldsymbol{y})$ can be computed by interpolating the set of points $\left\{(x_j, f(\boldsymbol{y}_{\bar{j}})(x_j)) \mid q_i(x_j) = y_j\right\}$ and evaluating the obtained polynomial at z.

We consider x_j, x'_j the two distinct roots of the polynomial $q_i(X) - y_j$. For

$$\boldsymbol{x} := (y_1, \dots, y_{j-1}, x_j, y_{j+1}, \dots, y_m) \text{ and } \boldsymbol{x}' := (y_1, \dots, y_{j-1}, x'_j, y_{j+1}, \dots, y_m),$$

we have

$$\mathsf{Fold}_j\left[f, z\right](\boldsymbol{y}) = P_{f,\boldsymbol{y}}(z) = \frac{1}{x_j - x'_j}\left(f(\boldsymbol{x})(z - x_j) - f(\boldsymbol{x}')(z - x'_j)\right).$$

In this case, computing $\mathsf{Fold}_j\left[f, z\right](\boldsymbol{y})$ takes at most 8 field operations.

At each round, the prover performs $8n_i$ computations. Summing over r rounds, we get the claimed prover complexity. During one round of the QUERY phase, the verifier evaluates $\mathsf{Fold}_j\left[f, z\right]$ at a single point, therefore the verifier complexity for a repetition parameter s is $8\alpha r \leqslant 8\alpha m \log k$.

Remark 6. *In the case where the polynomial q_i has degree larger than 2, one can compute the cost of evaluating the folding of a function at a single point by looking at the number of operations needed to interpolate and evaluate at a single point a polynomial of degree less than $\deg q_i$ (e.g. using Lagrange interpolation formula).*

Comparisons with the univariate case Soundness of the FRI protocol [BBHR18] has been analyzed in [BBHR18, BKS18, BGKS20, BCI+20]. For a Reed-Solomon code of blocklength N, relative distance λ and alphabet $\mathbb{F}_q$ of size *linear* in N, the soundness is given by [BGKS20]. Specifically, the FRI protocol is a r-round IOPP, $\mathsf{r} < \log N$, with soundness error (for a single repetition of the QUERY phase) bounded from above by

$$\frac{2\mathsf{r}}{\varepsilon^2\left|\mathbb{F}_q\right|} + (1 - \min(\delta, \gamma(\varepsilon, \lambda)) + \varepsilon\mathsf{r}),$$

where $\gamma(\varepsilon, \lambda) = 1 - (1 - \lambda + \varepsilon)^{1/3}$. Authors of [BGKS20] also showed that the bound $\gamma(\varepsilon, \lambda)$ is tight for RS codes evaluated over the entire field, and when this field has characteristic two. Subsequently, [BCI+20] improved soundness of the FRI protocol for quadratic-size fields using symbolic list-decoding algorithms for RS codes.

We point out that the soundness error of our IOPP for tensor product of RS code is given by the exact same formula than the one shown in [BGKS20] for the univariate case, albeit tensor codes have worse relative distance.

In Figure 3, we present the parameters of the FRI protocol for Reed-Solomon codes [BBHR18] and our IOPPs for tensor product of RS codes side by side, for a single repetition of the QUERY phase. We consider codes of blocklength N and dimension K and a single repetition of the QUERY phase. In order to achieve arbitrary constant soundness error, both protocols require to repeat the QUERY phase. This process increases query complexity and verifier running time by a multiplicative factor independent of N. However, the FRI protocol has better soundness, thus requires less repetitions.

Scheme	Prover	Verifier	Query	Length	Rounds
RS IOPP [BBHR18]	$< 8N$	$< 8\log K$	$2\log K + 1$	$< N$	$\log K$
$\mathrm{RS}^{\otimes m}$ IOPP (Thm. 3)	$< (2m+4)N$	$< 4\left(\frac{2^m}{m} + 1\right)\log K$	$\frac{2^m}{m}\log K$	$< \frac{N}{2^m - 1}$	$\frac{\log K}{m}$
$\mathrm{RS}^{\otimes m}$ IOPP (Thm. 4)	$< 8N$	$< 8\log K$	$2\log K + 1$	$< N$	$m\log K$

Figure 3: Comparison between the IOPP for a RS code of [BBHR18] and our IOPPs for a tensor product of RS code. We compare codes with the same blocklength N and same dimension K.

In order to simplify the exposition, the IOPPs underlying Theorem 3 and Theorem 4 are presented for the case where every polynomial $q_i(X)$ defined in Section 6 has degree 2. As done in [BBHR18], the degree of the polynomials $q_i(X)$ can be considered as a parameter of the protocol.

In [BBHR18], such a parameter is named “localization parameter”, and we will refer to it as such in the following comments about Figure 3.

The constants reported for the FRI protocol [BBHR18] are given for a localization parameter set equal to 2. Thus, the constants in the first line of Figure 3 differ from the constants that can be found in [BBHR18, Theorem 1.3], where the localization parameter is set to 4.

The IOP of Proximity from Theorem 4 outperforms the one of Theorem 3 for all parameters, except proof length and round complexity. By choosing a localization parameter $l > 2$ (*i.e.* by setting $\deg q_i(X) = l$ for all i in Section 6), the proof length in Theorem 4 could be divided by $l-1$. This would incurs a modest increase of the constants displayed for prover and verifier complexities, while dividing by l the total number of rounds.

9 Short Reed-Muller codes

9.1 Sequence of codes

Similarly to Section 7.1, we will consider two families of short Reed-Muller codes, depending on whether case 6.1 or case 6.2 holds. Let k be a power of two, $k < |L|$ and set $r = \log_2 k$. We consider $L_0 = L, L_1, \dots, L_r$ as constructed in Section 6.

Set $k_0 := k$. For $0 < i \leqslant r$, define $k_{i+1} := \frac{k_i}{2}$. In particular, for all i, we have $k_i < |L_i|$. Let us denote by SRM_i the short Reed-Muller code $\mathsf{SRM}\left[\mathbb{F}_q, L_i, m, k_i\right]$.

