Electronic Colloquium on Computational Complexity, Report No. 168 (2012)

Strong LTCs with inverse polylogarithmic rate and soundness

Michael Viderman*
Computer Science Department
Technion — Israel Institute of Technology
Haifa, 32000, Israel.
{viderman}@cs.technion.ac.il

November 26, 2012

Abstract

An error-correcting code $C \subseteq \mathbb{F}^n$ is called (q, ϵ)-strong locally testable code (LTC) if there exists a randomized algorithm (tester) that makes at most q queries to the input word. This algorithm accepts all codewords with probability 1 and rejects all non-codewords $x \notin C$ with probability at least $\epsilon \cdot \delta(x, C)$, where $\delta(x, C)$ denotes the relative Hamming distance between the word x and the code C. The parameter q is called the query complexity and the parameter ϵ is called soundness.

A well-known open question in the area of LTCs (Goldreich and Sudan, J.ACM 2006) asks whether exist strong LTCs with constant query complexity, constant soundness and inverse polylogarithmic rate.

In this paper, we construct strong LTCs with query complexity 3, inverse polylogarithmic soundness and inverse polylogarithmic rate.

*The research has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 259426.

ISSN 1433-8092

Contents

1 Introduction

A linear code over a finite field $\mathbb{F}$ is a linear subspace $\mathcal{C} \subseteq \mathbb{F}^n$. In this case, n is the blocklength of the code $\mathcal{C}$, denoted by blocklength($\mathcal{C}$). The dimension of $\mathcal{C}$, denoted by $\dim(\mathcal{C})$, is its dimension as a vector space. The rate of $\mathcal{C}$, denoted by rate($\mathcal{C}$), is defined to be $\frac{\dim(\mathcal{C})}{\text{blocklength}(\mathcal{C})} = \frac{\dim(\mathcal{C})}{n}$. We define the distance between two words $x, y \in \mathbb{F}^n$ to be $\Delta(x,y) = |\{i \mid x_i \neq y_i\}|$ and the relative distance to be $\delta(x,y) = \frac{\Delta(x,y)}{n}$. The distance of $\mathcal{C}$ is defined by $\Delta(\mathcal{C}) = \min_{x \neq y \in \mathcal{C}} \Delta(x,y)$ and its relative distance is defined by $\delta(\mathcal{C}) = \frac{\Delta(\mathcal{C})}{n}$. We note that $\Delta(\mathcal{C}) = \min_{c \in \mathcal{C} \setminus \{0\}} \{\text{wt}(c)\}$. One is typically interested in codes whose distance is linear to the blocklength of $\mathcal{C}$, i.e., $\Omega(n)$.

For $x \in \mathbb{F}^n$ and $\mathcal{C} \subseteq \mathbb{F}^n$, let $\delta(x,\mathcal{C}) = \min_{y \in \mathcal{C}} \{\delta(x,y)\}$ denote the relative distance of x from the code $\mathcal{C}$. If $\delta(x,\mathcal{C}) \geq \rho$, we say that x is ρ-far from $\mathcal{C}$ and otherwise x is ρ-close to $\mathcal{C}$.

In this work we investigate *locally testable codes* (LTCs). A code $\mathcal{C}$ is said to be (q,ϵ,ρ)-weak LTC if there exists a randomized algorithm T, called tester, that makes at most q queries to the input word w. If $w \in \mathcal{C}$ then T accepts w with probability 1, but if w is ρ-far from C the tester T rejects w with probability at least ϵ. Let us notice that the tester is not required to reject when $0 < \delta(w,\mathcal{C}) < \rho$. This is the reason why such codes are called *weak* LTCs.

In contrast to weak LTCs, the testers for strong LTCs are required to reject all non-codewords with corresponding probability. More formally, a code $\mathcal{C}$ is called (q,ϵ)-strong LTC if there exists a tester T that makes at most q queries to the input word w. If $w \in \mathcal{C}$ then T accepts w with probability 1, but if $w \notin \mathcal{C}$ then T rejects w with probability at least $\epsilon \cdot \delta(w,\mathcal{C})$. The parameter q is called the query complexity and the parameter ϵ is called soundness.

Informally, we say that a code $\mathcal{C}$ is a weak LTC if it has a linear distance and there exist constants $q, \epsilon > 0$ and $\rho \leq \delta(\mathcal{C})/3$ such that C is a (q,ϵ,ρ)-weak LTC. [1] Similarly, we say that a code $\mathcal{C}$ is a strong LTC if it has a linear distance and there exist constants $q, \epsilon > 0$ such that $\mathcal{C}$ is a (q,ϵ)-strong LTC.

Such codes are of interest in computer science due to their numerous connections to probabilistically checkable proofs (PCPs) and property testing (see the surveys [30, 21] for more information). LTCs were implicit already in [2] (cf. [21, Sec. 2.4]) and they were explicitly studied by Goldreich and Sudan [24]. The best known strong LTCs are due to Goldreich and Sudan [24], who presented probabilistic construction of strong LTCs. These LTCs achieve constant query complexity, constant soundness and rate $\frac{1}{\exp(\tilde{O}(\sqrt{\log n}))}$, where n denotes the blocklength.

Later, other constructions of LTCs [9, 17, 27] succeeded to obtain the rate $\frac{1}{\text{polylog}(n)}$ together with constant query complexity and soundness, however these codes were weak LTCs. It can be verified that every strong LTC is also a weak LTC (see Claim B.4), but some weak LTCs are not strong LTCs (see Proposition B.1). So, strong LTCs are strictly stronger objects than weak LTCs. In the journal version of [24], the authors pointed out that all known LTCs that achieve inverse polylogarithmic rate are weak LTCs, and asked about the existence of strong LTCs with polylogarithmic rate [24, Section 6].

[1] The parameter ρ is required to be less than $\delta(\mathcal{C})/2$ to avoid trivial solutions like claiming that every perfect code $\mathcal{C}$ is a $(0,1,\delta(\mathcal{C})/2)$-weak LTC. Recall that a code $\mathcal{C} \subseteq \mathbb{F}^n$ is called perfect if there are no words in $\mathbb{F}^n$ that are $(\delta(\mathcal{C})/2)$-far from $\mathcal{C}$. So, in this case one could say that no queries are needed and all $(\delta(\mathcal{C})/2)$-far words are rejected with probability 1 vacuously.

In this paper (Theorem 3.1) we construct binary linear 3-query strong LTCs with inverse polylogarithmic rate, inverse polylogarithmic soundness and linear distance. To the best of our knowledge this range of the parameters for strong LTCs was not obtained before. We notice that every tester for a (non-trivial) linear LTC has query complexity at least 3 (see [6])[2]. Although our construction does not resolve the open question raised in [24], we believe that our results make this question easier to solve. Let us explain why.

The best known weak LTCs [9, 17, 27] were constructed in two steps. In the first stage, the authors presented a construction of weak LTCs that achieved constant query complexity, inverse polylogarithmic rate, inverse polylogarithmic soundness and linear distance. Then, in the second stage, the gap amplification technique of Dinur [17] was applied to increase the soundness from inverse polylogarithmic to constant, while, roughly speaking, preserving all other parameters.

So, in this paper we show a construction of strong LTCs which have similar parameters to the weak LTCs from the first stage above. We stress that when this gap amplification is applied to strong LTCs, it will produce weak LTCs. Thus using the result of Dinur [17], our work implies weak LTCs with best known parameters as in [9, 17, 27]. This gap amplification technique is also known to preserve the linearity of the underlying LTCs (see e.g.,[27, Section 6.4]). Indeed, the arguments of [27, Section 6.4] show that this procedure can be applied to linear strong LTCs to increase its soundness parameter and this procedure even preserves the linearity of the underlying LTC. [3] However, this procedure outputs a weak LTC, or more precisely, a code accompanied with a probabilistically checkable proof that could be translated to a weak LTC.

Thus, in order to resolve the question raised in [24], one should modify the gap amplification technique to preserve the strong testability of the underlying LTCs. We believe that this is a feasible task but in this paper we were unable to do this.

1.1 The construction of Meir [27] vs. Our construction

We recall the open question of [24] about the strong LTCs with inverse polylogarithmic rate. This question was raised again in the work of Meir [27], who showed an alternative construction of weak LTCs that achieve constant query complexity, constant soundness and inverse polylogarithmic rate, which is the best known range of parameters for weak LTCs. Meir stressed that codes presented in [27] are weak LTCs and raised again the question about strong LTCs of polylogarithmic rate (see [27, Section 7.3]). More precisely, Meir constructed another kind of linear codes, called codes with proofs (CWPs). Informally, $\mathcal{C} \subseteq \mathbb{F}^n$ is a (q, ϵ)-CWP if all coordinates of the code are partitioned into two subsets $I \subseteq [n]$ and $[n] \setminus I$, and there exists a tester T that makes at most q queries to the given word $w \in \mathbb{F}^n$. This tester accepts with probability 1 if $w \in \mathcal{C}$, and otherwise rejects with probability at least $\epsilon \cdot \delta(w|_I, \mathcal{C}|_I)$.

It can be verified that a CWP can be easily converted to a weak LTC by repeating all bits indexed by I several times (see [27, Section 6.2]). However, a CWP is not equivalent to a strong LTC since the guarantee is only that the rejection probability of the tester is proportional to $\delta(w|_I, \mathcal{C}|_I)$. So, it can be the case that $w \notin \mathcal{C}$, but $w|_I \in \mathcal{C}|_I$ and the tester accepts with probability 1. Let us

[2] Ben-Sasson et al. [6] showed that testing any linear code with linear distance and non-constant dimension requires at least 3 queries.

[3] Meir [27] considered linear codes with proof (CWP) which are a special case of PCPs of proximity [9, 17, 18]. Such CWPs can be easily converted to weak LTCs by repeating all "code coordinates" several times (see [27, Section 6.2]). On the other hand, every (linear) strong LTC is a special case of a linear CWP, where the proof is empty. Thus the arguments of [27, Section 6.4] are valid for the strong LTCs as well.

cite from [27, Section 7.3]:

"Our construction does not yield strong LTCs because the transformation from CWPs to LTCs loses the strong rejection property. Thus, it remains an open problem to give a combinatorial construction of strong LTCs. It seems to us that such a construction will have to be very different from our construction, since it will not be able to use CWPs."

Surprisingly, our construction is almost identical to the construction of [27], and provides a family of strong LTCs. Our construction can be considered combinatorial according to [27, Remark 1.1], although we do not stress this in the paper. In the next section we explain shortly our contributions that allowed us to turn the codes of [27] into the strong LTCs.

1.1.1 The main observation — Core Oriented LTCs.

First, in Definition 2.1 we define a core of the code $\mathcal{C} \subseteq \mathbb{F}^n$ to be a (small) subset of indices, denoted by $A(\mathcal{C})$, such that $\mathcal{C}|_{A(\mathcal{C})}$ has the same dimension as $\mathcal{C}$. Intuitively, think that a core is very small, i.e., $|A(\mathcal{C})| = o(n)$, but the coordinates indexed by $A(\mathcal{C})$ are much more important than the other coordinates of the code. Next, we define a core oriented distance (Definition 2.2) between a word w and the code $\mathcal{C}$, denoted by $\delta_{A(\mathcal{C})}(w, \mathcal{C})$ that devotes especial importance to the coordinates in $A(\mathcal{C})$. Saying that w is close to the code $\mathcal{C}$ with respect to the core oriented distance implies immediately that there exists a codeword $c \in \mathcal{C}$ such that $\delta(w|_{A(\mathcal{C})}, c|_{A(\mathcal{C})})$ is small as well as $\delta(w, c)$.

Informally, a tester of a CWP (defined in [27]) provides an ability to test only the core of the code, while a tester of a strong LTC tests the entire code with no especial importance to the core coordinates. That means, whenever a tester of a CWP rejects w with a small probability we know that $\delta(w|_{A(\mathcal{C})}, \mathcal{C}|_{A(\mathcal{C})})$ is small, but we do not know whether $\delta(w, \mathcal{C})$ is small. On the other hand, if a tester of a strong LTC rejects w with a small probability, then we know that $\delta(w, \mathcal{C})$ is small but we do not know whether $\delta(w|_{A(\mathcal{C})}, \mathcal{C}|_{A(\mathcal{C})})$ is small.

We realized that a combination of both these features inside a single tester would be highly useful. Definition 2.4 contains the definition of core oriented LTCs (COLTCs). In words, a code $\mathcal{C}$ is a (q, ϵ)-COLTC if it has a tester that makes at most q queries to an input word w. This tester always accepts all codewords of $\mathcal{C}$ and rejects w with probability at least $\epsilon \cdot \delta_{A(C)}(w, \mathcal{C})$. In this way, the fact that a COLTC-tester rejects w with a small probability guarantees the existence of a codeword $c \in \mathcal{C}$ such that both $\delta(w, c)$ and $\delta(w|_{A(\mathcal{C})}, c|_{A(\mathcal{C})})$ are small. In particular, any COLTC is a strong LTC and a CWP[4].

The above observation becomes crucial in the iterative construction of LTCs, where we claim that every iteration the underlying code is a COLTC, and not just a CWP or a strong LTC. This lead us to the need to revise the definition of a standard notion of robust testing (Section 4) and design a definition of "core robustness" in Definition 7.1. It can be proven that the iterative construction suggested in [27] preserves this notion of "core robustness". It turns out that if COLTCs are involved in the iterative construction that preserves "core robustness", then the resulting code is a COLTC as well (see Claim 7.2).

The main step in this iterative construction is a product of codes, where a tensor product (Section 2.1) is applied only on a small fraction of codeword entries. This product of codes was

[4]We notice that saying "a code $\mathcal{C}$ is a CWP and is a strong LTC" is weaker than saying "$\mathcal{C}$ is a COLTC" since the fact that $\delta(w|_{A(\mathcal{C})}, \mathcal{C}|_{A(\mathcal{C})})$ and $\delta(w, \mathcal{C})$ are small does not imply that there exists a **single** codeword $c \in \mathcal{C}$ such that $\delta(w|_{A(\mathcal{C})}, c|_{A(\mathcal{C})})$ and $\delta(w, c)$ are small.

used in [27]. We define it explicitly and call it the star product of codes (Section 7). In words, in this star product the tensor is taken only on the core coordinates $A(C)$ and this is the reason why these coordinates play especially important role in the construction.

We would like to notice that if one takes repetitive standard tensor products then the rate of the code decreases drastically (see Section 2.1). On the other hand, when we use the star product and take care that $A(\mathcal{C})$ stays small in every iteration, then the rate decreases only by a constant in every iteration.

1.1.2 Our contributions to the construction of [27].

So, as we said previously our construction is almost the same as the construction of [27], and our main contribution is the modified proof showing that the obtained codes are strong LTCs and not just weak LTCs as were claimed in [27]. In this section we would like to mention briefly our modifications to the construction of [27].

How to avoid the increase in query complexity. One of the drawbacks of the repeated tensor product operation, even when it is applied only to a small fraction of coordinates, is that this operation strongly decreases the distance of the base codes. Meir [27] observed that the drastic decrease in the distance can be avoided using a "distance amplification" procedure DistAmp$(\cdot)$ which preserves local testability. I.e., given any locally testable code $\mathcal{C} \subseteq \mathbb{F}^n$ the distance amplification procedure outputs a locally testable code $\mathcal{C}' = \text{DistAmp}(\mathcal{C})$ such that $\dim(\mathcal{C}') = \dim(\mathcal{C})$ and $\text{rate}(\mathcal{C}') = \Omega(\text{rate}(\mathcal{C}))$. Furthermore, DistAmp$(\cdot)$ improves the distance of the code and preserves its local testability, but unfortunately, the query complexity is multiplied by some constant integer $\beta \geq 2$. During the construction of [27], the procedure DistAmp$(\cdot)$ was applied $\Theta(\log \log n)$ times, thus resulting in the query complexity polylog(n).

Now, if one is interested to obtain a weak LTC (or a CWP) with constant query complexity, then the query complexity can be reduced to 3 (see [27, Section 6.3] and [14, Section A.1]). But the problem arises when the goal is to obtain a strong LTC. The standard query complexity reduction technique is not known to preserve the strong local testability and it seems that it should not.

We solve this issue as follows. In Sections 8 and 9 we implement our version of DistAmp$(\cdot)$ procedure, which will improve the distance of the code, and on the other hand, the query complexity remains intact. In particular, if $\mathcal{C}$ is a 3-query COLTC and has a small distance then $\mathcal{C}' = \text{DistAmp}(\mathcal{C})$ is a 3-query COLTC, $\mathcal{C}'$ has larger distance than $\mathcal{C}$ and $\text{rate}(\mathcal{C}') = \Omega(\text{rate}(\mathcal{C}))$. This gives us an opportunity to apply DistAmp$(\cdot)$ procedure many times and the query complexity will stay unaffected.

A large distance of the code is preserved through way. In the work [27], the author cared to obtain CWPs, where the large distance was preserved only inside the core coordinates $(\mathcal{C}|_{A(\mathcal{C})})$, i.e., $\delta(\mathcal{C}|_{A(\mathcal{C})})$ was large but $\delta(C)$ was very small. Hence, to obtain a weak LTC with a good distance one is required to make many repetitions of the entries indexed by $A(\mathcal{C})$ (see [27, Section 6.2]).

It turns out that to prove the "core robustness" results for the star products of COLTCs (Section 7) we need to assume that the underlying COLTCs have large distance. Thus in our construction, we preserve through the way a large distance of the entire code $(\delta(\mathcal{C}))$ and not only the distance inside the core coordinates $(\delta(\mathcal{C}|_{A(\mathcal{C})}))$. Thus we also avoid the need to use a separate stage of repetitions at the end of the construction.

1.2 A new product of codes and a new analysis of robustness

Let us introduce another part of this paper, which is not involved in the construction of COLTCs. In [27] the author pointed out that his products of the base code $\mathcal{C} \subseteq \mathbb{F}^n$ rely on the assumption that a part of its coordinates must form a tensor product for some linear code. In our language, to obtain a desired result we need to assume that $\mathcal{C}|_{A(\mathcal{C})} = R \otimes R$ for some linear code R. Meir raised a question of whether one can avoid this assumption.

So, another contribution of this paper is that we define explicitly a new product of codes, called L-product (Section 5), which is similar to the star product but it does not assume that an underlying code has some particular structure. Similarly to the star products, in the L-products of codes the tensor product operation is applied only to the core coordinates $\mathcal{C}|_{A(\mathcal{C})}$. This operation reduces the rate of the underlying code only by a constant given that $A(\mathcal{C})$ is sufficiently small, while the standard tensor product squares the rate of the code (see Section 5). Also, L-product of codes affects the distance of the underlying code in the same way as the standard tensor product operation.

It is worth to stress that the investigation of L-products does not involved in the proof of our main result (Theorem 3.1), and comes on its own way. In spite of that it is interesting to notice that although L-products seem similar to the star products (Definition 7.4), and they can be robustly testable (Section 5.3) with regards to the standard notion of robustness (see Section 4), the author of this paper did not succeed to use them to construct efficient strong LTCs. More specifically, we were unable to implement the random projection operation for this type of code products, while it can be implemented for the star products (Section 10) exactly as it appeared in [27, Section 4.2].

1.2.1 Robust testing of codes

One of the basic concepts behind the constructions of some LTCs [5, 10, 32] was the concept of "robust testing". Let us notice that without loss of generality a tester for a linear code selects a small subset of entries, called a "local view", and then accepts if and only if this local view is consistent [8, 14]. In words, a code $\mathcal{C}$ is called robustly testable with respect to a tester T if given any non-codeword w that is far from $\mathcal{C}$, it holds that a typical local view of the tester T is far from being consistent.

In Section 4 we identify the properties sufficient for robust testing. I.e., if a code C and its tester T satisfy these properties then $\mathcal{C}$ is robustly testable. We proceed and prove in Section 5.3 that the repeated L-products of codes can be robustly testable. Then we reprove in Section 6 the robust testability of the repeated tensor product of codes [10, 32] using aforementioned observations to demonstrate the simplicity of such a proof.

Organization of the paper.

In the following section we provide some preliminary definitions and the new concepts. Then, in Section 3 we state our main results and, in particular, Theorem 3.1.

We present sufficient properties for the code to be robustly testable in Section 4. In Section 5 we define a new product of codes, called L-products, and show that they are robustly testable. Then, in Section 6 we reprove the main result of [32] using the general observations made in Section 4. In Section 7 we define the star products of codes (implicitly used in [27]) and show that these codes are robustly testable with regards to the "core robustness" notion presented in the same section.

Then, we turn to define and study the distance amplification procedure in Sections 8 and 9 and the random projection operation in Section 10. Finally, we prove our main results in Section 11.

The auxiliary proofs are postponed to Appendix: Sections A and B.

2 Preliminaries

In this work, we consider only linear codes. Let $\mathbb{F}$ be a finite field and $[n]$ be the set $\{1,\ldots,n\}$. For $w \in \mathbb{F}^n$, let $\mathrm{supp}(w) = \{i \in [n] \mid w_i \neq 0\}$ and $|w| = |\mathrm{supp}(w)|$. For $u = (u_1, u_2, \ldots, u_n)$, $v = (v_1, v_2, \ldots, v_n) \in \mathbb{F}^n$ let $\langle u, v\rangle$ denote the bilinear function from $\mathbb{F}^n \times \mathbb{F}^n$ to $\mathbb{F}$ defined by $\langle u, v\rangle = \sum_{i=1}^{n} u_i v_i$. The dual code $\mathcal{C}^\perp$ is defined as $\mathcal{C}^\perp = \{u \in \mathbb{F}^n \mid \forall c \in \mathcal{C} :\ \langle u, c\rangle = 0\}$. Similarly, we define $\mathcal{C}^\perp_{\leq t} = \left\{u \in \mathcal{C}^\perp \mid |u| \leq t\right\}$ and $\mathcal{C}^\perp_{t} = \left\{u \in \mathcal{C}^\perp \mid |u| = t\right\}$. For $w \in \mathbb{F}^n$ and $S = \{j_1, j_2, \ldots, j_m\} \subseteq [n]$ we let $w|_S = (w_{j_1}, w_{j_2}, \ldots, w_{j_m})$, where $j_1 < j_2 < \ldots < j_m$, be the restriction of w to the subset S. Similarly, we let $\mathcal{C}|_S = \{c|_S \mid c \in \mathcal{C}\}$ denote the projection of the code $\mathcal{C}$ onto S. We define $\mathcal{C}|_{-S} = \mathcal{C}|_{[n]\setminus S}$, i.e., projection of the code $\mathcal{C}$ to all coordinates besides S.

Every linear code $\mathcal{C}$ is associated with a set of its coordinates, denoted $\mathrm{coord}(\mathcal{C})$. For example if $\mathcal{C} \subseteq \mathbb{F}^n$ then $\mathrm{coord}(\mathcal{C}) = [n]$. It follows that for every code $\mathcal{C}$ over the field $\mathbb{F}$ it holds that $\mathcal{C} \subseteq \mathbb{F}^{\mathrm{coord}(\mathcal{C})}$. The blocklength of the code, denoted by $\mathrm{blocklength}(\mathcal{C})$, is equal to $|\mathrm{coord}(\mathcal{C})|$.

For two words $x_1 \in \mathbb{F}^{n_1}$ and $x_2 \in \mathbb{F}^{n_2}$ we define (x_1, x_2) to be a word in $\mathbb{F}^{n_1+n_2}$ such that $(x_1, x_2)|_{[n_1]} = x_1$ and $(x_1, x_2)|_{([n_1+n_2]\setminus[n_1])} = x_2$. With some abuse of notation, if C is a linear code over the field $\mathbb{F}$ and $I \subseteq \mathrm{coord}(\mathcal{C})$ we use $(\mathcal{C}|_I, \mathcal{C}|_{\mathrm{coord}(\mathcal{C})\setminus I})$ to glue two parts of the code, i.e., $\mathcal{C} = (\mathcal{C}|_I, \mathcal{C}|_{\mathrm{coord}(\mathcal{C})\setminus I})$. In this case, for every $c \in \mathcal{C}$ we have $c = (c|_I, c|_{\mathrm{coord}(\mathcal{C})\setminus I})$. This expression will be used when the goal is to stress that the code $\mathcal{C}$ contains two "different" parts.

2.1 Tensor Product of Codes

The definitions appearing here are standard in the literature on tensor-based LTCs (e.g. [10, 11, 12, 19, 27, 31]).

For $x \in \mathbb{F}^{n_1}$ and $y \in \mathbb{F}^{n_2}$ we let $x \otimes y$ denote the tensor product of x and y (i.e., the matrix M with entries $M_{(i,j)} = x_j \cdot y_i$ where $(i, j) \in [n_2] \times [n_1]$). Let $R \subseteq \mathbb{F}^{n_1}$ and $C \subseteq \mathbb{F}^{n_2}$ be linear codes. We define the tensor product code $R \otimes C$ to be the linear space spanned by words $r \otimes c \in \mathbb{F}^{n_2 \times n_1}$ for $r \in R$ and $c \in C$. Some known facts regarding the tensor products (see e.g., [19]):

- The code $R \otimes C$ consists of all $n_2 \times n_1$ matrices over $\mathbb{F}$ whose rows belong to R and columns belong to C,
- $\dim(R \otimes C) = \dim(R) \cdot \dim(C)$,
- $\mathrm{rate}(R \otimes C) = \mathrm{rate}(R) \cdot \mathrm{rate}(C)$,
- $\delta(R \otimes C) = \delta(R) \cdot \delta(C)$,

We let $C^{\otimes 1} = C$ and $C^{\otimes m} = C^{\otimes(m-1)} \otimes C$ for $m > 1$. Note by this definition, $C^{\otimes 2^0} = C$ and $C^{\otimes 2^m} = C^{\otimes 2^{m-1}} \otimes C^{\otimes 2^{m-1}}$ for $t > 0$. We also notice that for a code $C \subseteq \mathbb{F}^n$ and $m \geq 1$ it holds that $\mathrm{rate}(C^{\otimes m}) = (\mathrm{rate}(C))^m$, $\delta(C^{\otimes m}) = (\delta(C))^m$ and the blocklength of $C^{\otimes m}$ is n^m.

We notice that if $\mathrm{coord}(C) = [n]$ then a set of coordinates of $C \otimes C$ is $\mathrm{coord}(C \otimes C) = [n] \times [n]$.

2.2 New Concepts and Definitions

We start by defining the following auxiliary concept.

Definition 2.1 (A core of the code)**.** Let $C \subseteq \mathbb{F}^n$ be a linear code. A core of the code, denoted by $A(C)$, is a nonempty subset of $[n]$ such that $\dim(C|_{A(C)}) = \dim(C)$, and if $A(C) \neq [n]$ then $\dim(C|_{A(C)}) = \dim(C) = \dim(C|_{-A(C)})$. Clearly, there might be many options for $A(C)$, and in this case we fix only one such option.

We say that $A(C)$ is a γ-core of the code C if $A(C)$ is a core of C, $\delta(C|_{A(C)}) = \frac{\Delta(C|_{A(C)})}{|A(C)|} \geq \gamma$, and if $A(C) \neq [n]$ then $\delta(C|_{-A(C)}) = \frac{\Delta(C|_{-A(C)})}{n-|A(C)|} \geq \gamma$.

Since $A(C) \subseteq [n]$ is a subset, we note that $A(C)$ and $[n] \setminus A(C)$ is a partition of $[n]$.

We notice that under the assumption that $A(C) \neq [n]$, $\dim(C|_{A(C)}) = \dim(C|_{-A(C)}) = \dim(C)$ and so, it holds that for all $c_1, c_2 \in C$: $c_1|_{A(C)} = c_2|_{A(C)}$ if and only if $c_1|_{-A(C)} = c_2|_{-A(C)}$. I.e., the bits in $A(C)$ defines all bits in $[n] \setminus A(C)$ and vise versa. We also notice that

$$\delta(C) = \frac{\Delta(C)}{n} = \frac{\Delta(C|_{A(C)}) + \Delta(C|_{-A(C)})}{n} \geq \frac{\gamma \cdot |A(C)| + \gamma \cdot (n - |A(C)|)}{n} = \gamma,$$

where we used that $\Delta(C) = \Delta(C|_{A(C)}) + \Delta(C|_{-A(C)})$ because, under an assumption that $A(C) \neq [n]$, two codewords of C are equal on $A(C)$ if and only if they equal on $[n] \setminus A(C)$.

Definition 2.2 (Core oriented distance)**.** Assume $C \subseteq \mathbb{F}^n$ is a linear code and $A(C)$ is its core. We define a core oriented distance between two words $w, w' \in \mathbb{F}^n$ to be

$$\delta_{A(C)}(w, w') = \max\left\{\delta(w, w'), \delta(w|_{A(C)}, w'|_{A(C)})\right\},$$

and a core oriented distance between the word $w \in \mathbb{F}^n$ and the code C to be

$$\delta_{A(C)}(w, C) = \min_{c \in C}\left\{\delta_{A(C)}(w, c)\right\}.$$

We note that for every code $C \subseteq \mathbb{F}^n$ with a core $A(C)$ and $w \in \mathbb{F}^n$ it holds that $\delta_{A(C)}(w, C) \geq \delta(w, C)$.

Remark 2.3. We stress that the standard triangle inequality holds with regards to the core oriented distance, i.e., for every $w_1, w_2, w_3 \in \mathbb{F}^n$ it holds that

$$\delta_{A(C)}(w_1, w_2) + \delta_{A(C)}(w_2, w_3) =$$

$$= \max\left\{\delta(w_1, w_2), \delta(w_1|_{A(C)}, w_2|_{A(C)})\right\} + \max\left\{\delta(w_2, w_3), \delta(w_2|_{A(C)}, w_3|_{A(C)})\right\} \geq$$

$$\geq \max\left\{\delta(w_1, w_2) + \delta(w_2, w_3), \delta(w_1|_{A(C)}, w_2|_{A(C)}) + \delta(w_2|_{A(C)}, w_3|_{A(C)})\right\} \geq$$

$$\geq \max\left\{\delta(w_1, w_3), \delta(w_1|_{A(C)}, w_3|_{A(C)})\right\} = \delta_{A(C)}(w_1, w_3).$$

A *standard q-query tester* for a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ is a randomized algorithm T that on the input word $w \in \mathbb{F}^n$ picks non-adaptively a subset $I \subseteq [n]$ such that $|I| \leq q$. Then T reads all symbols of $w|_I$ and accepts if $w|_I \in \mathcal{C}|_I$, and rejects otherwise (see [8, Theorem 2]). Hence a q-query tester can be associated with a distribution over subsets $I \subseteq [n]$ such that $|I| \leq q$.

Definition 2.4 (LTCs and Testers)**.** A q-query tester for a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ is a distribution $\mathcal{D}$ over subsets $I \subseteq [n]$ such that $|I| \leq q$.

- A q-query tester $\mathcal{D}$ is a (q, ϵ, ρ)-tester if for all $w \in \mathbb{F}^n$, $\delta(w, \mathcal{C}) \geq \rho$ we have $\Pr_{I \sim \mathcal{D}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon$.
- A q-query tester $\mathcal{D}$ is a (q, ϵ)-strong tester if for all $w \in \mathbb{F}^n$ we have $\Pr_{I \sim \mathcal{D}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon \cdot \delta(w, \mathcal{C})$.
- A q-query tester $\mathcal{D}$ is a (q, ϵ)-COLTC tester if, given that $A(\mathcal{C})$ is a core of $\mathcal{C}$, for all $w \in \mathbb{F}^n$ we have $\Pr_{I \sim \mathcal{D}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon \cdot \delta_{A(\mathcal{C})}(w, \mathcal{C})$.

A code $\mathcal{C} \subseteq \mathbb{F}^n$ is a (q, ϵ, ρ)-*weak LTC* if it has a (q, ϵ, ρ)-tester. A code $\mathcal{C} \subseteq \mathbb{F}^n$ is a (q, ϵ)-*strong LTC* if it has a (q, ϵ)-strong tester. A code $\mathcal{C} \subseteq \mathbb{F}^n$ is a (q, ϵ)-*COLTC* if it has a (q, ϵ)-COLTC tester.

Remark 2.5. Although the tester in Definition 2.4 does not output accept or reject, the way a standard tester does, it can be converted to output accept, reject as follows. Whenever the task is to test whether $w \in \mathcal{C}$ and a subset $I \subseteq [n]$ is selected by the tester, the tester can output accept if $w|_I \in \mathcal{C}|_I$ and otherwise output reject. In this manner, the tester always accepts the codewords of $\mathcal{C}$. We notice that the linearity of $\mathcal{C}$ implies that $\mathcal{C}|_I$ is a linear code.

Remark 2.6. Let $\mathcal{C}$ be a linear code and $A(\mathcal{C})$ be its core. If $\mathcal{C}$ is a (q, ϵ)-COLTC (with respect to the tester $\mathcal{D}_{\mathcal{C}}$) then $\mathcal{C}$ is a (q, ϵ)-strong LTC. To see this let $w \in \mathbb{F}^n$ and note that

$$\Pr_{I \sim \mathcal{D}_{\mathcal{C}}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon \cdot \delta_{A(C)}(w, \mathcal{C}) \geq \epsilon \cdot \delta(w, \mathcal{C}).$$

3 Main Results

We state our main result in Theorem 3.1 that gives a probabilistic construction of linear strong LTCs over the binary field. All code constructions we present in this paper can be taken directly over any field. For simplicity of the presentation, we prefer to use the binary field.

Theorem 3.1 (Main Theorem)**.** *There exists a probabilistic algorithm that constructs (with positive probability) a family of binary linear codes $\{C_m\}_m$, where C_m has a core $A(C_m)$, such that letting $k_m = \dim(C_m)$ and $n_m = \text{blocklength}(C_m)$ for every $m \geq 1$ we have*

- *$k_m = 2^{2^m}$ and $\text{rate}(C_m) \geq \frac{1}{\text{polylog}(k_m)} = \frac{1}{\text{polylog}(n_m)}$,*
- *$C_m \subseteq \mathbb{F}_2^{n_m}$ is a $(3, \frac{1}{\text{polylog}(n_m)})$-COLTC and thus is a $(3, \frac{1}{\text{polylog}(n_m)})$-strong LTC (see Remark 2.6), and*
- *$\delta(C_m) = \Omega(1)$.*

The proof of Theorem 3.1, which describes the probabilistic construction of the required family of strong LTCs, is postponed to Section 11.

The rejection probability of a tester can be easily amplified by sampling the tester polylog(n) times. This proves that the strong LTCs presented in Theorem 3.1 have testers with polylogarithmic query complexity and constant soundness. Thus we conclude the following corollary.

Corollary 3.2. *There exists a probabilistic algorithm that constructs a family of linear codes $\{C_m\}_m$ such that*

- $C_m \subseteq \mathbb{F}_2^{n_m}$ *is a* $(\text{polylog}(n_m), \frac{1}{2})$*-strong LTCs,*
- $\delta(C_m) = \Omega(1)$*, and*
- $\dim(C_m) = 2^{2^m}$ *and* $\text{rate}(C_m) \geq \frac{1}{\text{polylog}(n_m)}$.

The proof of Corollary 3.2 appears in Section 11.

4 Properties sufficient for Robust Testing

In this section we define some properties of codes that are sufficient for robust testing. We start this section by defining the notion of *robustness* (Definition 4.2) as was introduced in [10]. To do that we provide the definition of *local distance* (Definition 4.1), which will be used in Definition 4.2 and later in our proofs. Then we proceed to define "nice codes" in Definition 4.3 and to state Theorem 4.4 which argues that nice codes are robustly testable.

In this section we use n to denote the blocklength of the code C, i.e., $n = |\text{coord}(C)|$. Without loss of generality we assume that $\text{coord}(C) = [n]$.

Definition 4.1 (Local distance)**.** Let $C \subseteq \mathbb{F}^n$ be a code and $w|_I$ be the view on the coordinate set $I \subseteq [n]$ obtained from the word $w \in \mathbb{F}^n$. The *local distance* of w from C with respect to I is $\Delta(w|_I, C|_I) = \min_{c \in C} \{\Delta(w|_I, c|_I)\}$ and similarly the *relative local distance* of w from C with respect to I is $\delta(w|_I, C|_I) = \min_{c \in C} \{\delta(w|_I, c|_I)\}$.

Informally, we say that a tester is robust if for every word that is far from the code, the tester's view is far on average from any consistent view. This notion was defined for LTCs following an analogous definition for PCPs [5, 17]. We are ready to provide a general definition of robustness.

Definition 4.2 (Robustness)**.** Given a tester (i.e., a distribution) $\mathbf{D}$ for the code $C \subseteq \mathbb{F}^n$, we let

$$\rho^{\mathbf{D}}(w) = \mathop{\mathbf{E}}_{I \sim \mathbf{D}} [\delta(w|_I, C|_I)] \text{ be the expected relative local distance of input } w.$$

We say that the tester $\mathbf{D}$ has robustness $\rho^{\mathbf{D}}(C)$ on the code C if for every $w \in \mathbb{F}^n$ it holds that $\rho^{\mathbf{D}}(w) \geq \rho^{\mathbf{D}}(C) \cdot \delta(w, C)$.

Let $\{C_n\}_n$ be a family of codes where C_n is of blocklength n and $\mathbf{D_n}$ is a tester for C_n. A family of codes $\{C_n\}_n$ is *robustly testable* with respect to testers $\{\mathbf{D_n}\}_n$ if there exists a constant $\alpha > 0$ such that for all n we have $\rho^{\mathbf{D_n}}(C_n) \geq \alpha$.