Starting from the code $\mathsf{SRM}_0 = \mathsf{SRM}\left[\mathbb{F}_q, L, m, k\right]$, this defines a sequence of Reed-Muller codes $(\mathsf{SRM}_i)_{0\leqslant i\leqslant r}$. For each i, the relative distance λ_i of SRM_i is at least $1 - \frac{k_i - 1}{|L_i|}$, hence $\min_i \lambda_i \geqslant 1 - \frac{k}{|L|}$.

9.2 Folding operators

Let $(\mathsf{SRM}_i)_{0\leqslant i\leqslant r}$ be a sequence of short Reed-Muller codes defined as described in Section 9.1 (regardless we are in case 6.1 or 6.2). For each $i \in [0 \,..\, r-1]$, we define a family of folding operators which will enables us to iteratively reduce the problem of proximity testing to a code SRM_i to a problem of size 2^m times smaller, namely proximity testing to SRM_{i+1}.

Note that the sequences of evaluation domains $(L_i^m)_i$ and degree bounds $(k_i)_i$ are defined exactly the same way as in the tensor product case. However, if we design folding operators for Reed-Muller codes by following the same construction than in Definition 9, then the distance preservation property does not hold anymore. For this reason, some *balancing functions* are involved in the definition of folding operators for Reed-Muller codes.

Definition 12 (Balancing functions)**.** *Let $i \in [0 \,..\, r-1]$. For any $\boldsymbol{e} \in \{0,1\}^m$, we call a* balancing function *any map $h_{\boldsymbol{e}} \colon L_{i+1}^m \to \mathbb{F}_q$ which corresponds to the evaluation of a m-variate multilinear monic monomial $\widehat{h}_{\boldsymbol{e}}$ of total degree exactly $\left\lfloor \frac{w_H(\boldsymbol{e})}{2} \right\rfloor$. We call $(h_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m}$ a* balancing tuple *for the* code SRM_{i+1}.

Definition 13 (Folding operator)**.** *Let $i \in [0, r-1]$. Let $(h_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m}$ be a balancing tuple for SRM_{i+1} and let $f \colon L_i^m \to \mathbb{F}_q$ be an arbitrary function. Given $(\widehat{g}_{\boldsymbol{e}})_{\boldsymbol{e}\in\{0,1\}^m}$ the 2^m m-variate polynomials of the decomposition of Proposition 2, denote $g_{\boldsymbol{e}}$ the evaluation on L_{i+1}^m of $\widehat{g}_{\boldsymbol{e}}$. For any $(\boldsymbol{p}, \boldsymbol{p}') \in \left(\mathbb{F}_q^m\right)^2$, we define the* folding of f *as the function $\textbf{\textsf{Fold}}\left[f, (\boldsymbol{p}, \boldsymbol{p}')\right] : L_{i+1}^m \to \mathbb{F}_q$ such that*

$$\textbf{\textsf{Fold}}\left[f, (\boldsymbol{p}, \boldsymbol{p}')\right](\boldsymbol{y}) = \sum_{\boldsymbol{e}\in\{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} g_{\boldsymbol{e}}(\boldsymbol{y}) + \sum_{\substack{\boldsymbol{e}\in\{0,1\}^m \\ \boldsymbol{e}\neq\boldsymbol{0}}} \boldsymbol{p}'^{\boldsymbol{e}} h_{\boldsymbol{e}}(\boldsymbol{y}) g_{\boldsymbol{e}}(\boldsymbol{y}). \tag{18}$$

Lemmas 10 and 11 show that this defines a folding operator for SRM_i as per Definition 5.

Lemma 10 (Completeness)**.** *Let $(\boldsymbol{p}, \boldsymbol{p}') \in \left(\mathbb{F}_q^m\right)^2$, and $f\colon L_i^m \to \mathbb{F}_q \in \mathsf{SRM}_i$, then* $\textbf{\textsf{Fold}}[f, (\boldsymbol{p}, \boldsymbol{p}')] : L_{i+1}^m \to \mathbb{F}_q$ *belongs to* SRM_{i+1}.

Proof. Proof relies on Proposition 2. If $f \in \mathsf{SRM}_i$, then the polynomial $\widehat{f}(\boldsymbol{X})$ associated to f has total degree at most $k_i - 1$. Therefore, for any $\boldsymbol{e} \in \{0,1\}^m$, $\deg \widehat{g}_{\boldsymbol{e}} \leqslant \left\lfloor \frac{k_i - 1 - w_H(\boldsymbol{e})}{2} \right\rfloor$. Since k_i is even, we have both $\deg \widehat{g}_{\boldsymbol{e}} < k_{i+1}$ and $\deg \left(\widehat{h}_{\boldsymbol{e}} \widehat{g}_{\boldsymbol{e}}\right) \leqslant \left\lfloor \frac{w_H(\boldsymbol{e})}{2} \right\rfloor + \left\lfloor \frac{k_i - 1 - w_H(\boldsymbol{e})}{2} \right\rfloor < k_{i+1}$. This means $\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p}')\right] : L_{i+1}^m \to \mathbb{F}_q$ corresponds to the evaluation of a polynomial in $\mathbb{F}_q[\boldsymbol{X}]$ of total degree less than k_{i+1}. $\square$

Lemma 11 (Locality)**.** *Let $f\colon L_i^m \to \mathbb{F}_q$ be an arbitrary function and let $(\boldsymbol{p}, \boldsymbol{p}') \in \left(\mathbb{F}_q^m\right)^2$. Given $\boldsymbol{y} \in L_{i+1}^m$, the value* $\textbf{\textsf{Fold}}[f, (\boldsymbol{p}, \boldsymbol{p}')](\boldsymbol{y})$ *can be computed with exactly 2^m queries to f.*

Proof. The proof follows from the one of Lemma 7. For any $\boldsymbol{y} \in L_{i+1}^m$, the vector $(g_{\boldsymbol{e}}(\boldsymbol{y}))_{\boldsymbol{e} \in \{0,1\}^m}$ corresponds to the vector of coefficients of the low-degree extension of the function $f_{|S_{\boldsymbol{y}}}$. $\square$

Let us now show that the folding operator of Definition 13 satisfies distance preservation (Definition 7).