Let $C \subseteq \mathbb{F}^n$ be a code and let $S \subseteq 2^{[n]}$ be a set of some subsets of $[n]$. Intuitively, think that S is a set of tests (local views) for C. Let $M \in \mathbb{F}^n$ be a word. Let $r_M(I)$ be the closest word of $C|_I$ to $M|_I$ (if there are more than one such codewords fix any of them arbitrarily). Intuitively, the subset I of M "thinks" that the symbols of $M|_I$ should be changed to $r_M(I)$, or equivalently, one can assume that $r_M(I)$ is the decoding of $M|_I$ to the closest codeword of $C|_I$. In this sense every subset of S has its own "opinion".

We say that a point $p \in [n]$ is almost fixed if for all $I_1, I_2 \in S$ such that $p \in I_1 \cap I_2$ we have $r_M(I_1)|_p = r_M(I_2)|_p$, but p is contained in some I such that $r_M(I)|_p \neq M|_p$. We let $ToFix(M) = \{p \in [n] \mid p \text{ is almost fixed}\}$. Intuitively, a point p is almost fixed if all subsets $I \in S$ containing p agree on this point, but "think" that its value in M ($M|_p$) should be changed (to $r_M(I)|_p$).

We say that a point $p \in [n]$ is inconsistent with respect to M if there exist two subsets $I_1, I_2 \in S$ such that $p \in I_1 \cap I_2$ and $r_M(I_1)|_p \neq r_M(I_2)|_p$. We let $Incon(M)$ to denote the set of all inconsistent points with respect to M. A subset $I \in S$ is called α-bad with respect to M if $|I \cap Incon(M)| \geq \alpha \cdot |I|$. We say that p is contained in an α-bad subset if there exists an α-bad subset I such that $p \in I$.

Definition 4.3 (Nice codes)**.** Let $\alpha, \beta \leq 1$ and $m \in \mathbb{N}^+$. The code C is called m-bounded with respect to S if for every $i \in [n]$ it holds that $1 \leq |\{I \in S \mid i \in I\}| \leq m$. The code C is called α-regulated with respect to S if for every $M \in \mathbb{F}^{\text{coord}(C)}$ it holds that every inconsistent point with respect to M is contained in some α-bad subset of S. We say that C is β-recoverable with respect to S if for every $M \in \mathbb{F}^{\text{coord}(C)}$ and $\hat{S} \subseteq S$ such that $\frac{|\bigcup_{I \in \hat{S}} I|}{|\text{coord}(C)|} < \beta$ and $Incon(M) \cup ToFix(M) \subseteq (\bigcup_{I \in \hat{S}} I)$, then $\delta(M, C) \leq \frac{|\bigcup_{I \in \hat{S}} I|}{|\text{coord}(C)|}$.

We say that C is (m, α, β)-nice with respect to S if C is m-bounded, α-regulated and β-recoverable with respect to S.

Now we state Theorem 4.4 which claims that a nice code is robustly testable.

Theorem 4.4 (Nice codes are robustly testable)**.** *Let $C \subseteq \mathbb{F}^n$ be a code. Assume that $\mathcal{D}$ is a uniform distribution over a set S and all subsets $I \in S$ have the same size $\text{sz} = |\text{I}|$ (for some sz). If C is (m, α, β)-nice with respect to S then $\rho^{\mathcal{D}}(C) \geq \frac{\alpha\beta}{m^2}$.*

The proof of Theorem 4.4 is postponed to Section 4.1.

Remark 4.5. Theorem 4.4 considers only the uniform distribution $\mathcal{D}$ over the local views (I) of the same size. It should be possible to extend the theorem to any distribution over the local views and allow the local views of different size.

4.1 Proof of Theorem 4.4

Let $M \in \mathbb{F}^n$ be an input word. We prove that $\rho^{\mathcal{D}}(M) \geq \frac{\alpha\beta}{m^2} \cdot \delta(M, C)$.

The overview of the proof All points in M can be classified into the three categories: almost fixed points ($ToFix(M)$), inconsistent points ($Incon(M)$) and good points (where p is a good point if for all subsets $I \in S$ such that $p \in I$ it holds that $M|_p = r_M(I)|_p$).

Intuitively, we prove that whenever $\rho^{\mathcal{D}}(M)$ is small then $\delta(M, C)$ is small. In Proposition 4.6 we show that $\rho^{\mathcal{D}}(M)$ is proportional to the number of inconsistent points and to the number of almost fixed points. That means that if $\rho^{\mathcal{D}}(M)$ is small then the number of inconsistent points and almost fixed points is small. By assumption, every inconsistent point of M is contained in some (at least one) α-bad subset of S. By Proposition 4.7 it follows that the number of points covered by all α-bad subsets is small. We use this observation to argue that M is close to C.

Now we state and prove Proposition 4.6.

Proposition 4.6. *It holds that* $\rho^{\mathcal{D}}(M) \geq \frac{1}{m} \cdot \left(\frac{|Incon(M)|}{n} + \frac{|ToFix(M)|}{n} \right)$.

Proof. By definition of robustness (Definition 4.2) we have

$$\rho^{\mathcal{D}}(M) = \mathop{\mathbf{E}}_{I \sim \mathcal{D}}[\delta(M|_I, r_M(I))] = \frac{\mathbf{E}_{I \sim \mathcal{D}}[\Delta(M|_I, r_M(I))]}{\text{sz}} = \frac{\sum_\tau \Delta(M|_I, r(I))}{|S| \cdot \text{sz}},$$

where we used the fact that there are $|S|$ subsets and every subset contains sz points. Since $nm \geq \text{sz} \cdot |\text{S}|$ we conclude that

$$\rho^{\mathcal{D}}(M) \geq \frac{1}{m} \cdot \frac{\sum_I \Delta(M|_I, r(I))}{n}.$$

It is sufficient to prove that $\sum_{I \in S} \Delta(M|_I, r_M(I)) \geq |Incon(M)| + |ToFix(M)|$.

Note that by definition the set $Incon(M)$ is disjoint to the set $ToFix(M)$. On the other hand, for every subset $I \in S$ and every point $p \in I$, if $(M|_I)|_p \neq r_M(I)|_p$ it holds that $p \in Incon(M) \cup ToFix(M)$. Thus for every subset $I \in S$ we have

$$\Delta(M|_I, r_M(I)) = |\{p \in I \mid p \in ToFix(M)\}| + |\{p \in I \mid p \in Incon(M)\}|.$$

Now, we recall that for every point $p \in Incon(M) \cup ToFix(M)$ there exists at least one subset $I \in S$ (of at most m spaces containing the point p) such that $r(I)|_p \neq M|_p$. We conclude that $\sum_{I \in S} \Delta(M|_I, r_M(I)) \geq |Incon(M)| + |ToFix(M)|$. □

We let $\hat{S} \subseteq S$ be the set of α-bad subsets with respect to M.

Proposition 4.7. *It holds that* $\left|\bigcup_{I \in \hat{S}} I\right| \leq \frac{|Incon(M)|m}{\alpha}$.

Proof. Every α-bad subset I contains at least $\alpha|I|$ elements. On the other hand, every element of $Incon(M)$ is contained in at most m subsets of $\hat{S}$. Thus $(|\bigcup_{I \in \hat{S}} I| \cdot \frac{\alpha}{m}) \leq |Incon(M)|$. □

We are ready to prove Theorem 4.4.

Proof of Theorem 4.4. First we argue that if $|\bigcup_{I \in \hat{S}} I| \geq \beta n$ then $\rho^{\mathcal{D}}(M) \geq \frac{\alpha\beta}{m^2}$ and we are done. Proposition 4.6 says that $\rho^{\mathcal{D}}(M) \geq \frac{1}{m} \cdot \frac{|Incon(M)|}{n}$. If $|\bigcup_{I \in \hat{S}} I| \geq \beta n$ then Proposition 4.7 implies that $\frac{|Incon(M)|m}{\alpha} \geq \beta n$. Thus $\rho^{\mathcal{D}}(M) \geq \frac{1}{m} \cdot \frac{|Incon(M)|}{n} \geq \frac{\alpha\beta}{m^2}$.

Otherwise $|\bigcup_{I \in \hat{S}} I| < \beta n$. First of all, we define the vector $M' \in \mathbb{F}^n$ which is identical to M on all points outside $ToFix(M)$, and for every point $p \in ToFix(M)$ we let $M'|_p = r_M(I)|_p$ for some subset $I \in S$ containing p (note that all subsets "want" the same value on p). I.e., we fix all points in $ToFix(M)$ to the "right" values. It holds that $\delta(M, M') \leq \frac{|ToFix(M)|}{n}$.

Notice that there are no almost fixed points in M', i.e., $ToFix(M') = \emptyset$. On the other hand, it can be easily verified that $Incon(M') = Incon(M)$. Thus every point $p \in Incon(M') \cup ToFix(M')$ is contained in some $I \in \hat{S}$. By assumption, the code C is (m, α, β)-nice with respect to S (Definition 4.3) and hence $\delta(M', C) \leq \frac{|\bigcup_{I \in \hat{S}} I|}{n} \leq \frac{|Incon(M)|m}{\alpha n}$, where the last inequality holds due to Proposition 4.7. Then it holds that

$$\delta(M, C) \leq \delta(M, M') + \delta(M', C) \leq \frac{|ToFix(M)|}{n} + \frac{|Incon(M)|m}{\alpha \cdot n} \leq$$

$$\leq \frac{m}{\alpha} \cdot \left(\frac{|Incon(M)|}{n} + \frac{|ToFix(M)|}{n}\right).$$

Thus Proposition 4.6 yields

$$\rho^{\mathcal{D}}(M) \geq \frac{1}{m} \cdot \left(\frac{|Incon(M)|}{n} + \frac{|ToFix(M)|}{n} \right) \geq \frac{\alpha}{m^2} \cdot \delta(M, C).$$

□

5 L-product of codes and their testing

In this section we define a new form of composition of error-correcting codes, called L-product of codes. We study the properties of these codes in Sections 5.1 and 5.2. Then, in Sections 5.3 and 5.5 we study the robust testing of L-products. Finally, in Section 5.5 we consider the testing of two-wise L-products.

Let $C \subseteq \mathbb{F}^n$ be a linear code and let $A(C) \subseteq [n]$ be its core (see Definition 2.1). We define $L^1(C) = C$ to be the 1-wise L-product.

2-wise L product. We let $L^2(C)$ to be the 2-wise L-product of C whose coordinates are $(A(C) \times [n]) \cup ([n] \times A(C))$, and for every $i \in A(C)$ it holds that $L^2(C)|_{(\{i\} \times [n])} \in C$ and $L^2(C)|_{([n] \times \{i\})} \in C$, i.e., every axis parallel line that crosses $A(C) \times A(C)$ (either $\{i\} \times [n]$ or $[n] \times \{i\}$) belongs to C. The core of the code $L^2(C)$ is defined to be $A(L^2(C)) = A(C) \times A(C)$.

Remark 5.1. Intuitively, think that $A(C) = [|A(C)|]$, i.e., it is a set of the first $|A(C)|$ coordinates in $[n]$. We chose to call this operation L-product since such two-wise product ($L^2(C)$) reminds the letter "L".

m-wise L product. This L product of codes can be easily generalized to m-wise L-product for every $m \geq 2$. For $j \in [m]$ and $l_1, l_2, \ldots, l_{j-1}, l_{j+1}, \ldots, l_m \in A(C)$ we define

$$(j, (l_1, l_2, \ldots, l_{j-1}, l_{j+1}, \ldots, l_m)) - \text{line} \equiv \{(l_1, l_2, \ldots, l_{j-1}, l_j, l_{j+1}, \ldots, l_m) \mid l_j \in [n]\}.$$

We let $L^m(C)$ to be the m-wise L-product of C whose coordinates are the union of all lines, i.e.,

$$\{(j, (l_1, l_2, \ldots, l_{j-1}, l_{j+1}, \ldots, l_m)) - \text{line} \mid j \in [m]\} =$$
$$\{(l_1, l_2, \ldots, l_m) \mid \exists j \in [m] : (l_j \in [n] \text{ and } \forall i \in [m] \setminus \{j\} : l_i \in A(C))\}$$

and for every $j \in [m]$ and $l_1, l_2, \ldots, l_m \in A(C)$, letting l' be the $(j, (l_1, l_2, \ldots, l_{j-1}, l_{j+1}, \ldots, l_m))-$line it holds that $L^m(C)|_{l'} \in C$.

We let $A^m(C) = \overbrace{A(C) \times A(C) \times \ldots \times A(C)}^{m}$. The core of $L^m(C)$, denoted by $A(L^m(C))$, is defined to be $A(L^m(C)) = A^m(C)$.

Definition 5.2 (Petals of the code)**.** Given a linear code $C \subseteq \mathbb{F}^n$ and its core $A(C)$, we say that $C|_{-A(C)}$ is a single petal of the code C. Note that the coordinate set for this petal is $[n] \setminus A(C)$.

Recall that for $m \geq 2$ the code $L^m(C)$ has the core $A^m(C)$. We say that the code $L^m(C)$ has m petals:

$$L^m(C)|_{A^{m-1}(C) \times ([n] \setminus A(C))}, L^m(C)|_{A^{m-2}(C) \times ([n] \setminus A(C)) \times A(C)}, L^m(C)|_{A^{m-3}(C) \times ([n] \setminus A(C)) \times A^2(C)},$$
$$\ldots\ldots, L^m(C)|_{A(C) \times ([n] \setminus A(C)) \times A^{m-2}(C)}, L^m(C)|_{([n] \setminus A(C)) \times A^{m-1}(C)}.$$

We note that if $A(C)$ is a core of C then every petal of $L^m(C)$ has the same dimension as $L^m(C)$. I.e., whenever we know the values of all bits in some petal in the given codeword, we know all bits in this codeword. That means on the one hand, a core of the code determines all bits of the code, and on the other hand, any petal can determine all bits of the code.

5.1 Properties of $L^m(C)$

We summarize the properties of $L^m(C)$ in the following claim.

Claim 5.3. *Assume that $C \subseteq \mathbb{F}^n$ is a linear code, where $A(C)$ is a γ-core of C. Then the following properties hold.*

1. *We have $\delta(C) \geq \gamma$.*
2. *It holds that $\delta(L^m(C)|_{A^m(C)}) \geq \gamma^m$, $\delta(L^m(C)|_{-A(L^m(C))}) \geq \gamma^m$, and $\delta(L^m(C)) \geq \gamma^m$. In particular, $\Delta(L^m(C)) \geq \gamma^m \cdot |\mathrm{coord}(L^m(C))|$.*
3. *It holds that $\dim(L^m(C)) = \dim(L^m(C)|_{A^m(C)}) = (\dim(C))^m = \dim(C|_{A(C)})^m$, where we used the fact that $\dim(C|_{A(C)}) = \dim(C)$.*
4. *It holds that $\mathrm{blocklength}(L^m(C)) = |\mathrm{coord}(L^m(C))| = |A(C)|^m + m \cdot |A(C)|^{m-1}(n - |A(C)|)$. Thus for $m \geq 2$ we have $|\mathrm{coord}(L^{m-1}(C))| \cdot |A(C)| \leq |\mathrm{coord}(L^m(C))| \leq |\mathrm{coord}(L^{m-1}(C))| \cdot |A(C)| \cdot 2$.*
5. *It holds that*

$$\mathrm{rate}(L^m(C)) = \frac{\dim(L^m(C))}{\mathrm{blocklength}(L^m(C))} \geq \frac{(\dim(C))^m}{m|A(C)|^{m-1}n} = \frac{(\mathrm{rate}(C|_{A(C)}))^{m-1}}{m} \cdot \mathrm{rate}(C),$$

   where the last equality holds since $\mathrm{rate}(C|_{A(C)}) = \frac{\dim(C|_{A(C)})}{|A(C)|} = \frac{\dim(C)}{|A(C)|}$.
6. *We have $L^4(C) = L^2(L^2(C))$.*

Remark 5.4. Assume $C \subseteq \mathbb{F}^n$ is a linear code, where $A(C)$ is a γ-core of C for some constant $\gamma > 0$ (see Definition 2.1).

Then $L^m(C)$ has the same dimension and the same distance promise to $C^{\otimes m}$, since $\dim(C^{\otimes m}) = \dim(L^m(C)) = (\dim(C))^m$, $\delta(C^{\otimes m}) = (\delta(C))^m \geq \gamma^m$ and $\delta(L^m(C)) \geq \gamma^m$.

We stress that $L^m(C)$ has much better rate than the m-wise tensor product of C if $\mathrm{rate}(C) = o(1)$ and $\mathrm{rate}(C|_{A(C)}) = \Omega(1)$. This is true since $\mathrm{rate}(C^{\otimes m}) = (\mathrm{rate}(C))^m$ but $\mathrm{rate}(L^m(C)) \geq \frac{(\mathrm{rate}(C|_{A(C)}))^{m-1}}{m} \cdot \mathrm{rate}(C)$. In particular, if $m \geq 2$ is a fixed constant, then $\mathrm{rate}(L^m(C))$ decreases only by a constant with respect to $\mathrm{rate}(C)$.

E.g., if $m \geq 2$ is a constant, $\mathrm{rate}(C) = \frac{1}{\log n}$ and $\mathrm{rate}(C|_{A(C)}) = \Omega(1)$ then $\mathrm{rate}(C \otimes C) = (\mathrm{rate}(C))^2 = \frac{1}{\log^2 n}$, while $\mathrm{rate}(L^2(C)) = \Omega(\frac{1}{\log n})$. This demonstrates a benefit of the L-products versus the tensor products.

5.2 L-products for different codes

In Section 5 we defined the L-product for a single code C. In this section we define the L-product for a number of linear codes $C_1, C_2, \ldots, C_m$ such that for $i \in [m]$ it holds that $C_i \subseteq \mathbb{F}^{n_i}$ has the core $A(C_i)$. This product will be denoted by $L(C_1, C_2, \ldots, C_m)$. The coordinate set of $L(C_1, C_2, \ldots, C_m)$ is defined by

$$\text{coord}(L(C_1, \ldots, C_m)) = \{(l_1, l_2, \ldots, l_m) \mid \exists j \in [m] : (l_j \in [n_j] \text{ and } \forall i \in [m] \setminus \{j\} : l_i \in A(C_i))\}.$$

Every line in $\text{coord}(L(C_1, \ldots, C_m))$ belongs to the appropriate code C_j, or more formally, it holds that for every $j \in [m]$ and $l_1 \in A(C_1), l_2 \in A(C_2), \ldots, l_m \in A(C_m)$, letting l' be the $(j, (l_1, l_2, \ldots, l_{j-1}, l_{j+1}, \ldots, l_m))$ − line it holds that $L^m(C)|_{l'} \in C_j$.

The core of $L(C_1, C_2, \ldots, C_m)$ is defined by $A(L(C_1, C_2, \ldots, C_m)) = A(C_1) \times A(C_2) \times \ldots \times A(C_m)$.

Claim 5.5. *Assume that for every $i \in [m]$ a linear code C_i has the core $A(C_i)$. Then*

$$\dim(L(C_1, \ldots, C_m)) = \dim(C_1) \cdot \dim(C_2) \cdot \ldots \cdot \dim(C_m).$$

Proof. For every $i \in [m]$ we know that $\dim((C_i)|_{A(C_i)}) = \dim(C_i)$. The claim follows since

$$\dim(L(C_1, \ldots, C_m)) = \dim(L(C_1, \ldots, C_m)|_{A(C_1) \times A(C_2) \times \ldots \times A(C_m)}) =$$

$$\dim(C_1|_{A(C_1)}) \cdot \dim(C_2|_{A(C_2)}) \cdot \ldots \cdot \dim(C_m|_{A(C_m)}) = \dim(C_1) \cdot \dim(C_2) \cdot \ldots \cdot \dim(C_m).$$

□

5.3 Robust Testing of L-products

In this section we consider the "L-space tester" (or briefly, the space tester), which is analogous to the "hyperplane tester" defined in the work of Ben-Sasson and Sudan [10] (see also [28, 32]). To do this let us define two auxiliary notations: points and L-spaces (or shortly, spaces).

Definition 5.6 (Points and L-spaces)**.** Let $C \subseteq \mathbb{F}^n$ be a linear code where we recall that $A(C)$ is a core for C.

A point with respect to the code $L^m(C)$ can be associated with an m-tuple $(i_1, i_2, ..., i_m)$ such that $(i_1, i_2, ..., i_m) \in \text{coord}(L^m(C))$. For $b \in [m]$ and $i \in A(C)$ we say that τ is a (b, i)-space if

$$\tau = \{(i_1, i_2, \ldots, i_m) \mid i_b = i \text{ and } (i_1, i_2, \ldots, i_m) \in \text{coord}(L^m(C))\}.$$

If τ is a (b, i)-space then $L^m(C)|_\tau$ denotes the projection of the code $L^m(C)$ to the space τ.

Remark 5.7. Let τ is a (b, i)-space. Note that the projection of any codeword $M \in L^i(C)$ to the space τ produces the word $M|_\tau \in L^{i-1}(C)$.

Definition 5.8 (L-space Tester)**.** Let $m \geq 3$. Let $M \in \mathbb{F}^{\text{coord}(L^m(C))}$ be an input word and think of testing whether $M \in L^m(C)$. The L-space tester $\mathcal{D}$ picks (non-adaptively) a random $b \in [m]$ and random $i \in A(C)$, and returns $M|_{(b,i)}$, i.e., the word M projected on the selected (b, i)-space. It is not hard to prove that if $M \in L^m(C)$ then $M|_{(b,i)} \in L^{m-1}(C)$.

Notice that the L-space tester $\mathcal{D}$ is a uniform distribution over the spaces. Now we state our Lemma 5.9.

Lemma 5.9. *Let $C \subseteq \mathbb{F}^n$ be a linear code such that $A(C)$ is a γ-core. Let S be the set of spaces with respect to $L^m(C)$ and $\mathcal{D}$ be the space tester for $L^m(C)$. Then $L^m(C)$ is $(m, \frac{\gamma^{m-1}}{4}, \frac{\gamma}{2})$-nice with respect to S and by Theorem 4.4 it holds that $\rho^{\mathcal{D}}(L^m(C)) \geq \frac{\gamma^m}{8m^2}$.*

The proof of Lemma 5.9 appears in Section 5.4. Let us prove Theorem 5.10 using Lemma 5.9.

Theorem 5.10 (Main Technical Theorem)**.** *Let C be a code such that $A(C)$ is a γ-core. If $L^2(C)$ is a (q, ϵ)-strong LTC then $L^m(C)$ is a $\left(q, \epsilon \cdot \left(\frac{\gamma^m}{8m^2}\right)^{m-2}\right)$-strong LTC.*

We are ready to prove Theorem 5.10.

Proof of Theorem 5.10. Let $\mathcal{D}_m$, $\mathcal{D}_{m-1}$,$\mathcal{D}_{m-2}$, ..., and $\mathcal{D}_3$ be the space testers for $L^m(C)$, $L^{m-1}(C)$, $L^{m-2}(C)$, ..., and $L^3(C)$, respectively. Assume that $M \in \mathbb{F}^{\text{coord}(L^m(C))}$ be an input word.

Lemma 5.9 says that the tester $\mathcal{D}_i$ (for $3 \leq i \leq m$) on an input word M_i outputs the local view (which is a candidate to be in $L^{i-1}(C)$) such that the expected relative distance of this local view from the code $L^{i-1}(C)$ is at least $\frac{\gamma^m}{8 \cdot m^2} \cdot \delta(M_i, L^i(C))$. Recall that $L^2(C)$ is a (q, ϵ)-strong LTC and let $\mathcal{D}_2$ be its tester.

Let us describe the "composed" tester, where the composition of testers is done similarly to [10, 32]. We sample the tester $\mathcal{D}_m$ on the word M and obtain a local view $M|_I$ for some subset of coordinates $I \subseteq \text{coord}(L^m(C))$. Then we sample the tester $\mathcal{D}_{m-1}$ on $M|_I$ and obtain a smaller local view $(M|_I)|_{I'}$ for some subset of coordinates $I' \subseteq I$, etc. Finally, we obtain a local view of $\mathcal{D}_3$, called X, which is a candidate word to be in $L^2(C)$ and invoke the tester $\mathcal{D}$ of $L^2(C)$ on the word X. The composed tester accepts/rejects as the tester of $L^2(C)$, which accepts X with probability 1 if $X \in L^2(C)$, and otherwise rejects X with probability at least $\epsilon \cdot \delta(X, L^2(C))$.

So, the rejection probability of the composed tester on the word M is at least

$$\delta(M, L^m(C)) \cdot \frac{\gamma^m}{8 \cdot m^2} \cdot \frac{\gamma^{m-1}}{8 \cdot (m-1)^2} \cdot \ldots \cdot \frac{\gamma^3}{8 \cdot 3^2} \cdot \epsilon \geq \epsilon \cdot \left(\frac{\gamma^m}{8m^2}\right)^{m-2} \cdot \delta(M, L^m(C)).$$

Clearly, the composed tester has query complexity q and always accepts all codewords of $L^m(C)$. Hence $L^m(C)$ is a $\left(q, \epsilon \cdot \left(\frac{\gamma^m}{8m^2}\right)^{m-2}\right)$-strong LTC. □

5.4 Proof of Lemma 5.9

First we define the basic concepts (some of the terms defined similarly to [10, 32]) and recall Definition 5.8. For simplicity we recommend to the reader to think about the case where $m = 3$.

For $b \in [m]$ and $i_1, i_2, \ldots, i_m \in A(C)$ we say that l is a $(b, (i_1, i_2, \ldots, i_{b-1}, i_{b+1}, \ldots, i_m))$-line if

$$l = \{(i_1, i_2, \ldots, i_{b-1}, i, i_{b+1}, \ldots, i_m) \mid \quad \text{where } i \in [n]\}.$$

Note that $(b, (i_1, i_2, \ldots, i_{b-1}, i_{b+1}, \ldots, i_m))$-line is parallel to the b-th axis and crosses the core of the code $L^m(C)$, i.e., $A(L^m(C))$. A line l contains a point p if $p \in l$. Note that a $(b, (i_1, i_2, \ldots, i_{b-1}, i_{b+1}, \ldots, i_m))$-line contains a point $p = (j_1, i_2, \ldots, j_m)$ if for all $k \in [m] \setminus \{b\}$ we have $i_k = j_k$.

A space τ contains the point p if $p \in \tau$ and contains the line l if it contains all points of the line. We say that two (different) spaces τ_1 and τ_2 intersect if $\tau_1 \cap \tau_2 \neq \emptyset$.

Given a word $M \in \mathbb{F}^{\text{coord}(L^m(C))}$, $b \in [m]$ and $i \in A(C)$ we let $M|_{(b,i)}$ be a restriction of M to the (b,i)-space. We say that $M|_{(b,i)}$ is a (b,i)-space of M. Similarly, for the point p let $M|_p$ be a restriction of M to the point p and for the line l we let $M|_l$ be a restriction of M to the line l.

We recall the definitions made in Section 4. For every space τ of M let $r_M(\tau)$, or simply $r(\tau)$, to denote the closest codeword of $L^{m-1}(C)$ to $M|_\tau$ (if there are more than one such codewords fix any of them arbitrarily). We say that spaces τ_1 and τ_2 disagree on the point p if $p \in \tau_1 \cap \tau_2$ and $r(\tau_1)|_p \neq r(\tau_2)|_p$. In this case, the point p is called inconsistent with respect to M. The set $Incon(M)$ denotes all inconsistent points with respect to M. For the point p we say that the point is almost fixed if all spaces agree on this point but $r(\tau)|_p \neq M|_p$. We let $ToFix(M)$ be a set of almost fixed points with respect to M.

5.4.1 The rest of the proof

Let us state Propositions 5.11 and 5.12. Their proofs are postponed to Sections 5.4.2 and 5.4.3, respectively.

Proposition 5.11. *It holds that C is $\frac{\gamma^{m-1}}{4}$-regulated with respect to S.*

Proposition 5.12. *It holds that C is $\frac{\gamma}{2}$-recoverable with respect to S.*

We are ready to prove Lemma 5.9.

Proof of Lemma 5.9. We know that the distribution $\mathcal{D}$ is uniform over S and all spaces of S are of the same size. Note also that $L^m(C)$ is m-bounded with respect to S since for every point $p \in \text{coord}(L^m(C))$ it holds that $1 \leq |\{\tau \in S \mid p \in \tau\}| \leq m$.

Proposition 5.11 proves that C is $\frac{\gamma^{m-1}}{4}$-regulated with respect to S. Proposition 5.12 shows that C is $\frac{\gamma}{2}$-recoverable with respect to S. We conclude that C is $(m, \frac{\gamma^{m-1}}{4}, \frac{\gamma}{2})$-nice with respect to S. □

5.4.2 Proof of Proposition 5.11

We need the following simple claim about the intersection of spaces.

Claim 5.13 (space intersections)**.** *Let τ_1 be the (b_1, i_1)-space and τ_2 be the (b_2, i_2)-space.*

- *If $b_1 = b_2$ and $i_1 = i_2$ then $\tau_1 = \tau_2$.*
- *If $b_1 = b_2$ and $i_1 \neq i_2$ then $\tau_1 \cap \tau_2 = \emptyset$.*
- *If $b_1 \neq b_2$ then $\tau_1 \cap \tau_2 \neq \emptyset$, and if $M \in L^m(C)$ then $M|_{\tau_1 \cap \tau_2} \in L^{m-2}(C)$.*

Let $M \in \mathbb{F}^{\text{coord}(L^m(C))}$ be an input word and $p = (i_1, i_2, \ldots, i_m)$ be an inconsistent point with respect to M. We recall that a space $\tau \in S$ is $\frac{\gamma^{m-1}}{4}$-bad with respect to M if

$$|Incon(M) \cap \tau| \geq \frac{\gamma^{m-1}}{4} \cdot |\tau|.$$

We prove that p is contained in some $\frac{\gamma^{m-1}}{4}$-bad space.

Proof of Proposition 5.11. Since p is inconsistent there exist two spaces τ_1 and τ_2 such that $p \in \tau_1 \cap \tau_2$ and $r(\tau_1)|_p \neq r(\tau_2)|_p$. Due to the symmetry of $L^m(C)$, we assume without loss of generality that $\tau_1 = (1, i_1)$ and $\tau_2 = (2, i_2)$ for some $i_1, i_2 \in A(C)$. We will prove that either τ_1 is a bad space or τ_2 is a bad space.

Consider the intersection of τ_1 and τ_2, i.e.,

$$reg = \tau_1 \cap \tau_2 = \{(i_1, i_2, j_3, j_4, \ldots, j_m) \mid (i_1, i_2, j_3, j_4, \ldots, j_m) \in \text{coord}(L^m(C))\}.$$

Note that $p \in reg$. We are going to define a line l which will be parallel to some axis $h \neq 1, 2$ such that $p \in l$. Recall that $m \geq 3$. If for every $j \in [m] \setminus \{1, 2\}$ it holds that $i_j \in A(C)$ (recall that $p = (i_1, i_2, \ldots, i_m)$) then fix $h = 3$, otherwise fix $h \in [m] \setminus \{1, 2\}$ such that $i_h \notin A(C)$ (there is only one such h since $p \in \text{coord}(L^m(C))$).

Again, due to the symmetry we assume without loss of generality that $h = 3$. Now, let l be a line, which is parallel to the axis h and contains the point p, i.e.,

$$l = (h, (i_1, i_2, \ldots, i_{h-1}, i_{h+1}, \ldots, i_m)) = (3, (i_1, i_2, i_4, \ldots, i_m)),$$

since we assumed that $h = 3$. Note that $l \in \tau_1 \cap \tau_2$.

The spaces τ_1 and τ_2 disagree on the line l, i.e., $r(\tau_1)|_l \neq r(\tau_2)|_l$, because they disagree on the point p contained in the line l. But $r(\tau_1)|_l, r(\tau_2)|_l \in C$ by definition. Now we recall that $A(C)$ is a core for C and thus $(r(\tau_1)|_l)|_{A(C)} \neq (r(\tau_2)|_l)|_{A(C)} \in C|_{A(C)}$. Hence $\Delta((r(\tau_1)|_l)|_{A(C)}, (r(\tau_2)|_l)|_{A(C)}) \geq \Delta(C|_{A(C)})$.

Since the coordinates of the line l are naturally associated with the coordinates of C, we use $l|_{A(C)}$ to denote the projection of the line to the coordinate subset associated with $A(C)$.

Let $ErrPoints = \left\{p \in l|_{A(C)} \mid \tau_1 \text{ and } \tau_2 \text{ disagree on } p\right\}$ and

$$ErrSpaces = \{(3, i) - \text{space} \mid i \in A(C), \exists p \in (ErrPoints \cap (3, i) - \text{space})\}.$$

Note that $|ErrPoints| \geq \Delta(C|_{A(C)}) \geq \gamma|A(C)|$ and $|ErrSpaces| \geq \Delta(C|_{A(C)}) \geq \gamma|A(C)|$.

We claim that for every $\tau \in ErrSpaces$ we have that either τ disagrees with τ_1 on some point $p \in ErrPoints$ or with τ_2 on some point $p \in ErrPoints$. Hence at least one of τ_1, τ_2 disagrees with at least $\frac{1}{2} \cdot |ErrSpaces| \geq \frac{1}{2} \cdot \Delta(C|_{A(C)})$ spaces from $ErrSpaces$. Without loss of generality assume that τ_1 disagrees with at least $\frac{1}{2} \cdot \Delta(C|_{A(C)})$ spaces from $ErrSpaces$.

Let $ErrSpaces_{\tau_1} = \{\tau \in ErrSpaces \mid \tau \text{ disagrees with } \tau_1\}$. Recall that $|ErrSpaces_{\tau_1}| \geq \frac{1}{2} \cdot \Delta(C|_{A(C)})$. All spaces from $ErrSpaces$ are non-intersecting and thus all spaces from $ErrSpaces_{\tau_1}$ are non-intersecting (see Claim 5.13). Every space $\tau \in ErrSpaces_{\tau_1}$ disagrees with the space τ_1 on some point and hence disagree on at least $\Delta(L^{m-2}(C))$ points in their intersection region $(\tau \cap \tau_1)$ since $r(\tau)|_{\tau \cap \tau_1} \neq r(\tau_1)|_{\tau \cap \tau_1} \in L^{m-2}(C)$. To see this note that $L^m(C)|_{\tau \cap \tau_1}$ is a code $L^m(C)$ restricted to the points in $\tau \cap \tau_1$, and thus $\delta(L^m(C)|_{\tau \cap \tau_1}) = \delta(L^{m-2}(C))$.

Let $total = \{p = (j_1, j_2, \ldots, j_m) \mid \exists \tau \in ErrSpaces_{\tau_1} \text{ s.t. } p \in \tau \cap \tau_1, r(\tau)|_p \neq r(\tau_1)|_p\}$ be the set of all "disagreement" points for τ_1. We have $|total| \geq \Delta(L^{m-2}(C)) \cdot (\frac{1}{2} \cdot \Delta(C|_{A(C)}))$ since every intersection region (as above) contains at least $\Delta(L^{m-2}(C))$ inconsistency points and there are at least $(\frac{1}{2} \cdot \Delta(C|_{A(C)}))$ such regions. We stress that we do not count any inconsistency point more than once, since the spaces in $ErrSpaces_{\tau_1}$ are non-intersecting.

Hence the space τ_1 contains at least $\frac{1}{2} \cdot \Delta(C|_{A(C)}) \cdot \Delta(L^{m-2}(C)) \geq \frac{1}{4} \cdot \gamma^{m-1} \cdot |\tau_1|$ inconsistent points, where we used that by Claim 5.3 we have $\Delta(C|_{A(C)}) \cdot \Delta(L^{m-2}(C)) \geq \gamma \cdot |A(C)| \cdot \gamma^{m-2} \cdot |\text{coord}(L^{m-2}(C))| \geq \gamma^{m-1} \cdot |\text{coord}(L^{m-1}(C))|/2$ and $|\tau_1| = |\text{coord}(L^{m-1}(C))|$.

We conclude that the point $p \in \tau_1$, where τ_1 is $\frac{\gamma^{m-1}}{4}$-bad space. □

5.4.3 Proof of Proposition 5.12

Let $\hat{S} \subseteq S$ such that $\left|\bigcup_{\tau\in\hat{S}} \tau\right| < (\gamma/2) \cdot |\mathrm{coord}(L^m(C))|$. Assume that $Incon(M) \cup ToFix(M) \subseteq \bigcup_{\tau\in\hat{S}} \tau$ belongs to some space in $\hat{S}$. We prove that $\delta(M, L^m(C)) \leq \frac{\left|\bigcup_{\tau\in\hat{S}} \tau\right|}{|\mathrm{coord}(L^m(C))|}$.

Proof of Proposition 5.12. For $b \in [m]$ we let

$$S_b = \{\tau = (b, j) \mid j \in A(C), \tau \in S\} \quad \text{and} \quad S'_b = \{j \mid (b, j) - \text{space} \in S_b\}.$$

For every space $\tau \in S$ it holds that $|\tau| = |\mathrm{coord}(L^{m-1}(C))|$ and $|\mathrm{coord}(L^{m-1}(C))| \cdot |A(C)| \leq |\mathrm{coord}(L^m(C))|$. Note that for every $b \in [m]$ the set S_b contains non-intersecting spaces. Hence the number of points covered by spaces in S_b is $\left|\bigcup_{\tau\in S_b} \tau\right| = |S_b| \cdot |\mathrm{coord}(L^{m-1}(C))|$. On the other hand,

$$\left|\bigcup_{\tau\in S_b} \tau\right| \leq \left|\bigcup_{\tau\in\hat{S}} \tau\right| < (\gamma/2) \cdot |\mathrm{coord}(L^m(C))| \leq \gamma|A(C)| \cdot |\mathrm{coord}(L^{m-1}(C))|.$$

Thus for every $b \in [m]$ we have $|S'_b| = |S_b| < \gamma|A(C)| \leq \Delta(C|_{A(C)})$. Thus for every $b \in [m]$ we have $\dim(C) = \dim(C|_{-S'_b}) = \dim((C|_{A}(C))|_{-S'_b})$.