Proposition 7 (Distance preservation)**.** *Denote λ_{i+1} the minimum relative distance of SRM_{i+1}. Let $f\colon L_i^m \to \mathbb{F}_q$ be an arbitrary function. Let $\varepsilon \in \left(0, \frac{2}{3}\right)$ and*

$$\delta < \min\left(1 - (1 - \lambda_{i+1} + \varepsilon)^{\frac{1}{3}}, \frac{1}{2}(\lambda_{i+1} + m\frac{\varepsilon}{2})\right).$$

Let $\phi_i\colon L_i^m \to [0,1]$ and $\phi_{i+1}\colon L_{i+1}^m \to [0,1]$ be weight functions such that

$$\forall \boldsymbol{y} \in L_{i+1}^m,\ \phi_{i+1}(\boldsymbol{y}) \leqslant \frac{1}{2^m} \sum_{\boldsymbol{x} \in S_{\boldsymbol{y}}} \phi_i(\boldsymbol{x}),$$

if $f\colon L_i^m \to \mathbb{F}_q$ has weighted agreement $\mu_{\phi_i}(f, \mathsf{SRM}_i) > 1 - \delta$, then

$$\Pr_{\boldsymbol{p}, \boldsymbol{p}' \in \mathbb{F}_q^m}\left[\mu_{\phi_{i+1}}\left(\textbf{\textsf{Fold}}[f, (\boldsymbol{p}, \boldsymbol{p}')], \mathsf{SRM}_{i+1}\right) > 1 - \delta + m\varepsilon\right] < \frac{16m}{\varepsilon^2 q}.$$

Proof. Let $f\colon L_i^m \to \mathbb{F}_q$ be such that $\mu_{\phi_i}(f, \mathsf{SRM}_i) > 1 - \delta$, and $(\widehat{g}_{\boldsymbol{e}})_{\boldsymbol{e} \in \{0,1\}^m}$ the 2^m m-variate polynomials appearing in the decomposition of $\widehat{f}$ in Proposition 2. For any $\boldsymbol{p} \in \mathbb{F}_q^m$, denote $u_{\boldsymbol{p}}$ the function $u_{\boldsymbol{p}} = \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} g_{\boldsymbol{e}}$, and for any $\boldsymbol{e} \in \{0,1\}^m \setminus \{\boldsymbol{0}\}$, define $u_{\boldsymbol{e}} = h_{\boldsymbol{e}} g_{\boldsymbol{e}}$. One can rewrite $\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p}')\right]$ as follows:

$$\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p}')\right] = u_{\boldsymbol{p}} + \sum_{\substack{\boldsymbol{e} \in \{0,1\}^m \\ \boldsymbol{e} \neq \boldsymbol{0}}} \boldsymbol{p}'^{\boldsymbol{e}} u_{\boldsymbol{e}}.$$

We proceed by contraposition, assuming that

$$\Pr_{\boldsymbol{p}, \boldsymbol{p}' \in \mathbb{F}_q^m}\left[\mu_{\phi_{i+1}}\left(\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p}')\right], \mathsf{SRM}_{i+1}\right) > 1 - \delta + m\varepsilon\right] \geqslant \frac{16m}{\varepsilon^2 q},$$

or, in other words,

$$\Pr_{\boldsymbol{p} \in \mathbb{F}_q^m}\left[\Pr_{\boldsymbol{p}' \in \mathbb{F}_q^m}\left[\mu_{\phi_{i+1}}\left(\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p}')\right], \mathsf{SRM}_{i+1}\right) > 1 - \delta + m\varepsilon\right] \geqslant \frac{8m}{\varepsilon^2 q}\right] \geqslant \frac{8m}{\varepsilon^2 q}.$$

Let

$$A := \left\{ \boldsymbol{p} \in \mathbb{F}_q^m \mid \Pr_{\boldsymbol{p}' \in \mathbb{F}_q^m} \left[\mu_{\phi_{i+1}} \left(\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p}') \right], \mathsf{SRM}_{i+1} \right) > 1 - \delta + m\varepsilon \right] \geqslant \frac{8m}{\varepsilon^2 q} \right\}.$$

Proposition 4 implies that, for any $\boldsymbol{p} \in A$, there exist $T_{\boldsymbol{p}} \subset L_{i+1}^m$ and $(w_{\boldsymbol{p},\boldsymbol{e}})_{\boldsymbol{e} \in \{0,1\}^m}$ with $w_{\boldsymbol{p},\boldsymbol{e}} \in \mathsf{SRM}_{i+1}$ such that

- $\sum_{\boldsymbol{y} \in T_{\boldsymbol{p}}} \phi_{i+1}(\boldsymbol{y}) \geqslant (1 - \delta + m\frac{\varepsilon}{2}) \left| L_{i+1}^m \right|$,
- $w_{\boldsymbol{p},\boldsymbol{0}\,|T_{\boldsymbol{p}}} = u_{\boldsymbol{p}\,|T_{\boldsymbol{p}}}$,
- for each $\boldsymbol{e} \in \{0,1\}^m \setminus \{\boldsymbol{0}\}$, $w_{\boldsymbol{p},\boldsymbol{e}\,|T_{\boldsymbol{p}}} = u_{\boldsymbol{e}\,|T_{\boldsymbol{p}}}$.

Thus, for all $\boldsymbol{p} \in A$,

$$\mu_{\phi_{i+1}} \left(\sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} g_{\boldsymbol{e}}, \mathsf{SRM}_{i+1} \right) \geqslant \frac{1}{\left| L_{i+1}^m \right|} \sum_{\boldsymbol{y} \in T_{\boldsymbol{p}}} \phi_{i+1}(\boldsymbol{y}) \geqslant 1 - \delta + m\frac{\varepsilon}{2}.$$

Since $|A| > \frac{2m}{\varepsilon^2} q^{m-1}$, we have

$$\Pr_{\boldsymbol{p} \in \mathbb{F}_q^m} \left[\mu_{\phi_{i+1}} \left(\sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{p}^{\boldsymbol{e}} g_{\boldsymbol{e}}, \mathsf{SRM}_{i+1} \right) > 1 - \delta + m\frac{\varepsilon}{2} \right] \geqslant \frac{8m}{\varepsilon^2 q}.$$

Again, by Proposition 4, we obtain $T \subset L_{i+1}^m$ and $(v_{\boldsymbol{e}})_{\boldsymbol{e} \in \{0,1\}^m}$ with $v_{\boldsymbol{e}} \in \mathsf{SRM}_{i+1}$ such that

- $\sum_{\boldsymbol{y} \in T} \phi_{i+1}(\boldsymbol{y}) \geqslant (1 - \delta) \left| L_{i+1}^m \right|$,
- for each $\boldsymbol{e} \in \{0,1\}^m$, $v_{\boldsymbol{e}\,|T} = g_{\boldsymbol{e}\,|T}$.