It holds that $L^m(C)|_{-\hat{S}} = L(C|_{-S'_1}, C|_{-S'_2}, \ldots, C|_{-S'_m})$ and $\dim(L(C|_{-S'_1}, C|_{-S'_2}, \ldots, C|_{-S'_m})) = \dim(L(\overbrace{C, C, \ldots, C}^{m})) = \dim(L^m(C))$ (see Claim 5.5). Notice that $M|_{-\hat{S}} \in L^m(C)|_{-\hat{S}}$ since all inconsistent and almost fixed points were projected out.

We claim that every codeword c' of $L^m(C)|_{-\hat{S}}$ can be extended to a unique codeword c of $L^m(C)$. To see this note that the projection of C to $C|_{-S'_b}$ is bijective for every $b \in [m]$. It is surjective because it is a projection, and it is injective because $\dim(C) = \dim(C|_{-S'_b})$. So, the projection of $L^m(C)$ to $L^m(C)|_{-\hat{S}}$ is bijection, because both codes are of dimension $\dim(L(C|_{-S'_1}, C|_{-S'_2}, \ldots, C|_{-S'_m})) = \dim(L^m(C))$. Thus every word in $L^m(C)|_{-\hat{S}}$ has a unique preimage in $L^m(C)$. This observation implies that every codeword of $L^m(C)|_{-\hat{S}}$ can be uniquely extended to the codeword of $L^m(C)$ by appending $\left|\bigcup_{\tau\in\hat{S}} \tau\right|$ symbols. We conclude that $\delta(M, L^m(C)) \leq \frac{\left|\bigcup_{\tau\in\hat{S}} \tau\right|}{|\mathrm{coord}(L^m(C))|}$. □

5.5 $L^2(C)$ is testable but is not robustly testable

Let $C \subseteq \mathbb{F}^n$ be a linear code with γ-core $A(C)$. Consider $L^2(C)$ and recall that $\mathrm{coord}(L^2(C)) = (A(C) \times [n]) \cup ([n] \times A(C))$. We also recall that $M \in \mathbb{F}^{\mathrm{coord}(L^2(C))}$ is a codeword of $L^2(C)$ if and only if for every $i \in A(C)$ it holds that $M|_{\{i\}\times[n]} \in C$ and $M|_{[n]\times\{i\}} \in C$.

We define the line tester for the code $L^2(C)$, which is an analogue to the row/column tester for the two-wise tensor product [10, 12, 11, 19, 31].

Definition 5.14 (Line Tester)**.** Assume that $M \in \mathbb{F}^{\mathrm{coord}(L^2(C))}$ is an input word.

- Toss a coin.
- If "head" - pick random $i \in A(C)$ and accept iff $M|_{\{i\}\times[n]} \in C$.
- Otherwise - pick random $i \in A(C)$ and accept iff $M|_{[n]\times\{i\}} \in C$.

Using this line tester we prove the following Proposition.

Proposition 5.15 ($L^2(C)$ is testable)**.** *Let $C \subseteq \mathbb{F}^n$ be a linear code with γ-core $A(C)$. Then $L^2(C)$ is $(n, \frac{1}{2})$-strong LTC.*

Proof. Let $\mathcal{D}$ be the line tester for the code $L^2(C)$. Clearly, the query complexity of the tester is n. It also easy to see that the tester accepts all codewords $M \in L^2(C)$ with probability 1.

We turn to analyze its soundness parameter. Let $M \in \mathbb{F}^{\mathrm{coord}(L^2(C))}$ be an input word.

For $i \in A(C)$ we say that a line $\{i\} \times [n]$ is erroneous if $M|_{\{i\}\times[n]} \notin C$, and similarly, we say that a line $[n] \times \{i\}$ is erroneous if $M|_{[n]\times\{i\}} \notin C$.

The number of all lines is $2|A(C)|$ since for every $i \in A(C)$ there are two lines: $\{i\} \times [n]$ and $[n] \times \{i\}$. Let Err be the number of erroneous line. Then the rejection probability of the tester is $\dfrac{Err}{2|A(C)|}$.

On the other hand, $\delta(M, L^2(C)) \leq \dfrac{Err}{|A(C)|}$. This is true since all erroneous lines can be removed from the word M, and then can be reconstructed back. This statement is very similar to the corresponding statement in the 2-wise tensor products [11].

Thus the rejection probability is at least $\frac{1}{2} \cdot \delta(M, L^2(C))$. □

However, this line tester (Definition 5.14) does not provide a robust testing.

Proposition 5.16 ($L^2(C)$ is not robustly testable)**.** *For arbitrary large $n > 0$, there exists a linear code $C \subseteq \mathbb{F}_2^n$ with γ-core $A(C)$ such that letting $\mathcal{D}$ be its line tester we have $\rho^{\mathcal{D}}(L^2(C)) = o(1)$.*

Proof. We recall the result of [16], which was based on the work of [31] (see also [23]). In [16] the authors constructed a code $R \subset \mathbb{F}_2^h$ (for arbitrary large $h > 0$) such that $\delta(R) \geq \gamma$ for some constant $\gamma > 0$ and a word $M \in \mathbb{F}_2^{h\times h}$ such that every row and every column of M is $o(1)$-close to R but $\delta(M, R \otimes R) \geq \Omega(1)$.

Let $n = 2h$ and $C \subseteq \mathbb{F}_2^n$ such that $C|_{[h]} = R$ and $C|_{[n]\setminus[h]} = R$. I.e., $C = (R, R)$. We fix $A(C) = [h]$ and note that it is a γ-core of C since $\delta(C|_{A(C)}) \geq \gamma$ and $\delta(C|_{-A(C)}) \geq \gamma$.

Now, let $X \in \mathbb{F}_2^{\mathrm{coord}(L^2(C))}$ be a word such that for every $i \in A(C)$ we have

$$X|_{\{i\}\times[h]} = X|_{\{i\}\times([n]\setminus[h])} = M|_{\{i\}\times[h]}$$

and

$$X|_{[h]\times\{i\}} = X|_{([n]\setminus[h])\times\{i\}} = M|_{[h]\times\{i\}}.$$

It can be verified that for every $i \in A(C)$ we have $\delta(X|_{\{i\}\times[n]}, C) = o(1)$ and $\delta(X|_{[n]\times\{i\}}, C) = o(1)$. But $\delta(X, L^2(C)) \geq \dfrac{\delta\left(X|_{A(C)\times A(C)}, L^2(C)|_{A(C)\times A(C)}\right)}{3} = \dfrac{\delta(M, R \otimes R)}{3} = \Omega(1)$. Hence $\rho^{\mathcal{D}}(L^2(C)) = o(1)$. □

6 Robust Testing of Tensor Products

In this section we reprove the main result of [32]. Let $C \subseteq \mathbb{F}^n$ be a linear code. We consider the code $C^{\otimes m} \subseteq \mathbb{F}^{n^m}$ and note that $\mathrm{coord}(C^{\otimes m}) = [n]^m$.

We briefly recall the basic notations from [32]. A point p is denoted as $(i_1, i_2, \ldots, i_m)$ such that $i_j \in [n]$. We say that τ is a (b, i)-hyperplane if

$$\tau = \{(i_1, i_2, \ldots, i_m) \mid i_b = i \text{ and for all } j \in [m] \setminus \{b\} \text{ we have } i_j \in [n]\}.$$

In the rest of the paper, such hyperplanes τ will be called $(m-1)$-dimensional hyperplanes, since all points of such hyperplanes are determined using m dimensions $(i_1, i_2, \ldots, i_m)$, while one of them is fixed $(i_b = i)$ and the other are to be chosen.

Given a word $M \in \mathbb{F}^{n^m}$ and hyperplane τ we let $M|_\tau$ be a restriction of M to the hyperplane τ. Similarly, for the point p let $M|_p$ be a restriction of M to the point p. Let S be a set of hyperplanes with respect to $C^{\otimes m}$.

Definition 6.1 (Hyperplane Tester)**.** Let $m \geq 3$. Let $M \in \mathbb{F}^{n^m}$ be an input word and think of testing whether $M \in C^{\otimes m}$. The hyperplane tester $\mathcal{D}$ picks (non-adaptively) a random $b \in [m]$ and random $i \in [n]$, and returns (b, i)-hyperplane (the corresponding local view is $M|_{(b,i)}$). Notice that if $M \in C^{\otimes m}$ then $M|_{(b,i)} \in C^{\otimes(m-1)}$.

Now we state Theorem 6.2 and postpone its proof to Section 6.1.

Theorem 6.2 (Robust Testing of Tensor Products)**.** *Let $C \subseteq \mathbb{F}^n$ be a linear code and $m \geq 3$. Let $\mathcal{D}$ be the hyperplane tester for $C^{\otimes m}$. Then $C^{\otimes m}$ is $\left(m, \left(\frac{(\delta(C))^{m-1}}{2}\right), \delta(C)\right)$-nice with respect to S and by Theorem 4.4 it follows that $\rho^{\mathcal{D}}(C^{\otimes m}) \geq \frac{(\delta(C))^m}{2m^2}$.*

6.1 Proof of Theorem 6.2

First we recall some notations. Then we state and prove Propositions 6.3 and 6.4. Finally, we prove Theorem 6.2.

For $b \in [m]$ and $i \in [n]$ we say that l is a $(b, (i_1, i_2, \ldots, i_{b-1}, i_{b+1}, \ldots, i_m))$-line if

$$l = \{(i_1, i_2, \ldots, i_{b-1}, i, i_{b+1}, \ldots, i_m) \mid \quad \text{where } i \in [n]\}.$$

Note that $(b, (i_1, i_2, \ldots, i_{b-1}, i_{b+1}, \ldots, i_m))$-line is parallel to the b-th axis. For the line l we let $M|_l$ be a restriction of M to the line l.

Recall that given $M \in \mathbb{F}^{n^m}$, a hyperplane τ is $\left(\frac{(\delta(C))^{m-1}}{2}\right)$-bad if it contains at least $\left(\frac{(\delta(C))^{m-1}}{2}\right)$ inconsistent points with respect to M.

Proposition 6.3. *Let p be an inconsistent point with respect to M. Then p is contained in some $\left(\frac{(\delta(C))^{m-1}}{2}\right)$-bad hyperplane.*

Proof. We know that there are (at least) two hyperplanes that disagree on the point p. Assume without loss of generality (symmetry) that the hyperplanes $\tau_1 = (1, i_1)$ and $\tau_2 = (2, i_2)$ disagree on the point p.

Consider the intersection of τ_1 and τ_2, i.e., $reg = \tau_1 \cap \tau_2 = \{(i_1, i_2, j_3, j_4, \ldots, j_m) \mid j_k \in [n]\}$. Note that $p \in reg$. Let l be a line, which is parallel to the third axis and contains the point p (recall that $m \geq 3$). Then the hyperplanes τ_1 and τ_2 disagree on this line (since they disagree on the point p contained in the line l), i.e., $r(\tau_1)|_l \neq r(\tau_2)|_l$. But $r(\tau_1)|_l, r(\tau_2)|_l \in C$ by definition.

This implies that $\Delta(r(\tau_1)|_l, r(\tau_2)|_l) \geq \delta(C) \cdot n$, i.e., for at least $\delta(C) \cdot n$ points $p \in l$ it holds that $r(\tau_1)|_p \neq r(\tau_2)|_p$.

Let $BadPoints = \{p \in l \mid \tau_1 \text{ and } \tau_2 \text{ disagree on } p\}$. Note that $|BadPoints| \geq \delta(C) \cdot n$. Let

$$BadPlanes = \{(3,i) - \text{hyperplane} \mid i \in [n], \exists p \in BadPoints \text{ s.t. } p \in (3,i) - \text{hyperplane}\}.$$

Note that $|BadPlains| \geq \delta(C) \cdot n$.

We claim that for every $\tau \in BadPlanes$ we have that either τ disagrees with τ_1 on some point $p \in BadPoints$ or with τ_2 on some point $p \in BadPoints$. Hence at least one of τ_1, τ_2 disagrees with at least $\frac{1}{2} \cdot |BadPlanes| \geq \frac{1}{2} \cdot \delta(C)n$ hyperplanes from $BadPlanes$. Without loss of generality assume that τ_1 disagrees with at least $\frac{1}{2} \cdot \delta(C) \cdot n$ hyperplanes from $BadPlanes$.

Let $BadPlanes_{\tau_1} = \{\tau \in BadPlanes \mid \tau \text{ disagrees with } \tau_1\}$. All hyperplanes from $BadPlanes$ are non-intersecting and thus all hyperplanes from $BadPlanes_{\tau_1}$ are non-intersecting. Every hyperplane $\tau \in BadPlanes_{\tau_1}$ disagrees with the hyperplane τ_1 on some point and hence disagree on at least $(\delta(C)n)^{m-2}$ points in their intersection region $(\tau \cap \tau_1)$ since $r(\tau)|_{\tau \cap \tau_1} \neq r(\tau_1)|_{\tau \cap \tau_1} \in C^{\otimes(m-2)}$.

Let $total = \{p = (i_1, j_2, \ldots, j_m) \mid \exists \tau \in BadPlanes_{\tau_1} \text{ s.t. } p \in \tau \cap \tau_1, r(\tau)|_p \neq r(\tau_1)|_p\}$. We have $|total| \geq (\delta(C)n)^{m-2} \cdot \frac{\delta(C) \cdot n}{2} = \frac{(\delta(C) \cdot n)^{m-1}}{2}$ since every intersection region (as above) contains at least $(\delta(C)n)^{m-2}$ inconsistency points and there are at least $\frac{1}{2} \cdot \delta(C) \cdot n$ such regions. We stress that we do not count any inconsistency point more than once, since the hyperplanes in $BadPlanes_{\tau_1}$ are non-intersecting.

Hence the hyperplane τ_1 disagree with other hyperplanes in at least $\frac{(\delta(C) \cdot n)^{m-1}}{2}$ points (on the hyperplane τ_1). We conclude that τ_1 is a $\left(\frac{(\delta(C))^{m-1}}{2}\right)$-bad hyperplane and $p \in \tau_1$. □

Proposition 6.4. *Let $M \in \mathbb{F}^{n^m}$. Assume $\hat{S} \subseteq S$ is a subset of hyperplanes such that $Incon(M) \cup ToFix(M) \subseteq \left(\bigcup_{\tau \in \hat{S}} \tau\right)$ and $\frac{\left|\bigcup_{\tau \in \hat{S}} \tau\right|}{n^m} < \delta(C)$. Then $\delta(M, C^{\otimes m}) \leq \frac{\bigcup_{\tau \in \hat{S}} \tau}{n^m}$.*

Proof. For $b \in [m]$ let $S_b = \left\{i \mid (b,i) - \text{hyperplane} \in \hat{S}\right\}$ and note that $S = S_1 \times S_2 \times \ldots \times S_m$. Let $C' = C|_{S_1} \otimes C|_{S_2} \otimes \ldots \otimes C|_{S_m}$ and note that for every $i \in [n]$ we have $|S_i| > n - \delta(C) \cdot n$.

First we argue that every codeword c' of $C^{\otimes m}|_{-\hat{S}}$ can be extended to a unique codeword c of $C^{\otimes m}$. To see this note that the projection of C to $C|_{S_i}$ is bijective. It is surjective because it is a projection, and it is injective because $|S_i| > n - \Delta(C)$. So, the projection of $C^{\otimes m}$ to C' is bijection, because both codes are of dimension $(\dim(C))^m$. Thus, every word in $C^{\otimes m}|_{-\hat{S}}$ has a unique preimage in $C^{\otimes m}$. This argument was used by Ben-Sasson and Sudan in [10, Proposition 3.1]. We turn to prove Proposition 6.4. We know that $M|_{-\hat{S}} \in C^{\otimes m}|_{-\hat{S}}$ since all inconsistent and almost fixed points were projected out. Thus only changing the symbols in $M|_{\hat{S}}$ we can obtain a codeword of $C^{\otimes m}$. We conclude that $\delta(M, C^{\otimes m}) \leq \frac{\bigcup_{\tau \in \hat{S}} \tau}{n^m}$. □

We are ready to prove Theorem 6.2.

Proof of Theorem 6.2. We know that all hyperplanes of the code $C^{\otimes m}$ are of the equal size and that the hyperplane tester (Definition 6.1) has a uniform distribution over the hyperplanes. We know also that $C^{\otimes m}$ is m-bounded with respect to S.

Proposition 6.3 shows that $C^{\otimes m}$ is $\left(\frac{(\delta(C))^{m-1}}{2}\right)$-regulated with respect to S. Proposition 6.4 implies that $C^{\otimes m}$ is $\delta(C)$-recoverable with respect to S. We recall Definition 4.3 and conclude that $C^{\otimes m}$ is $\left(m, \left(\frac{(\delta(C))^{m-1}}{2}\right), \delta(C)\right)$-nice. □

7 Robust Testing of Star products

We start this section by providing an auxiliary Definition 7.1 and Claims 7.2 and 7.3 that will be highly useful in our studying of star products (Section 7.1) and robust testing of star products (Section 7.2). Then we proceed to define the star products and to study their basic properties in Section 7.1. Finally, in Section 7.2 we prove that the star products are robustly testable with respect to the core robustness (Definition 7.1).

Now we present one of the cental concepts in this paper.

Definition 7.1 (Core robustness)**.** Assume that $\mathbf{D}$ is a tester (i.e., a distribution) for the code $C \subseteq \mathbb{F}^n$ with a core $A(C)$. Assume that for every subset $I \subseteq [n]$ such that $\mathbf{D}(I) > 0$ it holds that $C|_I$ is a code with a core $A(C|_I)$. We let $\rho^{\mathbf{D}}_{A(C)}(w) = \mathop{\mathbf{E}}_{I\sim\mathbf{D}}\left[\delta_{A(C|_I)}(w|_I, C|_I)\right]$ be the expected core oriented relative local distance of input w. We say that the tester $\mathbf{D}$ for the code C has core robustness $\rho^{\mathbf{D}}_{A(C)}(C)$ if for every $w \in \mathbb{F}^n$ it holds that $\rho^{\mathbf{D}}_{A(C)}(w) \geq \rho^{\mathbf{D}}_{A(C)}(C) \cdot \delta_{A(C)}(w, C)$.

It turns out that a combination of "core robustness" with COLTCs is highly useful.