Fix $\boldsymbol{p} \in A$. For any $\boldsymbol{e} \in \{0,1\}^m$, $\boldsymbol{e} \neq \boldsymbol{0}$, we have

$$w_{\boldsymbol{p},\boldsymbol{e}\,|T_{\boldsymbol{p}} \cap T} = u_{\boldsymbol{e}\,|T_{\boldsymbol{p}} \cap T} = (h_{\boldsymbol{e}} g_{\boldsymbol{e}})_{|T_{\boldsymbol{p}} \cap T} = (h_{\boldsymbol{e}} v_{\boldsymbol{e}})_{|T_{\boldsymbol{p}} \cap T}.$$

Besides, the intersection of $T_{\boldsymbol{p}}$ and T satisfies

$$\begin{aligned} |T_{\boldsymbol{p}} \cap T| &= |T_{\boldsymbol{p}}| + |T| - |T_{\boldsymbol{p}} \cup T| \\ &\geqslant \sum_{\boldsymbol{y} \in T_{\boldsymbol{p}}} \phi_{i+1}(\boldsymbol{y}) + \sum_{\boldsymbol{y} \in T} \phi_{i+1}(\boldsymbol{y}) - \left| L_{i+1}^m \right| \\ &\geqslant \left(1 - 2\delta + m\frac{\varepsilon}{2} \right) \left| L_{i+1}^m \right|, \\ &\geqslant (1 - \lambda_{i+1}) \left| L_{i+1}^m \right|. \end{aligned}$$

Since λ_{i+1} is the minimum relative distance of SRM_{i+1}, we deduce that $w_{\boldsymbol{p},\boldsymbol{e}} = h_{\boldsymbol{e}} v_{\boldsymbol{e}}$ for every $\boldsymbol{e} \in \{0,1\}^m \setminus \{\boldsymbol{0}\}$.

For any $\boldsymbol{e} \in \{0,1\}^m$, consider polynomials $\widehat{v}_{\boldsymbol{e}}, \widehat{w}_{\boldsymbol{e},\boldsymbol{p}} \in \mathbb{F}_q[\boldsymbol{X}]$ of total degrees at most k_{i+1}, such that for all $\boldsymbol{x} \in L_{i+1}^m$, $\widehat{v}_{\boldsymbol{e}}(\boldsymbol{x}) = v_{\boldsymbol{e}}(\boldsymbol{x})$ and $\widehat{w}_{\boldsymbol{e},\boldsymbol{p}}(\boldsymbol{x}) = w_{\boldsymbol{e},\boldsymbol{p}}(\boldsymbol{x})$. Hence, for all $\boldsymbol{x} \in L_{i+1}^m$, $\widehat{w}_{\boldsymbol{e},\boldsymbol{p}}(\boldsymbol{x}) = \widehat{v}_{\boldsymbol{e}}(\boldsymbol{x}) \widehat{h}_{\boldsymbol{e}}(\boldsymbol{x})$, which means that

$$\widehat{w}_{\boldsymbol{e},\boldsymbol{p}} - \widehat{v}_{\boldsymbol{e}} \widehat{h}_{\boldsymbol{e}} = 0 \bmod \left(Z_{i+1}(X_1), \ldots, Z_{i+1}(X_m) \right), \tag{19}$$

where $Z_{i+1}(X) = \prod_{a \in L_{i+1}}(X - a)$ has degree $|L_{i+1}|$. Since $k_{i+1} < |L_{i+1}|$, we have that for any j, $\deg_{X_j} \widehat{v}_{\boldsymbol{e}} \leqslant |L_{i+1}| - 2$. Moreover, $\deg_{X_i} \widehat{h}_{\boldsymbol{e}} \leqslant 1$, thus the above equality is true without the modulo:

$$\widehat{w}_{\boldsymbol{e},\boldsymbol{p}} - \widehat{v}_{\boldsymbol{e}}\widehat{h}_{\boldsymbol{e}} = 0. \tag{20}$$

Therefore, $\deg \widehat{v}_{\boldsymbol{e}} < k_{i+1} - \left\lfloor \frac{w_H(\boldsymbol{e})}{2} \right\rfloor$. For all $\boldsymbol{e} \in \{0,1\}^m$, we have

$$\deg \boldsymbol{X}^{\boldsymbol{e}} \widehat{v}_{\boldsymbol{e}}(q_i(X_1) \dots, q_i(X_m)) \leqslant w_H(\boldsymbol{e}) + 2\left(k_{i+1} - 1 - \frac{w_H(\boldsymbol{e})}{2}\right) < k_i,$$

hence the polynomial $R \in \mathbb{F}_q[\boldsymbol{X}]$ defined by

$$R(\boldsymbol{X}) := \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{X}^{\boldsymbol{e}} \widehat{v}_{\boldsymbol{e}}(q_i(X_1) \dots, q_i(X_m))$$

has total degree $\deg R < k_i$. Thus the evaluation of R on L_i^m is a codeword $v \in \mathsf{SRM}_i$. For any $\boldsymbol{y} \in T$ and $\boldsymbol{x} \in S_{\boldsymbol{y}}$, we have

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{x}^{\boldsymbol{e}} g_{\boldsymbol{e}}(\boldsymbol{y}) = \sum_{\boldsymbol{e} \in \{0,1\}^m} \boldsymbol{x}^{\boldsymbol{e}} v_{\boldsymbol{e}}(\boldsymbol{y}) = v(\boldsymbol{x}).$$

Hence, v agrees with the function f on the set $S_T := \bigsqcup_{\boldsymbol{y} \in T} S_{\boldsymbol{y}}$. Since $v \in \mathsf{SRM}_i$, we have

$$\mu_{\phi_i}(f, \mathsf{SRM}_i) \geqslant \frac{1}{|L_i^m|} \sum_{\boldsymbol{x} \in S_T} \phi_i(\boldsymbol{x}) = \frac{1}{|L_i^m|} \sum_{\boldsymbol{y} \in T} \sum_{\boldsymbol{x} \in S_{\boldsymbol{y}}} \phi_i(\boldsymbol{x}) \geqslant \frac{1}{|L_{i+1}^m|} \sum_{\boldsymbol{y} \in T} \phi_{i+1}(\boldsymbol{y}).$$

Eventually, we conclude that $\mu_{\phi_i}(f, \mathsf{SRM}_i) \geqslant 1 - \delta$ by definition of T. □

Remark 7. *Note that for Reed-Muller codes whose degree bound is larger than $|L|$, we would not be able to deduce* (20) *from* (19)*. Nonetheless, for applications to proof systems, the evaluation map is typically injective,* i.e. $k \leqslant |L|$.