Claim 7.2. *Let C be a (q,ϵ)-COLTC and let $\mathcal{D}_C$ be its tester. Let $\hat{C} \subseteq \mathbb{F}^{\mathrm{coord}(\hat{C})}$ be a linear code with a core $A(\hat{C})$ and let $\mathcal{D}_{\hat{C}}$ be its tester. Assume that $\rho^{\mathcal{D}_{\hat{C}}}_{A(\hat{C})}(\hat{C}) \geq \alpha$ and for every local view $I \subseteq \mathrm{coord}(\hat{C})$ such that $\mathcal{D}_{\hat{C}}(I) > 0$ it holds that $\hat{C}|_I = C$. Then $\hat{C}$ is a $(q, \alpha\cdot\epsilon)$-COLTC.*

Proof. We consider a composed tester of $\mathcal{D}_{\hat{C}}$ and $\mathcal{D}_C$ that on the input word invokes the tester $\mathcal{D}_{\hat{C}}$ and then, on the obtained local view invokes the tester $\mathcal{D}_C$. The composed tester has query complexity at most q (as $\mathcal{D}_C$) and always accepts the codewords of $\hat{C}$. Moreover, the rejected probability of this composed tester on the word $w \in \mathbb{F}^{\mathrm{coord}(\hat{C})}$ is at least

$$\mathop{\mathbf{E}}_{I\sim\mathcal{D}_{\hat{C}}}\left[\epsilon\cdot\delta_{A(C)}(w|_I, \hat{C}|_I)\right] = \epsilon\cdot\mathop{\mathbf{E}}_{I\sim\mathcal{D}_{\hat{C}}}\left[\delta_{A(C)}(w|_I, \hat{C}|_I)\right] \geq \epsilon\cdot(\alpha\cdot\delta_{A(C)}(w,\hat{C})).$$

□

Following definitions of Section 4, given a code $C \subseteq \mathbb{F}^n$, $I \subseteq [n]$ and $w \in \mathbb{F}^n$ we set $r(w|_I)$ to be a closest word of $C|_I$ to $w|_I$ (if there are more than one such word, we fix one arbitrary). Similarly, we let $r(w)$ to be a closest word of C to w.

Claim 7.3. *Let C be a linear code with a γ-core $A(C)$ for some constant $\gamma > 0$. Let $w \in \mathbb{F}^{\mathrm{coord}(C)}$ be a word such that $\delta_{A(C)}(w, C) < \gamma/2$. Then $r(w)|_{A(C)} = r(w|_{A(C)})$.*

Proof. The fact that $\delta_{A(C)}(w, C) < \gamma/2$ implies that there exists a codeword $c \in C$ such that $\delta_{A(C)}(w, c) < \gamma/2$. That means $\delta(w|_{A(C)}, c|_{A(C)}) < \gamma/2$ and $\delta(w, c) < \gamma/2$, where $\delta(C|_{A(C)}) \geq \gamma$ and $\delta(C) \geq \gamma$, because $A(C)$ is a γ-core (see Definition 2.1). Thus $c|_{A(C)}$ is the closest codeword of $C|_{A(C)}$ to $w|_{A(C)}$, and c is the closest codeword of C to w. Hence, $r(w) = c$ and $r(w|_{A(C)}) = c|_{A(C)}$. Thus $r(w)|_{A(C)} = r(w|_{A(C)})$. □

7.1 Star Products

Now we provide a definition of star products. These products of codes are very similar to ones used in [27, Section 4], although there exist minor differences.

Definition 7.4 (Star Products - $C^{\star m}$)**.** Let $C \subseteq \mathbb{F}_2^n$ be a linear code and with a γ-core $A(C)$, where $\gamma > 0$ is a constant. Assume that for a linear code $R \subseteq \mathbb{F}_2^{n_R}$ we have $A(C) = [n_R]^2$ and $C|_{A(C)} = R^{\otimes 2}$. In this case we let $C^{\star 2} = C$ and $A(C^{\star 2}) = A(C) = [n_R]^2$.

Let residue_2 be an operator such that $\text{residue}_2(\text{A}(\text{C}^{\star 2})) = \text{coord}(\text{C}^{\star 2}) \backslash \text{A}(\text{C}^{\star 2})$. That means given a core coordinates of the code $C^{\star 2}$ it outputs the non-core coordinates of the code $C^{\star 2}$. This operator is important since the coordinates of the code might be enumerated in a different way. Note that $|\text{residue}_2(\text{A}(\text{C}^{\star 2}))| = \text{n} - |\text{A}(\text{C}^{\star 2})|$ and that for every $c \in C^{\star 2}$ we have $c = (c|_{A(C^{\star 2})}, c|_{\text{residue}_2(\text{A}(\text{C}^{\star 2}))})$.

We define $C^{\star m}$ and $\text{residue}_{\text{m}}(\cdot)$ by induction for $m \geq 2$ be an integer. Assume we defined $C^{\star(m-1)}$ and $\text{residue}_{\text{m}-1}(\cdot)$. We will use the term "hyperplane" to denote an $(m-1)$-dimensional hyperplane. Let $C^{\star m}$ to be a linear code over $\mathbb{F}_2$ and $\text{residue}_{\text{m}}(\cdot)$ be an operator such that

- $A(C^{\star m}) = [n_R]^m$ and $(C^{\star m})|_{A(C^{\star m})} = R^{\otimes m}$.
- $\text{coord}(C^{\star m}) \backslash A(C^{\star m})$ is defined to be a union of all coordinate subsets $\text{residue}_{\text{m}-1}(\tau)$ for all hyperplanes τ, where we assume that for every two hyperplanes $\tau_1 \neq \tau_2$ we have $\text{residue}_{\text{m}-1}(\tau_1) \cap \text{residue}_{\text{m}-1}(\tau_2) = \emptyset$, i.e., the coordinates of every subset $\text{residue}_{\text{m}-1}(\tau)$ are unique. [5] The projected code $C^{\star m}|_{-A(C^{\star m})}$ is defined to be a code such that for every hyperplane τ it holds that $C^{\star m}|_{\text{residue}_{(\text{m}-1)}(\tau)} = C^{\star(m-1)}|_{-A(C^{\star(m-1)})}$ and $C^{\star m}|_{(\tau \cup \text{residue}_{(\text{m}-1)}(\tau))} = C^{\star(m-1)}$.
- We define $\text{residue}_{\text{m}}(\text{A}(\text{C}^{\star \text{m}})) = \text{coord}(\text{C}^{\star \text{m}}) \setminus \text{A}(\text{C}^{\star \text{m}})$. So, for every $M \in C^{\star m}$ we have $M = \left(M|_{A(C^{\star m})}, M|_{\text{residue}_{\text{m}}(\text{A}(\text{C}^{\text{m}}))}\right)$.

Remark 7.5. We note that given $M \in C^{\star m}$ and an $(m-1)$-dimensional hyperplane τ we have $(M|_\tau, M|_{\text{residue}_{\text{m}-1}(\tau)}) \in C^{\star(m-1)}$. This is true since $C^{\star m}|_{(\tau \cup \text{residue}_{(\text{m}-1)}(\tau))} = C^{\star(m-1)}$.

The following claim shows the affect of the star product on the dimension, the distance and the rate of the underlying code.

Claim 7.6. *Let $C^{\star 2} \subseteq \mathbb{F}_2^n$ be a linear code and with a γ^2-core $A(C^{\star 2})$, where $\gamma > 0$ is a constant. Assume that for a linear code $R \subseteq \mathbb{F}_2^{n_R}$ we have $A(C^{\star 2}) = [n_R]^2$ and $C^{\star 2}|_{A(C^{\star 2})} = R^{\otimes 2}$. Then,*

- $\dim(C^{\star 4}) = (\dim(C^{\star 2}))^2$,
- $\text{blocklength}(C^{\star 4}) \leq 12|A(C^{\star 2})| \cdot \text{blocklength}(C^{\star 2})$,
- *if* $\frac{\dim(C^{\star 2})}{|A(C^{\star 2})|} \geq \beta > 0$ *then* $\text{rate}(C^{\star 4}) \geq \frac{\beta}{12} \cdot \text{rate}(C^{\star 2})$, *and*
- $A(C^{\star 4})$ *is a* (γ^4)*-core of* $C^{\star 4}$*. In particular,* $\delta(C^{\star 4}) \geq \gamma^4$ *and* $\delta(C^{\star 4}|_{A(C^{\star 4})}) \geq \gamma^4$.

Proof. We recall that $\text{blocklength}(C^{\star 2}) = n$ and $|A(C^{\star 2})| = n_R^2 \leq n$. We know that $\dim(C^{\star 4}) = (\dim(C^{\star 2}))^2$ since the code $C^{\star 4}$ projected on $A(C^{\star 4})$ is equal to $R^{\otimes 4}$, while $C^{\star 2}$ projected on $A(C^{\star 2})$ is equal to $R^{\otimes 2}$. Thus $\dim(C^{\star 4}) = (\dim(R))^4$ and $\dim(C^{\star 2}) = (\dim(R))^2$.

By definition we have

$$\text{blocklength}(C^{\star 4}) = 4 \cdot n_R \cdot \text{blocklength}(C^{\star 3}) = (4 \cdot n_R) \cdot (3 \cdot n_R) \cdot \text{blocklength}(C^{\star 2}) =$$

[5] The term hyperplane was defined in Section 6 and here it has exactly the same meaning with regards to $R^{\otimes m}$, i.e., $\tau = (b, i)$ for some $b \in [m]$, $i \in [n_R]$ means

$$\tau = \{(i_1, i_2, \ldots, i_m) \mid i_b = i \text{ and for all } j \in [m] \setminus \{b\} \text{ we have } i_j \in [n_R]\}.$$

$$= 12 \cdot n_R^2 \cdot \text{blocklength}(C^{\star 2}) = 12 \cdot |A(C^{\star 2})| \cdot \text{blocklength}(C^{\star 2}),$$

where we used the fact that there are $(4 \cdot n_R)$ 3-dimensional hyperplanes with regards to the code $C^{\star 4}$ and there are $(3 \cdot n_R)$ 2-dimensional hyperplanes with regards to the code $C^{\star 3}$. Thus,

$$\text{rate}(C^{\star 4}) = \frac{\dim(C^{\star 4})}{\text{blocklength}(C^{\star 4})} \geq \frac{(\dim(C^{\star 2}))^2}{12|A(C)| \cdot \text{blocklength}(C^{\star 2})} \geq \frac{\beta}{12} \cdot \text{rate}(C^{\star 2}).$$

We know that $A(C^{\star 4})$ is a core of $C^{\star 4}$ such that $C^{\star 4}|_{A(C^{\star 4})} = R^{\otimes 4}$. Similarly, $A(C^{\star 3})$ is a core of $C^{\star 3}$ such that $C^{\star 3}|_{A(C^{\star 3})} = R^{\otimes 3}$. Thus $\delta(C^{\star 4}|_{A(C^{\star 4})}) \geq \gamma^4$ and $\delta(C^{\star 3}|_{A(C^{\star 3})}) \geq \gamma^3$. Every residue_2 of a 2-dimensional hyperplane of $C^{\star 3}$ has relative distance γ^2, i.e., for a 2-dimensional hyperplane τ we have $\delta(C^{\star 3}|_{\text{residue}_2(\tau)}) \geq \gamma^2$ because $A(C^{\star 2})$ is a (γ^2)-core. Thus $\delta(C^{\star 3}|_{-A(C^{\star 3})}) \geq \gamma^3$ and similarly, $\delta(C^{\star 4}|_{-A(C^{\star 4})}) \geq \gamma^4$.

We conclude that $A(C^{\star 4})$ is a (γ^4)-core of $C^{\star 4}$. □

7.2 Star Products are Robustly Testable

We are ready to define a tester for the star products and prove Theorem 7.8.

Definition 7.7 (Tester for the Star Product)**.** The star-tester $\mathcal{D}_{\star m}$ for the code $C^{\star m}$, on the given word M acts as follows.

- picks a random $(m-1)$-dimensional hyperplane τ with regards to $C^{\star m}|_{A(C^{\star m})} = R^{\otimes m}$,
- outputs $M|_{(\tau \cup \text{residue}_{m-1}(\tau))}$.

We notice that this star-tester is m-bounded (Definition 4.3) with respect to $C^{\star m}$ since every coordinate/point of the code $C^{\star m}$ is contained in at most m local views and in at least 1 local view selected by the star-tester. Moreover, all local views are of the same size and the tester has a uniform distribution over these local views.

Now we state and prove Theorem 7.8 that shows that the star products are robustly testable with respect to "core robustness" (see Definition 7.1). Then we will conclude Corollary 7.9 showing that if $C^{\star 2}$ is a q-query COLTC, then $C^{\star 4}$ is a q-query COLTC.

Theorem 7.8. *Let C be a linear code with a γ^2-core $A(C)$ such that $C|_{A(C)} = R^{\otimes 2}$ for a linear code $R \subseteq \mathbb{F}_2^{n_R}$. Assume that $\mathcal{D}$ is the star-tester for the code $C^{\star m}$, where $m \geq 3$. Then,*

$$\rho^{\mathcal{D}}_{A(C^{\star m})}(C^{\star m}) \geq \frac{\gamma^m}{7 \cdot m^2}.$$

Proof. We know that $\delta(C|_{A(C)}) \geq \gamma^2$, $\delta(C|_{-A(C)}) \geq \gamma^2$ and $\delta(C) \geq \gamma^2$ (see Definition 2.1). Since $\delta(C|_{A(C)}) = (\delta(R))^2$ we know that $\delta(R) \geq \gamma$.

Let $M \in \mathbb{F}_2^{\text{coord}(C^{\star m})}$ be an input word and $\alpha = \rho^{\mathcal{D}}_{A(C^{\star m})}(M)$. If $\rho^{\mathcal{D}}_{A(C^{\star m})}(M) \geq \frac{\gamma^m}{7 \cdot m^2}$ we are done. Otherwise, assume that $\rho^{\mathcal{D}}_{A(C^{\star m})}(M) < \frac{\gamma^m}{7 \cdot m^2}$ for the rest of the proof.

In the rest of the proof, when we say "a hyperplane" the intention is "$(m-1)$-dimensional hyperplane". Notice that the local views selected by the tester can be denoted by $(\tau, \text{residue}_{m-1}(\tau))$ for a hyperplane τ selected at random. Recall that $\tau \subseteq A(C^{\star m})$. We have

$$\rho^{\mathcal{D}}_{A(C^{\star m})}(M) = \mathop{\mathbf{E}}_{(\tau \cup \text{residue}_{m-1}(\tau)) \sim \mathcal{D}} \left[\delta_{A(C^{\star (m-1)})}(M|_{(\tau \cup \text{residue}_{m-1}(\tau))}, C^{\star (m-1)}) \right] = \alpha.$$

Let us call the local view $(\tau \cup \text{residue}_{m-1}(\tau))$ *far* if $\delta_{A(C^{\star(m-1)})}(M_{(\tau\cup\text{residue}_{m-1}(\tau))}, C^{\star(m-1)}) \geq \frac{\gamma^{m-1}}{4}$, and otherwise it is *close*. This implies that the fraction of *far* local views is at most $\beta = \frac{4\alpha}{\gamma^{m-1}}$.

By arguments similar to Claim 7.6 we have $\delta(C^{\star m}|_{\tau\cup\text{residue}_{m-1}(\tau)}) = \delta(C^{\star(m-1)}) \geq \gamma^{m-1}$ and $\delta(C^{\star m}|_\tau) \geq \gamma^{m-1}$. Hence $\delta_{A(C^{\star(m-1)})}(C^{\star m}|_{\tau\cup\text{residue}_{m-1}(\tau)})$.

Now we recall Claim 7.3 saying that besides far local views $(\tau\cup\text{residue}_{m-1}(\tau))$, we have $r(M|_\tau) = r(M|_{(\tau\cup\text{residue}_{m-1}(\tau))})|_\tau$. This is a crucial fact, which shows that decoding of the "close" local view $M|_{(\tau\cup\text{residue}_{m-1}(\tau))}$ to the closest codeword of $C^{\star(m-1)}$ is equivalent to decoding of its "core" part $M|_\tau$ to the closest codeword of $R^{\otimes(m-1)}$ and then fixing the bits of its non-core part $M|_{(\text{residue}_{m-1}(\tau))}$ to correspond to the obtained "core" part of the local view. Recall that the core bits uniquely defines the non-core bits (see Definition 2.1).

We recall the concepts of inconsistent and almost fixed points from Section 4 with respect to our local views $((\tau\cup\text{residue}_{m-1}(\tau)))$ for hyperplanes τ and the word M: $Incon(M)$ and $ToFix(M)$. The intuition of this paragraph is similar to one that was used in the proof of Theorem 4.4 in Section 4.1. Let us fix all almost fixed point to their "right values" in M and to obtain M', i.e., if the coordinate $p \in (\tau\cup\text{residue}_{m-1}(\tau))$ is almost fixed, then $M'|_p = r(M|_{(\tau\cup\text{residue}_{m-1}(\tau))})|_p$. By Proposition 4.6, we have $\frac{|ToFix(M)|}{|\text{coord}(C^{\star m})|} \leq \alpha \cdot m$. Thus $\delta(M, M') \leq \alpha m$. Similarly, we have $\frac{|ToFix(M|_{A(C^{\star m})})|}{|A(C^{\star m})|} \leq \alpha \cdot m$ since α denotes the core robustness of the tester, which was defined with respect to the core oriented distance. Hence $\delta(M|_{A(C^{\star m})}, M'|_{A(C^{\star m})}) \leq \alpha m$. We conclude that $\delta_{A(C^{\star m})}(M, M') \leq \alpha m$. Now, it holds that M' has no almost fixed points but $Incon(M') = Incon(M)$.

Notice that after almost fixed points were fixed, the average distance of a typical local view on M' could become only smaller and, in particular, we have

$$\alpha \geq \mathop{\mathbf{E}}_{(\tau\cup\text{residue}_{m-1}(\tau))\sim\mathcal{D}}\Big[\delta_{A(C^{\star(m-1)})}(M'|_{\tau\cup\text{residue}_{m-1}(\tau)}, C^{\star(m-1)})\Big] \geq$$

$$\geq \mathop{\mathbf{E}}_{\tau}\Big[\delta(M'|_\tau, C^{\star(m-1)}|_{A(C^{\star(m-1)})})\Big] = \mathop{\mathbf{E}}_{\tau}\Big[\delta(M'|_\tau, R^{\otimes(m-1)})\Big].$$

In words, an average hyperplane of M' is α-close to being consistent.

We say that a hyperplane τ is bad with respect to M' if at least $\frac{\gamma^{(m-1)}}{2} \cdot n_R^{m-1}$ of its entries are inconsistent points, and recall that $|\tau| = n_R^{m-1}$. By Theorem 6.2, every inconsistent point is contained in at least one bad hyperplane. Moreover, every inconsistent point is covered by at most m bad hyperplanes (notice that all hyperplanes are subsets of $A(C^{\star m})$). We notice also that inconsistent points can be only in $A(C^{\star m})$ since every coordinate of $\text{coord}(C^{\star m}) \setminus A(C^{\star m})$ is covered by a single local view, which is a residue of the corresponding hyperplane. Recall that every hyperplane has its own (unique) residue.