9.3 IOPP for short Reed-Muller codes

Given a sequence of codes $(\mathsf{SRM}_i)_{0 \leqslant i \leqslant r}$ as defined in Section 9.1 and a family of folding operators for each code SRM_i (see Section 9.2), the generic construction described proposed in Section 3.2 leads to a public-coin IOPP $(\mathcal{P}_{\mathsf{RM}}, \mathcal{V}_{\mathsf{RM}})$ for the code SRM_0. As in Section 7, the last function f_r is supposed to be constant. Therefore, we use the variant of the protocol described in Remark 2. Specifically, instead of sending f_r during the COMMIT phase, the prover $\mathcal{P}_{\mathsf{RM}}$ sends a single field element $\beta \in \mathbb{F}_q$. The verifier $\mathcal{V}_{\mathsf{RM}}$ does not run a membership test to C_r but checks the equation $\beta = \mathbf{Fold}\left[f_{r-1}, \boldsymbol{p}_{r-1}\right](\boldsymbol{y}_r)$. The properties of the resulting IOPP system $(\mathcal{P}_{\mathsf{RM}}, \mathcal{P}_{\mathsf{RM}})$ are displayed in the following theorem.

Theorem 5. *Let k, m be positive integers. Assume k is a power of two. Let $L \subset \mathbb{F}_q^\times$ as described in Section 6 such that $k < |L|$. There exists a public-coin IOPP system $(\mathcal{P}_{\mathsf{RM}}, \mathcal{V}_{\mathsf{RM}})$ testing proximity of a function $f \colon L^m \to \mathbb{F}_q$ to the short Reed-Muller code $\mathsf{SRM}\left[\mathbb{F}_q, L, m, k\right]$ with the following properties:*

1. **Round complexity** *is $r(n^m) < \log n$.*
2. **Query complexity** *is $q(n^m) < \alpha(2^m \log n + 1)$ for a QUERY phase with repetition parameter α.*

3. **Proof length** *is* $l(n^m) < \frac{n^m}{(2^m - 1)}$.

4. **Prover complexity** *is* $t_p(n^m) < \left(\frac{11}{2}m + 14\right) n^m$.

5. **Verifier decision complexity** *is* $t_v(n^m) < \alpha 2^m \left(\frac{11}{4}m + 7\right) \log k$.

6. **Perfect completeness:** *If* $f \in \mathsf{SRM}\left[\mathbb{F}_q, L, m, k\right]$ *and if the oracles* $f_1, \ldots f_r$ *are computed by an honest prover, then* $\mathcal{V}_{\mathsf{RM}}$ *outputs* ***accept*** *with probability 1.*

7. **Soundness:** *Assume that* $f \colon L^m \to \mathbb{F}_q$ *is* δ*-far from* $\mathsf{SRM}\left[\mathbb{F}_q, L, m, k\right]$*. Denote* $\lambda = 1 - \frac{k}{|L|}$*. For any* $\varepsilon \in (0, \frac{2}{3})$*, set* $\gamma(\varepsilon, \lambda) := \min\left(1 - (1 - \lambda + \varepsilon)^{1/3}, \frac{1}{2}(\lambda + m\frac{\varepsilon}{2})\right)$*. Then, for any unbounded prover* $\mathcal{P}^*$*, the verifier* $\mathcal{V}$ *outputs* accept *after* α *repetitions of the* QUERY *phase with probability at most*

$$r\frac{16m}{\varepsilon^2 q} + (1 - \min(\delta, \gamma(\varepsilon, \lambda)) + rm\varepsilon)^{\alpha}.$$

Proof. We apply the construction of the public-coin IOPP system presented in Section 3.2 with the family of folding operators define in Section 9.2. Completeness and soundness follow from Theorem 2.The number of round is $r = \log k < \log |L|$. Query complexity and proof length are the same than in Theorem 3. For soundness, recall that $\min_i \lambda_i \geqslant 1 - \frac{k}{|L|}$ where λ_i is the relative distance of SRM_i.

Let $f \colon L_i^m \to \mathbb{F}_q$ be an arbitrary function and let $(\boldsymbol{p}, \boldsymbol{p'}) \in \left(\mathbb{F}_q^m\right)^2$. We analyze prover complexity by first computing the cost of evaluating $\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p'})\right]$ on L_{i+1}^m. The prover $\mathcal{P}_{\mathsf{RM}}$ can compute the vectors $(\boldsymbol{p}^{\boldsymbol{e}})_{\boldsymbol{e} \in \{0,1\}^m}$ and $\left(\boldsymbol{p'}^{\boldsymbol{e}}\right)_{\boldsymbol{e} \in \{0,1\}^m}$ in less than $2 \cdot 2^m$ multiplications. Given $\boldsymbol{y} \in L_{i+1}^m$, we look at the cost of computing $\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p'})\right](\boldsymbol{y})$ (see Equation (18)). Recalling Definition 12, computing the values $\widehat{h}_{\boldsymbol{e}}(\boldsymbol{y})$ for all $\boldsymbol{e} \in \{0,1\}^m$ takes at most $m2^{m-2}$ operations.

As shown in proof of Lemma 7, the vector $(g_{\boldsymbol{e}}(\boldsymbol{y}))$ corresponds to the coefficients of the multilinear low-degree extension of $f_{|S_{\boldsymbol{y}}}$. By Lemma 1, this interpolation can be performed with $5m2^{m-1}$ arithmetic operations. Prover then computes the first sum of Equation (18) using 2^m multiplications and $2^m - 1$ additions. Similarly, the second sum can be computed in less than $3 \cdot 2^m$ arithmetic operations.