Thus, recalling that $Incon(M') = Incon(M)$, we use $\big|Incon(M|_{A(C^{\star m})})\big|$ to denote that all inconsistent points of M (or M') are in $M|_{A(C^{\star m})}$, and so, the number of bad hyperplanes is at most

$$\frac{m \cdot \big|Incon(M|_{A(C^{\star m})})\big|}{(\gamma^{(m-1)}/2) \cdot n_R^{m-1}} = \frac{2m \cdot \big|Incon(M|_{A(C^{\star m})})\big|}{\gamma^{(m-1)} n_R^{m-1}}.$$

So, the fraction of bad hyperplanes is at most

$$\beta' = \frac{2m \cdot \big|Incon(M|_{A(C^{\star m})})\big|}{\gamma^{(m-1)} m \cdot n_R^m} = \frac{2\,\big|Incon(M|_{A(C^{\star m})})\big|}{\gamma^{(m-1)} n_R^m},$$

since the total number of hyperplanes is $m \cdot n_R$.

Proposition 4.6 implies that

$$\alpha = \rho^{\mathcal{D}}_{A(C^{\star m})}(M') \geq \frac{Incon(M|_{A(C^{\star m})})}{m \cdot n_R^m},$$

which means that $\left|Incon(M|_{A(C^{\star m})})\right| \leq \alpha(m \cdot n_R^m)$. Thus, $\beta' = \frac{2\alpha(m \cdot n_R^m)}{\gamma^{(m-1)} \cdot n_R^m} = \frac{2m\alpha}{\gamma^{(m-1)}}$.

Remove from M' all local views $(M'|_{(\tau \cup \text{residue}_{m-1}(\tau))})$ if the hyperplane τ is bad or the local view $(\tau \cup \text{residue}_{m-1}(\tau))$ is far. So we remove

$$\beta + \beta' \leq \frac{4\alpha}{\gamma^{m-1}} + \frac{2m\alpha}{\gamma^{(m-1)}} \leq \frac{6m\alpha}{\gamma^{m-1}} < \frac{\gamma}{m}$$

fraction of hyperplanes and local views, where we use the assumption that $\alpha < \frac{\gamma^m}{7m^2}$.

Since this fraction is less than $\frac{\gamma}{m}$, the fraction of points covered by the removed hyperplanes in $M'|_{A(C^{\star m})}$ is strictly less than $\gamma \leq \delta(R)$ and thus by Proposition 6.4, all removed local views can be restored back (modified) to form a codeword. This can be done first, as in Proposition 6.4, by restoring the removed hyperplanes in $M'|_{A(C^{\star m})}$, which is a candidate to be in $R^{\otimes m}$, and after that restoring the residues of the restored hyperplanes. Recall that every hyperplane has its unique residue, where different residues are not intersected and the bits of a residue is uniquely determined by the bits of its hyperplane. Call the resulting word X and note that $X \in C^{\star m}$. It holds that

$$\delta_{A(C^{\star m})}(M', X) = \max\left\{\delta(M'|_{A(C^{\star m})}, X|_{A(C^{\star m})}), \delta(M', X)\right\} \leq \beta + \beta' \leq \frac{6m\alpha}{\gamma^{m-1}}.$$

Remark 2.3 implies that $\delta_{A(C^{\star m})}(M, X) \leq \delta_{A(C^{\star m})}(M, M') + \delta_{A(C^{\star m})}(M', X)$. Since $X \in C^{\star m}$ we have

$$\delta_{A(C^{\star m})}(M, C^{\star m}) \leq \alpha m + \frac{6m\alpha}{\gamma^{m-1}} \leq \frac{7m\alpha}{\gamma^{m-1}}.$$

We conclude that

$$\alpha = \rho^{\mathcal{D}}_{A(C^{\star m})}(M) \geq \frac{\gamma^{m-1}}{7m} \cdot \delta_{A(C^{\star m})}(M, X) \geq \frac{\gamma^m}{7m^2} \cdot \delta_{A(C^{\star m})}(M, X).$$

□

We conclude the following corollary.

Corollary 7.9. *Assume that $C^{\star 2}$ is a (q, ϵ)-COLTC and $A(C^{\star 2})$ is a γ^2-core of $C^{\star 2}$. Assume that $C^{\star 2}|_{A(C^{\star 2})} = R^{\otimes 2}$ for some linear code $R \subseteq \mathbb{F}_2^{n_R}$. Then $C^{\star 4}$ is a $\left(q, \epsilon \cdot \left(\frac{\gamma^7}{10^4}\right)\right)$-COLTC.*

Proof. The proof of Corollary 7.9 follows from Theorem 7.8 and Claim 7.2.

Let $\mathcal{D}_3$ be the star-tester for the code $C^{\star 3}$ and let $\mathcal{D}_4$ be the star-tester for the code $C^{\star 4}$. Theorem 7.8 implies that

$$\rho^{\mathcal{D}}_{A(C^{\star 3})}(C^{\star 3}) \geq \frac{\gamma^3}{7 \cdot 3^2}$$

and

$$\rho^{\mathcal{D}}_{A(C^{\star 4})}(C^{\star 4}) \geq \frac{\gamma^4}{7 \cdot 4^2}.$$

Notice that the local views of the tester $\mathcal{D}_4$ on any input codeword $M \in C^{\star 4}$ belong to $C^{\star 3}$. Thus we can define the combined tester $\mathcal{D}_{comb}$ that on the input word M will invoke the tester $\mathcal{D}_4$, and then on the obtained local view will invoke the tester $\mathcal{D}_3$. The resulting local views produced by the combined tester are candidates to belong to $C^{\star 2}$.

Thus $\rho^{\mathcal{D}_{comb}}_{A(C^{\star 4})}(C^{\star 4}) \geq \left(\frac{\gamma^4}{7\cdot 4^2}\right) \cdot \left(\frac{\gamma^3}{7\cdot 3^2}\right) \geq \frac{\gamma^7}{10^4}$. Since $C^{\star 2}$ is a (q,ϵ)-COLTC, Claim 7.2 implies that $C^{\star 4}$ is a $\left(q, \epsilon \cdot \left(\frac{\gamma^7}{10^4}\right)\right)$-COLTC. □

8 Concatenation can preserve the query complexity

In this section we develop our main tools that will be used in Section 9 to show how to implement the distance amplification procedure to preserve the query complexity of the underlying COLTC.

Let us start from the definition of locally decodable codes (LDCs). Then we will define a concatenation of error-correcting codes in Definition 8.3. Finally, we state the main theorem of this section (Theorem 8.4), whose proof is postponed to Section 8.2 and relies on the new kind of testing presented in Section 8.1.

Definition 8.1 (LDCs)**.** Let $C \subseteq \mathbb{F}_2^n$ be a linear code of dimension $k = \dim(C)$. Let $E_C : \mathbb{F}_2^k \to \mathbb{F}_2^n$ be the encoding function of C, i.e., $C = \left\{E_C(m) \mid m \in \mathbb{F}_2^k\right\}$. Then C is a (q,ϵ,α)-LDC (where $\epsilon < \frac{1}{|\mathbb{F}_2|} = \frac{1}{2}$) if there exists a randomized decoder $\mathbf{D}$ that reads at most q queries and the following condition holds.

- For all $m \in \mathbb{F}_2^k$ and $i \in [k]$ letting $c = E_C(m)$ we have that $\mathbf{Pr}[\mathbf{D}^c[i] = m_i] = 1$.
- For all $m \in \mathbb{F}_2^k$, $i \in [k]$ and $\hat{c} \in \mathbb{F}_2^n$ such that $\delta(E_C(m), \hat{c}) \leq \alpha$ we have $\mathbf{Pr}\left[\mathbf{D}^{\hat{c}}[i] \neq m_i\right] \leq \epsilon$.

Note that Definition 8.1 is a bit non-standard since it requires that given any codeword the decoder retrieves correctly any message symbol with probability 1. However, it seems that any linear LDC should satisfy this requirement, and in particular, the famous Hadamard code is known to be $(2, 2\alpha, \alpha)$-query LDC for $0 \leq \alpha < \frac{1}{4}$ (under Definition 8.1).

Notice also that Definition 8.1 implies that $\alpha < \delta(C)/2$ since otherwise there exists a word w which is α-close to two different codewords ($E_C(m_1) \neq E_C(m_2)$) and the decoder will not be able to retrieve with high probability some message entries.

The following fact is known due to [15, 4, 3, 25].

Fact 8.2 (Hadamard code)**.** *Let $H \subseteq \mathbb{F}_2^{2^t}$ be the Hadamard code. The generator matrix of H is a binary matrix $G_H \in \mathbb{F}_2^{2^t \times t}$ whose rows are all distinct vectors in $\mathbb{F}_2^t$, so $H = \left\{G_H \cdot x \mid x \in F_2^k\right\}$.*

Then H is a $(3, \frac{1}{2})$-strong LTC and $(2, 2\alpha, \alpha)$-LDC for every $0 \leq \alpha \leq \frac{1}{4}$.

We recall that the Hadamard code $H \subseteq \mathbb{F}_2^{2^t}$ contains the message bits as a part of its encoding. We let $\mathrm{msg}(H) \subseteq [2^t]$ to denote the message bits coordinates in the codewords of H, i.e., if for some $m \in \mathbb{F}_2^t$ we have $G_H \cdot m = h$ then $h|_{\mathrm{msg}(H)} = m$.

Let $m \in \mathbb{F}_2^t$ and $h \in H$. We say that h fits the message m if $h|_{\mathrm{msg}(H)} = m$. We say that a codeword h is α-close to fit the message m if there exists $h' \in H$ such that h' fits the message m and $\delta(h, h') \leq \alpha$.

Now let us define the repeated codes and the concatenation of codes (the definitions are similar to the analogues definitions in [27]).

Definition 8.3 (Repeated Hadamard code and Concatenation)**.** We define the l-repeated Hadamard code $H^{[l]} : \mathbb{F}_2^{l \cdot t} \to \mathbb{F}_2^{l \cdot 2^t}$ as the code that results from partitioning the message $x \in \mathbb{F}_2^{l \cdot t}$ to l blocks of length t, and encoding each of them by H. I.e., the code $H^{[l]}$ encodes every message $x \in \mathbb{F}_2^{l \cdot t}$ by

$$H^{[l]}(x) = H(x_1, x_2, \ldots, x_t)H(x_{t+1}, x_{t+2}, \ldots, x_{2t}) \ldots H(x_{(l-1)\cdot t+1}, x_{(l-1)\cdot t+2}, \ldots, x_{l \cdot t}).$$

Let $C : \mathbb{F}_2^k \to \mathbb{F}_2^{tn}$ be a linear code such that $A(C) = [|A(C)|]$ and $t \mid |A(C)|$. Recall that $H^{[n]} : \mathbb{F}_2^{n \cdot t} \to \mathbb{F}_2^{n \cdot 2^t}$ is the n-repeated Hadamard code. Then $H^{[n]} \circ C : \mathbb{F}_2^k \to \mathbb{F}_2^{n \cdot 2^t}$ is a linear code that encodes the message $x \in \mathbb{F}_2^k$ by encoding it with C (to the string in $\mathbb{F}_2^{tn}$) and then encoding the result with $H^{[n]}$ (to the string in $\mathbb{F}_2^{n \cdot 2^t}$).

We set $A(H^{[n]} \circ C) = [(2^t) \cdot \frac{|A(C)|}{t}]$ to be the core of $H^{[n]} \circ C$. Notice that $C|_{A(C)}$ is composed from $\frac{|A(C)|}{t}$ consequent substrings of length t, and each such substring is going to be encoded (in $H^{[n]} \circ C$) by the Hadamard code H to the string of length 2^t. Thus, $(H^{[n]} \circ C)|_{A(H^{[n]} \circ C)}$ is a sequence of $\frac{|A(C)|}{t}$ Hadamard codes H of length 2^t.

In the following statement (Theorem 8.4) we argue that the concatenation of a strong LTC with the Hadamard code preserve the query complexity. It can be verified that instead of the Hadamard code we could use any other linear code which is 3-query strong LTC, 2-query LDC and has the message bits as a part of its encoding. However, we note that any 2-query linear LDC has exponential blocklength [14, 20, 22], so the Hadamard code is optimal in this sense.

Theorem 8.4. *Let $C : \mathbb{F}_2^k \to \mathbb{F}_2^{tn}$ be a (q, ϵ)-COLTC for $q \geq 3$, with a core $A(C) = [|A(C)|]$ such that $t \mid |A(C)|$. Let $H : \mathbb{F}_2^t \to \mathbb{F}_2^{2^t}$ be the Hadamard code and $H^{[n]} : \mathbb{F}_2^{n \cdot t} \to \mathbb{F}_2^{n \cdot 2^t}$ be the n-repeated Hadamard code. Let $C' = H^{[n]} \circ C : \mathbb{F}_2^k \to \mathbb{F}_2^{n \cdot 2^t}$ be the concatenation of $H^{[n]}$ and C.*

Then C' is a $(q, \frac{\epsilon}{66t})$-COLTC and $A(C') = A(H^{[n]} \circ C)$.

The proof of Theorem 8.4 appears in Section 8.2 and this proof relies on the new concept (Definition 8.5) suggested in Section 8.1 and Proposition 8.6.

8.1 Local Testing to fit the message

The standard notion of the local testability for a code C assumes a tester that accepts or rejects a given word w. If this tester rejects with small probability then the given word w is close to the code C. That means some of the entries of w are "wrong" but we don't know exactly which one, and the only thing we know is that the number of such wrong entries is small.

In this section, we initiate the studying a special kind of local testing. Assume that a code contains all message bits as a part of its encoding (e.g., the Hadamard code). Now, assume that the message bits are much more "important" than non-message bits. So, we want to distinguish between two cases: (i) it is possible to modify a small number of non-message bits to obtain a codeword and (ii) to obtain a codeword one should change at least one message bit or a large fraction of non-message bits. In this case, if a tester rejects a given word w with a small probability we conclude that only a small fraction of non-message bits in w should be modified to get a codeword, and in particular, no message bits should be modified.

Fortunately, it can be shown that the Hadamard code has such type of testing. Now we provide a definition for the described type of LTC, which we call strong LTCM for strong Local Testing to

check a Codeword as well as a Message[6].

Definition 8.5 (Test to fit the message)**.** Let $C \subseteq \mathbb{F}^n$ be a linear code that contains its message entries as a part of its encoding. Assume without loss of generality that the first $\dim(C)$ entries of C are the message entries. Let $\delta_{msg}(w, C) = \min\left\{\delta(w, c) \mid c \in C, c|_{[\dim(C)]} = w|_{[\dim(C)]}\right\}$ be the relative distance between w and the closest codeword of C with the same message entries as in w.

We say that a code $C \subseteq \mathbb{F}^n$ is (q, ϵ)-strong LTCM if there exists a tester that makes at most q queries to the input word w. It always accept the codewords of C and rejects every word $w \notin C$ with probability at least $\epsilon \cdot \delta_{msg}(w, C)$. We call this tester the LTCM-tester of C.

Note that $\delta_{msg}(w, C) \geq \delta(w, C)$ and hence any (q, ϵ)-strong LTCM is also a (q, ϵ)-strong LTC. Now we prove that the Hadamard code is a strong LTCM. It is not hard to prove because the Hadamard code contains the message as a part of its encoding, it is a 3-query strong LTC and a 2-query LDC (as stated in Fact 8.2).

Proposition 8.6. *Let $C \subseteq \mathbb{F}_2^{2^k-1}$ be the Hadamard code, where $k = \dim(C)$. Then C is a $(3, \frac{1}{16k})$-strong LTCM.*

Proof. Let $\mathcal{T}_C$ be the $(3, \frac{1}{2})$-strong tester for the code C and $\mathbf{D}$ be the decoder for the code C, guaranteed by Fact 8.2. First we define a tester $\mathcal{T}_{LTCM}$ for the code C. Let $w \in \mathbb{F}_2^{2^k-1}$ be an input word and assume without loss of generality that the first k entries of C are the message entries.

- pick random $r \in \{1, 2\}$
- If $r = 1$ then invoke $\mathcal{T}_C$ on w and accept iff this tester accepted.
- Otherwise ($r = 2$)
  - pick a random coordinate $i \in [k]$
  - invoke $\mathbf{D}^w[i]$ and accept iff $\mathbf{D}^w[i] = w|_i$, i.e., the invocation of $\mathbf{D}$ returned the bit w_i. (Note that $w|_i$ is queried here.)

Clearly, this tester makes at most 3 queries and accepts all codewords of C. Assume that $w \notin C$. If $\delta(w, C) \geq \frac{1}{4}$, then the word w is rejected by $\mathcal{T}_{LTCM}$ with probability at least

$$\frac{1}{2} \cdot \frac{1}{2} \cdot \delta(w, C) \geq \frac{1}{16} \geq \frac{1}{16} \cdot \delta_{msg}(w, C),$$

since the tester $\mathcal{T}_C$ is invoked with probability $\frac{1}{2}$, and due to the Fact 8.2 it rejects w with probability at least $\frac{1}{2} \cdot \delta(w, C)$.

Otherwise $\delta(w, C) < \frac{1}{4}$. Thus there exists a codeword $c \in C$ such that $\delta(w, c) = \delta(w, C) < \frac{1}{4}$. If $c|_{[k]} = w|_{[k]}$ then $\delta_{msg}(w, C) = \delta(w, c) = \delta(w, C) < \frac{1}{4}$. In this case the word w is rejected with probability at least

$$\frac{1}{2} \cdot \frac{1}{2} \cdot \delta(w, C) \geq \frac{1}{4} \cdot \delta_{msg}(w, C).$$

[6]We stress that the definition of an LTCM code is not a special case of the definition of a COLTC with a core equal to the message bits subset. This is true since a small rejection probability for an LTCM-tester means that **no message bit** inside the input word should be changed to obtain a codeword, while a small rejection probability for an COLTC-tester means that only **a small fraction of message bits** should be changed to obtain a codeword.

Otherwise, there is a message entry $i \in [k]$ such that $c|_i \neq w|_i$ and recall that $\delta(w,c) = \delta(w,C) < \frac{1}{4}$. With probability $\frac{1}{2} \cdot \frac{1}{k}$ the decoder $\mathbf{D}$ is invoked on i ($\mathbf{D}^w[i]$) and with probability at least $1 - 2 \cdot \frac{1}{4} = \frac{1}{2}$ this decoder outputs $c|_i$ (see Fact 8.2), which means that w is rejected. In this case, the rejection probability is at least

$$\frac{1}{2} \cdot \frac{1}{k} \cdot \frac{1}{2} = \frac{1}{4k} \geq \frac{1}{4k} \cdot \delta_{msg}(w,C).$$

So, in any case the word w is rejected by our tester $\mathcal{T}_{LTCM}$ with probability at least

$$\min\left\{\frac{1}{16} \cdot \delta_{msg}(w,C),\ \ \frac{1}{4} \cdot \delta_{msg}(w,C),\ \ \frac{1}{4k} \cdot \delta_{msg}(w,C)\right\} \geq \frac{1}{16k} \cdot \delta_{msg}(w,C).$$

□

8.2 Proof of Theorem 8.4

Let $\mathcal{T}_C$ be a COLTC-tester for C. Let $\mathcal{T}_H$ be an LTCM-tester (see Definition 8.5) for the Hadamard code H as promised by Proposition 8.6). Recall that $A(C) = [|A(C)|]$ and $t \mid |A(C)|$. We also recall that $A(C')$ is a core of C'.