Overall, for any function $f : L_i^m \to \mathbb{F}_q$ and $\boldsymbol{p}, \boldsymbol{p'} \in \mathbb{F}_q^m$, the prover can evaluate $\mathsf{Fold}\left[f, (\boldsymbol{p}, \boldsymbol{p'})\right] : L_{i+1}^m \to \mathbb{F}_q$ in less than

$$2 \cdot 2^m + 5 \cdot 2^m \left(1 + \frac{11}{20}m\right) \left|L_{i+1}^m\right| \leqslant 2^m \left(\frac{11}{4}m + 7\right) \left|L_{i+1}^m\right|$$

arithmetic operations. We deduce that the cost of honestly generating $\mathcal{P}_{\mathsf{RM}}$'s messages is

$$\sum_{i=0}^{r-1} 2^m \left(\frac{11}{4}m + 7\right) \left|L_{i+1}^m\right| < 2^m \left(\frac{11}{4}m + 7\right) \frac{n^m}{2^m - 1} \leqslant \left(\frac{11}{2}m + 14\right) n^m.$$

From the discussion about prover complexity, we also get that the number of operations made by $\mathcal{V}_{\mathsf{RM}}$ for a single consistency test is less than $2 \cdot 2^m + 5 \cdot 2^m \left(1 + \frac{11}{20}m\right)$. Thus, verifier complexity is less than $\alpha r 2^m \left(\frac{11}{4}m + 7\right)$.

Comparisons with the univariate case When we compared the FRI protocol with our IOPP for the tensor product of RS codes in Section 7.3, we argued that soundness is affected by the worse relative distance of tensor codes. In constrast, a short Reed-Muller code $\mathsf{SRM}[\mathbb{F}_q, L, m, k]$ has relative distance which is at least the one of a Reed-Solomon code $\mathsf{RS}[\mathbb{F}_q, L, k]$. However, soundness of our IOPP for Reed-Muller code is worse than soundness of the FRI protocol for linear-size field [BGKS20] due to the more complex expression of the folding operators.

In Figure 4, we present the parameters of the FRI protocol for RS codes and our IOPP for Reed-Muller codes side by side for codes of blocklength N and a single repetition of the QUERY phase. The use of balancing functions in Definition 13 induces some extra costs compared to the IOPP for product codes.

Scheme	Prover	Verifier	Query	Length	Rounds
RS IOPP [BBHR18]	$< 6N$	$< 42 \log N$	$< 2 \log N$	$< N/3$	$< \log N$
RM IOPP	$< \left(\frac{11}{2}m + 14\right) N$	$2^m \left(\frac{11}{4} + \frac{7}{m}\right) \log N$	$< \frac{2^m}{m} \log N$	$< \frac{N}{2^m - 1}$	$< \frac{\log N}{m}$

Figure 4: Comparison between the IOPP for a RS code of [BBHR18] and our IOPPs for tensor of RS codes and RM codes. Blocklength of the codes is denoted by N and m is the number of variables of the multivariate codes.

10 Conclusion

In this paper, tensor products of Reed-Solomon codes and Reed-Muller codes over fields with smooth additive subgroups or smooth multiplicative subgroups have been shown to admit quite efficient interactive oracle proofs of proximity (IOPPs). These results can be interpreted as multivariate low degree tests, i.e. given a function $f \colon L^m \to \mathbb{F}_q$, a verifier distinguishes whether f corresponds to the evaluation of a degree-d polynomial or is far in relative Hamming distance from any evaluations of low-degree polynomials, either using the notion of individual degrees or total degree. For a constant dimension m, our constructions have linear oracle proof length and prover complexity, logarithmic query and verifier complexities. In the case of tensor products of Reed-Solomon codes, our construction can be generalized to distinct degree bounds and different evaluation domain.

Many constructions of succinct non-interactive arguments (SNARG) rely on univariate polynomials for arithmetization. One of the reason is that there exists an efficient IOPP for Reed-Solomon codes [BBHR18]. Proposing highly-efficient IOPPs for multivariate polynomial codes might open up a range of different arithmetization techniques for designing explicit constructions of proof systems.

Regarding total degree tests, we think that allowing support L^m with L much smaller than $\mathbb{F}_q$ gives more flexibility in the design of proof systems. However, we had to require d to be less than $|L|$ to ensure completeness and soundness. A natural question is whether an IOPP for multivariate polynomial codes with total degree $d > |L|$ can be designed.

We also note that our proximity parameter is not as good as the one from [BCI+20], where a formal Guruswami-Sudan [GS01] decoding algorithm is analyzed for a worst-case to average-case reduction. Obtaining such a large proximity parameter would involve an analysis of algebraic list-decoding algorithms of multivariate codes over the field of rational functions. To the best of our knowledge, such algorithms are not known. Alternatively, a potential direction for improving the soundness of our IOPs of Proximity would be to develop for multivariate codes the so-called "domain extension to eliminate contenders" technique introduced in [BGKS20], as this technique improved the soundness of the FRI protocol [BBHR18].

Acknowledgments

Sarah Bordage benefits from the support of the Chair “Blockchain & B2B Platforms”, led by *l’X – École polytechnique* and the *Fondation de l’École polytechnique*, sponsored by *Capgemini*, *NomadicLabs*, *Caisse des dépôts*.

References

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 2087–2104. ACM, 2017.

[AKK+03] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing Low-Degree Polynomials over GF(2). In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, volume 2764 of *Lecture Notes in Computer Science*, pages 188–199. Springer, 2003.

[AKK+05] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing Reed-Muller codes. *IEEE Trans. Inf. Theory*, 51(11):4032–4039, 2005.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. *J. ACM*, 45(3):501–555, 1998. extended version of FOCS’92.

[AS03] Sanjeev Arora and Madhu Sudan. Improved Low-Degree Testing and its Applications. *Combinatorica*, 23(3):365–426, 2003.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 14:1–14:17, 2018.

[BBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable Zero Knowledge with No Trusted Setup. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 701–732. Springer, 2019.

[BCG+17] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. Interactive Oracle Proofs with Constant Rate and Query Complexity. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 40:1–40:15, 2017.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. Linear-Time Arguments with Sublinear Verification from Tensor Codes. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 19–46. Springer, 2020.

[BCGT13] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, and Eran Tromer. On the Concrete Efficiency of Probabilistically-Checkable Proofs. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 585–594, New York, NY, USA, 2013. Association for Computing Machinery.

[BCI+20] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed-Solomon Codes. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 900–909. IEEE, 2020.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent Succinct Arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 103–128. Springer, 2019.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive Oracle Proofs. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, pages 31–60, 2016.

[BDN17] Amey Bhangale, Irit Dinur, and Inbal Livni Navon. Cube vs. Cube Low Degree Test. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 40:1–40:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[BFL90] László Babai, Lance Fortnow, and Carsten Lund. Non-Deterministic Exponential Time Has Two-Prover Interactive Protocols. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 16–25. IEEE Computer Society, 1990.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking Computations in Polylogarithmic Time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 21–31, 1991.

[BGH+04] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of Proximity, shorter PCPs and applications to coding. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 1–10. ACM, 2004.

[BGKS20] Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: Sampling Outside the Box Improves Soundness. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, pages 5:1–5:32, 2020.

[BKS18] Eli Ben-Sasson, Swastik Kopparty, and Shubhangi Saraf. Worst-Case to Average Case Reductions for the Distance to a Code. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, pages 24:1–24:23, 2018.

[BN20] Sarah Bordage and Jade Nardi. Interactive Oracle Proofs of Proximity to Algebraic Geometry Codes. *Electron. Colloquium Comput. Complex.*, 27:165, 2020.

[BS06] Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Struct. Algorithms*, 28(4):387–402, 2006.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with Polylog Query Complexity. *SIAM J. Comput.*, 38(2):551–607, 2008.

[BSVW03] Eli Ben-Sasson, Madhu Sudan, Salil Vadhan, and Avi Wigderson. Randomness-Efficient Low Degree Tests and Short PCPs via Epsilon-Biased Sets. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 612–621, New York, NY, USA, 2003. Association for Computing Machinery.

[CMS17] Alessandro Chiesa, Peter Manohar, and Igor Shinkar. On Axis-Parallel Tests for Tensor Product Codes. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 39:1–39:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[DR04] Irit Dinur and Omer Reingold. Assignment Testers: Towards a Combinatorial Proof of the PCP-Theorem. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 155–164. IEEE Computer Society, 2004.

[FGL+96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive Proofs and the Hardness of Approximating Cliques. *J. ACM*, 43(2):268–292, 1996.

[FHS94] Katalin Friedl, Zsolt Hátsági, and Alexander Shen. Low-Degree Tests. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, page 57–64, USA, 1994. Society for Industrial and Applied Mathematics.

[GLR+91] Peter Gemmell, Richard J. Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self-Testing/Correcting for Polynomials and for Approximate Functions. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 32–42. ACM, 1991.

[GS01] Venkatesan Guruswami and Madhu Sudan. On representations of algebraic-geometry codes. *IEEE Trans. Inf. Theory*, 47(4):1610–1613, 2001.

[GS02] Oded Goldreich and Madhu Sudan. Locally Testable Codes and PCPs of Almost-Linear Length. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 13–22. IEEE Computer Society, 2002.

[JPRZ04] Charanjit S. Jutla, Anindya C. Patthak, Atri Rudra, and David Zuckerman. Testing Low-Degree Polynomials over Prime Fields. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 423–432. IEEE Computer Society, 2004.

[KPV19] Assimakis Kattis, Konstantin Panarin, and Alexander Vlasov. RedShift: Transparent SNARKs from List Polynomial Commitment IOPs. Cryptology ePrint Archive, Report 2019/1400, 2019. `https://ia.cr/2019/1400`.

[KR04] Tali Kaufman and Dana Ron. Testing Polynomials over General Fields. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 413–422. IEEE Computer Society, 2004.

[KR08] Yael Tauman Kalai and Ran Raz. Interactive PCP. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 536–547. Springer, 2008.

[LN97] Rudolf Lidl and Harald Niederreiter. *Finite Fields*. Cambridge University Press, 1997.

[Mie09] Thilo Mie. Short PCPPs Verifiable in Polylogarithmic Time with O(1) Queries. *Annals of Mathematics and Artificial Intelligence*, 56(3–4):313–338, August 2009.

[MR08] Dana Moshkovitz and Ran Raz. Sub-Constant Error Low Degree Test of Almost-Linear Size. *SIAM J. Comput.*, 38(1):140–180, 2008.

[Pan94] Victor Y. Pan. Simple Multivariate Polynomial Multiplication. *J. Symb. Comput.*, 18(3):183–186, 1994.

[PS94] Alexander Polishchuk and Daniel A. Spielman. Nearly-linear size holographic proofs. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 194–203. ACM, 1994.

[RR20] Noga Ron-Zewi and Ron D. Rothblum. Local Proofs Approaching the Witness Length [Extended Abstract]. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 846–857. IEEE, 2020.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 49–62. ACM, 2016.

[RS92] Ronitt Rubinfeld and Madhu Sudan. Self-Testing Polynomial Functions Efficiently and Over Rational Domains. In Greg N. Frederickson, editor, *Proceedings of the Third Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 27-29 January 1992, Orlando, Florida, USA*, pages 23–32. ACM/SIAM, 1992.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust Characterizations of Polynomials with Applications to Program Testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[RS97] Ran Raz and Shmuel Safra. A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 475–484. ACM, 1997.

[RVW13] Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 793–802. ACM, 2013.

[Sch80] Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM*, 27(4):701–717, 1980.

[Sta21] StarkWare. ethSTARK Documentation. Cryptology ePrint Archive, Report 2021/582, 2021. `https://ia.cr/2021/582`.

[Vid15] Michael Viderman. A combination of testability and decodability by tensor products. *Random Struct. Algorithms*, 46(3):572–598, 2015.

[VSBW13] Victor Vu, Srinath T. V. Setty, Andrew J. Blumberg, and Michael Walfish. A Hybrid Architecture for Interactive Verifiable Computation. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 223–237. IEEE Computer Society, 2013.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979.

A Proof of Lemma 5

The beginning of the proof of Lemma 5 is the same as the one of [BGKS20, Lemma 3.2]. We rewrite the proof entirely since we need to rely notations introduced along the analysis.

Proof of Lemma 5. First, observe that if the function $\phi\colon D \to [0,1]$ is constant equal to 1, then $\mu_\phi(u, C) = 1 - \Delta(u, v)$. Therefore, for any weight function $\phi\colon D \to [0,1]$ and any $u, v \in \mathbb{F}_q^D$, $\mu_\phi(u, v) \leqslant 1 - \Delta(u, v)$. Consequently, $\mu_\phi(u, C) \leqslant 1 - \Delta(u, C)$. Thus, the set

$$\{z \in \mathbb{F}_q \mid \mu_\phi(u_0 + zu_1, C) > 1 - \delta\}$$

is contained in $A := \{z \in \mathbb{F}_q \mid \Delta(u_0 + zu_1, C) < \delta\}$ and the hypothesis implies $|A| \geqslant \frac{2}{\varepsilon^2}$. Now, the proof follows the one of [BGKS20, Lemma 3.2].

For each $z \in \mathbb{F}_q$, denote $u_z = u_0 + zu_1$ and let $v_z \in C$ be a codeword such that $\Delta(u_z, C) = \Delta(u_z, v_z)$. Let $T_z := \{x \in D \mid u_z(x) = v_z(x)\}$ be the agreement set of u_z and v_z.

For z_0, z_1, z_2, picked uniformly and independently in A and y picked uniformly from D, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{z_0,z_1,z_2}\left[\frac{|T_{z_0} \cap T_{z_1} \cap T_{z_2}|}{n}\right] &= \mathop{\mathbf{E}}_{y,z_0,z_1,z_2}[\mathbf{1}_{y \in T_{z_0} \cap T_{z_1} \cap T_{z_2}}] \\
&= \mathop{\mathbf{E}}_{y}[\mathop{\mathbf{E}}_{z}[\mathbf{1}_{y \in T_z}]^3] \\
&\geqslant \mathop{\mathbf{E}}_{y,z}[\mathbf{1}_{y \in T_z}]^3 \\
&\geqslant (1-\delta)^3 \\
&\geqslant 1 - \delta + \varepsilon.
\end{aligned}$$

From this, one obtains

$$\Pr_{z_0,z_1,z_2}[|T_{z_0} \cap T_{z_1} \cap T_{z_2}| \geqslant (1-\delta)|D|] \geqslant \varepsilon.$$

The probability of z_0, z_1, z_2 being all distinct is at least $1 - \frac{3}{|A|}$, which is greater than $1 - \frac{\varepsilon}{2}$ since $|A| > \frac{6}{\varepsilon}$. Thus, we get

$$\Pr_{z_0,z_1,z_2}[z_0, z_1, z_2 \text{ are all distinct and } |T_{z_0} \cap T_{z_1} \cap T_{z_2}| \geqslant (1-\delta)|D|] \geqslant \varepsilon/2.$$

Consequently, there are distinct z_1 and z_2 such that

$$\Pr_{z_0}[|T_{z_0} \cap T_{z_1} \cap T_{z_2}| \geqslant (1-\delta)|D|] \geqslant \varepsilon/2.$$

Fix a z_0 such that $|T_{z_0} \cap T_{z_1} \cap T_{z_2}| \geqslant (1-\delta)|D|$, and let $S = T_{z_0} \cap T_{z_1} \cap T_{z_2}$. We have that $u_{z_0}, u_{z_1}, u_{z_2}$ all lie on the line $l = \{u_0 + zu_1 : z \in \mathbb{F}_q\} \subset \mathbb{F}_q^D$. As a consequence, when restricted to S, we have that $u_{z_0|S}, u_{z_1|S}, u_{z_2|S}$ all lie on the line $l_{|S} = \left\{u_{0|S} + zu_{0|S} : z \in \mathbb{F}_q\right\} \subset \mathbb{F}_q^S$.

By definition of S, T_{z_0}, T_{z_1} and T_{z_2}, we also have that $v_{z_0|S}, v_{z_1|S}, v_{z_2|S}$ lie on the line $l_{|S}$. Since S is an information set of C, we can linearly reencode $v_{z_0|S}, v_{z_1|S}, v_{z_2}$ into $v_{z_0}, v_{z_1}, v_{z_2}$, and we observe that v_{z_0}, v_{z_1} and v_{z_2} all lie on a same line. Thus, there are $v_0, v_1 \in \mathbb{F}_q^D$ such that this line is defined by $\{v_0 + zv_1; z \in \mathbb{F}_q\} \subset \mathbb{F}_q^D$. There are $\frac{\varepsilon}{2}$-fraction of the $z_0 \in A$ such that v_{z_0} belongs to this line. Notice that for such z_0, $v_{z_0} = v_0 + z_0 v_1$.

Let $A' \subset A$ be the set of the z's such that v_z (the word closest to u_z) can be written $v_z = v_0 + zv_1$. Then, we have $|A'| \geqslant \frac{\varepsilon}{2}|A| \geqslant \frac{1}{\varepsilon}$ and for all $z \in A'$, $\mu_\phi(u_0 + zu_1, v_0 + zv_1) > 1 - \delta$. Therefore,

$$\begin{aligned}
1 - \delta &< \frac{1}{|A'|} \sum_{z \in A'} \mu_\phi(u_z, v_z) \\
&< \frac{1}{|A'|\,|D|} \sum_{z \in A'} \sum_{x \in D} \left(\phi(x) \cdot \mathbf{1}_{u_z(x) = v_z(x)}\right) \\
&< \frac{1}{|D|} \sum_{x \in D} \phi(x) \cdot \left(\frac{1}{|A'|} \sum_{z \in A'} \mathbf{1}_{u_z(x) = v_z(x)}\right).
\end{aligned}$$

Let us consider $T := \{x \in D \mid u_0(x) = v_0(x) \text{ and } u_1(x) = v_1(x)\}$. Given $x \in D \backslash T$, there is at most one element $z \in \mathbb{F}_q$ such that $u_0(x) + zu_1(x) = v_0(x) + zv_1(x)$. Thus, we conclude that

$$\begin{aligned}
1 - \delta &< \frac{1}{|D|} \sum_{x \in T} \phi(x) + \frac{1}{|D|} \sum_{x \in D \backslash T} \phi(x) \frac{1}{|A'|} \\
&< \frac{1}{|D|} \sum_{x \in T} \phi(x) + \varepsilon.
\end{aligned}$$

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il