We let $\text{msg} \subseteq [n \cdot 2^t]$ to denote the message coordinates with regards to the encoding by $H^{[n]}$, i.e., for every $w \in H^{[n]}$ we have $w|_{\text{msg}} = m$, where $H^{[n]}(m) = w$. Note that if $c' \in C'$ then $c'|_{\text{msg}} \in C$. We notice that $|\text{msg}| = n \cdot t$ and $A(C) \subseteq \text{msg}$.

For $j \in [n \cdot t]$ let $\text{block}_\text{j} = \left\{(\text{j}-1) \cdot 2^\text{t} + 1, (\text{j}-1) \cdot 2^\text{t} + 2, \ldots, (\text{j}-1) \cdot 2^\text{t} + 2^\text{t}\right\}$. Recall that the code C' is a sequence of the Hadamard encodings (with blocklength 2^t). We call them the Hadamard blocks of C'. I.e., for every $j \in [n \cdot t]$ it holds that for every $c' \in C'$: $c'|_{\text{block}_\text{j}} \in H$.

The tester for C'. We define the tester $\mathcal{T}_{C'}$ for C'. Assume $w' \in \mathbb{F}_2^{n \cdot 2^t}$ is an input word.

- pick random $r \in \{1,2\}$
  - If $r = 1$
    - invoke the tester $\mathcal{T}_C$ on $w'|_{\text{msg}}$ and accept iff $\mathcal{T}_C$ accepted.
  - Otherwise ($r = 2$),
    - pick random $r' \in \{1,2\}$
      - If $r' = 1$ then pick random $j \in A(C)$
      - Otherwise ($r' = 2$), pick random $j \in [n \cdot t]$
    - invoke the LTCM-tester $\mathcal{T}_H$ on $w'|_{\text{block}_\text{j}}$ and accept iff $\mathcal{T}_H$ accepted.

The query complexity of $\mathcal{T}_{C'}$. The tester $\mathcal{T}_{C'}$ makes at most q queries to the input word w' since the tester $\mathcal{T}_C$ makes at most q queries, and the LTCM-tester $\mathcal{T}_H$ makes at most 3 queries. Recall that $q \geq 3$.

All codewords are accepted with probability 1. Assume that $w' \in C'$. We know that $w'|_{\text{msg}} \in C$ and hence $\mathcal{T}_C$ accepts $w'|_{\text{msg}}$ with probability 1. It holds that for every $j \in [n]$ we have $w'|_{\text{block}_\text{j}} \in H$ and thus $\mathcal{T}_H$ always accepts the word $w'|_{\text{block}_\text{j}}$. Hence the tester $\mathcal{T}_{C'}$ accepts w' with probability 1.

8.2.1 The analysis of the soundness parameter of $\mathcal{T}_{C'}$

Assume that $w' \notin C'$ and let $\beta = \mathbf{Pr}[\mathcal{T}_{C'}[w'] = \mathsf{reject}]$. We argue that $\delta(w', C') \leq \frac{66t}{\epsilon} \cdot \beta$. This proves that C' is a $(q, \frac{\epsilon}{66t})$-COLTC.

It holds that $\delta_{A(C)}(w'|_{\mathrm{msg}}, C)$ is small. We argue that $\delta_{A(C)}(w'|_{\mathrm{msg}}, C) \leq \frac{1}{t} \cdot \left(\frac{2t}{\epsilon} \cdot \beta\right)$. Assume not, but then

$$\mathbf{Pr}\left[\mathcal{T}_{C'}[w'] = \mathsf{reject}\right] \geq \frac{1}{2} \cdot \left(\mathbf{Pr}\left[\mathcal{T}_{C}[w'|_{\mathrm{msg}}] = \mathsf{reject}\right]\right) \geq \frac{1}{2} \cdot \epsilon \cdot \delta_{A(C)}(w'|_{\mathrm{msg}}, C) >$$

$$\frac{\epsilon}{2} \cdot \left(\frac{1}{t} \cdot \left(\frac{2t}{\epsilon} \cdot \beta\right)\right) \geq \beta.$$

Contradiction. Thus $\delta_{A(C)}(w'|_{\mathrm{msg}}, C) \leq \frac{1}{t} \cdot \left(\frac{2t}{\epsilon} \cdot \beta\right)$, i.e., at most $\frac{1}{t} \cdot \left(\frac{2t}{\epsilon} \cdot \beta\right)$-fraction of bits in $w'|_{\mathrm{msg}}$ should be changed, such that this modifies at most $\frac{1}{t} \cdot \left(\frac{2t}{\epsilon} \cdot \beta\right)$-fraction of bits inside $(w'|_{\mathrm{msg}})|_{A(C)}$, to get $w'|_{\mathrm{msg}} \in C$.

Let $modified \subseteq \mathrm{msg}$ be these bits coordinates that should be changed. We say that an Hadamard block $(w'_{\mathrm{block_j}})$ is good if $\mathrm{block_j} \cap \mathrm{modified} = \emptyset$, and otherwise it is called bad. Intuitively, good Hadamard blocks contain only "correct" message bits. The bits indexed by $modified$ are contained in at most $t \cdot \left(\frac{1}{t} \cdot \left(\frac{2t}{\epsilon} \cdot \beta\right)\right)$-fraction of the Hadamard blocks, because each Hadamard block contains t message bits (the bits indexed by msg). Similarly, the bits indexed by $A(C) \cap modified$ are contained in at most $t \cdot \left(\frac{1}{t} \cdot \left(\frac{2t}{\epsilon} \cdot \beta\right)\right)$-fraction of the Hadamard blocks of $w'|_{A(C)}$.

In the worst case, all bits in bad Hadamard blocks containing message bits from $modified$ should be changed. Hence at most $\left(\frac{2t}{\epsilon} \cdot \beta\right)$-fraction of bits in w' should be changed in order to receive all message bits correct $(w'|_{\mathrm{msg}} \in C)$ and all bad Hadamard blocks belong to H. The same fraction of Hadamard blocks is modified inside $w'|_{A(C')}$, where we recall that bits of $A(C')$ are the sequential encodings by the Hadamard code of bits indexed by $A(C)$.

All bit modifications are done only inside the bad Hadamard blocks. In the rest of the proof we are going to modify some bits of w', inside the good Hadamard blocks. But all bits we will modify will be outside the set msg.

Recall that the total number of the Hadamard blocks is n, while the blocklength of the code is $n \cdot 2^t$, and every Hadamard block contains t message symbols from msg.

A typical Hadamard block is close to fit its message. Recall that Proposition 8.6 says that every Hadamard block $w'_{\mathrm{block_j}}$ is rejected by the LTCM-tester $\mathcal{T}_H$ with probability at least $\frac{1}{16t}\delta_{msg}(w'_{\mathrm{block_j}}, H)$. We also recall that $\delta_{msg}(w'_{\mathrm{block_j}}, H)$ shows the fraction of non-message bits that should be changed to get a codeword, i.e., no message bit is modified.

By definition of $\mathcal{T}_{C'}$, with probability $\frac{1}{4}$ the LTCM-tester $\mathcal{T}_H$ is invoked on a random Hadamard block. Since the rejection probability of $\mathcal{T}_{C'}$ is β, it follows that for an average Hadamard block $w'_{\mathrm{block_j}}$ it holds that $\delta_{msg}(w'_{\mathrm{block_j}}, H) \leq (4 \cdot 16t) \cdot \beta$. It also holds that with probability $\frac{1}{4}$ the LTCM-tester $\mathcal{T}_H$ is invoked on a random j-th Hadamard block for $j \in A(C)$. Thus, it follows that for an average Hadamard block $w'_{\mathrm{block_j}}$ with $j \in A(C)$ it holds that $\delta_{msg}(w'_{\mathrm{block_j}}, H) \leq (4 \cdot 16t) \cdot \beta$.

We consider only good Hadamard blocks and conclude that it is sufficient to modify at most $(4 \cdot 16t) \cdot \beta$ fraction of bits in w', which includes at most $(4 \cdot 16t) \cdot \beta$ fraction of bits in $w'|_{A(C')}$, to obtain a situation where all good Hadamard blocks belong to H. Note that only the non-message bits of the good Hadamard blocks are modified here.

Summary. It follows that we modify at most $\frac{2t}{\epsilon} \cdot \beta + (4 \cdot 16t) \cdot \beta \leq \frac{66t}{\epsilon} \cdot \beta$ fraction of bits in w', which includes at most $\frac{66t}{\epsilon} \cdot \beta$ fraction of bits inside $w'|_{A(C')}$, to receive a codeword of C'. Hence $\delta_{A(C')}(w', C') \leq \frac{66t}{\epsilon} \cdot \beta$.

9 Distance Amplification Procedure

In this section we present a procedure DistAmp$(\cdot)$ that increases the distance of a strong LTC, while preserving its query complexity. The idea behind this result is similar to [27, Section 4.3]. The distance amplification procedures are well-known in the area of error-correcting codes (e.g., [1]). Meir [27] showed that the distance of the underlying LTC can be amplified such that the resulting code is still an LTC but the query complexity was slightly increased. In Theorem 9.1 we show how to implement this procedure to remain the query complexity intact. Then, in Theorem 9.3 we use Theorem 9.1 to show a way to improve the distance of a COLTC whose core is a two-wise tensor product to preserve the query complexity of the COLTC and preserve the tensor structure of its core.

We stress that the distance amplification procedure we present here can be considered as a special case of the corresponding procedure in the work of Meir [27, Section 4.3]. This is true since in [27] the author considered the concatenation for general codes, while we used the concatenation with the repeated Hadamard code (Definition 8.3 and Theorem 8.4), exploiting that fact that the Hadamard code is locally testable, locally decodable and its codewords contain message bits as a part of the encoding. Hence the observations made in [27] are valid for our case as well.

Theorem 9.1. *Let $0 < \delta_0 < \delta_1 < \frac{1}{2}$ and $q \geq 3$ be two constants. Then there exists a sufficiently large constant $t = t(\delta_1/\delta_0)$ and deterministic distance amplification procedure* $\mathrm{DistAmp_t}(\cdot)$ *that given a linear code $C \subseteq \mathbb{F}_2^n$ with a δ_0-core $A(C)$ such that C is a (q, ϵ)-COLTC, outputs the code* $C' = \mathrm{DistAmp_t(C)}$ *that satisfies:*

- *$A(C')$ is a δ_1-core of C',*
- $\mathrm{blocklength}(C') = 2^t \cdot \mathrm{blocklength}(C)$,
- $\dim(C') = \dim(C)$,
- *C' is a $(q, \frac{\epsilon}{528t})$-COLTC.*

Proof of Theorem 9.1. Any codeword of C can be viewed as a binary string of length n. We recall Definition A.1 presented in Section A and let $C_{(t)} \subseteq \mathbb{F}_2^{t \cdot n}$ be the t-repetition of the code C, where t is sufficiently large integer that will be fixed in the rest of the proof. Proposition A.3 says that $C_{(t)}$ is a $(q, \epsilon/8)$-COLTC (for every $t \geq 1$). It is also easy to see that $\delta(C_{(t)}) = \delta(C) \geq \delta_0$. Note that a code $C_{(t)}$ can be viewed as a sequence of t blocks, where every block has length n and belongs to C. Note that $A(C_{(t)})$ is a δ_0-core for $C_{(t)}$ because $A(C)$ is a δ_0-core for C.

Now for a permutation $\sigma : [n] \to [n]$ and a word $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ let $\sigma(x) = (x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)})$ be the word x permuted by σ. Similarly, let $\sigma(C) = \{\sigma(c) \mid c \in C\}$. In this case we say that a permutation σ is a permutation over $[n]$.

Given $t-1$ permutations $\sigma_1, \sigma_2, \ldots, \sigma_{t-1}$ over $[n]$ we say that a word $((\sigma_1, \sigma_2, \ldots, \sigma_{t-1})(w)) \in \mathbb{F}_2^{t \cdot n}$ is a word $w \in \mathbb{F}_2^{t \cdot n}$ permuted by $(\sigma_1, \sigma_2, \ldots, \sigma_{t-1})$ if letting $w' = ((\sigma_1, \sigma_2, \ldots, \sigma_{t-1})(w))$ it holds that $w'|_{[n]} = w|_{[n]}$ and for every $j \in [t-1]$ we have $w'|_{[n]+jn} = \sigma_j(w|_{[n]+jn})$. Similarly,

with some abuse of notations we let $((\sigma_1,\sigma_2,\ldots,\sigma_{t-1})(C_{(t)}))\subseteq\mathbb{F}_2^{t\cdot n}$ be the code $C_{(t)}$ permuted by $(\sigma_1,\sigma_2,\ldots,\sigma_{t-1})$ if letting $\hat{C}=((\sigma_1,\sigma_2,\ldots,\sigma_{t-1})(C_{(t)}))$ it holds that

$$\hat{C}=\left\{(\sigma_1,\sigma_2,\ldots,\sigma_{t-1})(c)\mid c\in C_{(t)}\right\}.$$

Meir [27] (see Sections 3.2.3, 4.3, and Definition 4.10 in [27]) explained that when $t=t(\delta_1/\delta_0)$ is sufficiently large integer it is possible to deterministically select permutations $\sigma_1,\sigma_2,\ldots,\sigma_{t-1}$ over $[n]$ such that letting

$$\hat{C}=((\sigma_1,\sigma_2,\ldots,\sigma_{t-1})(C_{(t)})),$$

for every $\hat{c}\in\hat{C}\setminus\left\{0^{t\cdot n}\right\}$ it follows that for at least $(2\delta_1)$ fraction of indices $j\in[n]$ we have at least one non-zero bit in $\left\{\hat{c}|_j,\hat{c}|_{j+n},\hat{c}|_{j+2n},\ldots,\hat{c}|_{j+(t-1)n}\right\}$. Note that $2\delta_1<1$. In the work [27], this observation was shown using the (deterministic) walk of length $t-1$ on the regular expander graph G, where $[n]$ was associated with the vertex set of G (see Section 3.2.3 in [27]). So, for the coordinate j the walk in the graph was a t-tuple: $j=j_1,j_2,\ldots,j_t$. In our proof, this walk corresponding to the sequence $\hat{c}|_j,\hat{c}|_{j+n},\hat{c}|_{j+2n},\ldots,\hat{c}|_{j+(t-1)n}$, where the codeword $\hat{c}$ belongs to the permuted code $\hat{C}$. We notice that $\dim(\hat{C})=\dim(C_{(t)})$ and $\hat{C}$ is a $(q,\epsilon/8)$-COLTC.

Now, since $A(C_{(t)})$ is a δ_0-core for $C_{(t)}$ that means $\delta(C_{(t)}|_{A(C_{(t)})})\geq\delta_0$ and $\delta(C_{(t)}|_{-A(C_{(t)})})\geq\delta_0$. Moreover, the core coordinates $A(C_{(t)})$ is a t-repetition of $A(C)$. Thus the permutations $\sigma_1,\sigma_2,\ldots,\sigma_{t-1}$ over $[n]$ can be selected exactly as described above but such that the repetition of $A(C)$ coordinates will be permuted between them and the repetition of $[n]\setminus A(C)$ coordinates will be permuted between them. In this way, we will also deterministically select these permutations $\sigma_1,\sigma_2,\ldots,\sigma_{t-1}$ over $[n]$ such that letting

$$\hat{C}=((\sigma_1,\sigma_2,\ldots,\sigma_{t-1})(C_{(t)})),$$

for every $\hat{c}\in\hat{C}\setminus\left\{0^{t\cdot n}\right\}$ it follows that for at least $(2\delta_1)$ fraction of indices $j\in A(C)$ we have at least one non-zero bit in $\left\{\hat{c}|_j,\hat{c}|_{j+n},\hat{c}|_{j+2n},\ldots,\hat{c}|_{j+(t-1)n}\right\}$;
and for at least $(2\delta_1)$ fraction of indices $j\in[n]\setminus A(C)$ we have at least one non-zero bit in $\left\{\hat{c}|_j,\hat{c}|_{j+n},\hat{c}|_{j+2n},\ldots,\hat{c}|_{j+(t-1)n}\right\}$. We fix this $\hat{C}$ and notice that the core $A(\hat{C})$ is obtained by the corresponding permutation of $A(C_{(t)})$.

Let $R\subseteq\mathbb{F}_2^{t\cdot n}$ be a code which is equal to $\hat{C}$ besides a regrouping of bits such that for every $j\in[n]$ the indices $j,j+n,j+2n,\ldots,j+(t-1)n$ become adjacent. More formally, $r\in R$ iff there exists $c\in\hat{C}$ such that for every $i\in[t]$ and $j\in[n]$ we have $c|_{(j+(i-1)n)}=r|_{((j-1)t+i)}$. Clearly, the core $A(R)$ will be obtained by the corresponding regrouping of $A(\hat{C})$. We notice that R is a linear code, $\dim(R)=\dim(C_{(t)})$ and R is a $(q,\epsilon/8)$-COLTC. By construction, for every non-zero codeword $r\in R$ it holds that for at least $2\delta_1$ fraction of indices $j\in A(R)$ we have at least one non-zero bit in $\left\{r|_{((j-1)t+1)},r|_{((j-1)t+2)},\ldots,r|_{((j-1)t+t)}\right\}$, and for at least $2\delta_1$ fraction of indices $j\in[n]\setminus A(R)$ we have at least one non-zero bit in $\left\{r|_{((j-1)t+1)},r|_{((j-1)t+2)},\ldots,r|_{((j-1)t+t)}\right\}$. We assume without loss of generality that $A(R)=[|A(R)|]$, since the coordinates can be re-enumerated.

Let $H\subseteq\mathbb{F}_2^{2^t}$ be the Hadamard code. Let $C'=H^{[n]}\circ R$ be the concatenation of $H^{[n]}$ and R, where $H^{[n]}$ and the concatenation operation were defined in Definition 8.3. Note that $C':\mathbb{F}_2^k\to\mathbb{F}_2^{n\cdot 2^t}$, i.e., $\mathrm{blocklength}(C')=2^t\cdot\mathrm{blocklength}(C)$. Theorem 8.4 says that C' is a $(q,\frac{\epsilon/8}{66t})$-COLTC, where the core $A(C')$ is defined as in Definition 8.3.

We argue that $\delta(C')\geq\delta_1$ and $\delta(C'|_{A(C')})\geq\delta_1$. Too see this let $c'\in C'$ be a non-zero codeword. Since C' is linear it is sufficient to prove that $\frac{|c'|}{\mathrm{blocklength}(C')}\geq\delta_1$ and $\frac{|c'|_{A(C')}|}{|A(C')|}\geq\delta_1$. We recall that

$\delta(H) \geq \frac{1}{2}$, i.e., the encoding of every non-zero message by H has relative weight at least $\frac{1}{2}$. We also recall that for at least $2\delta_1$ fraction of indices $j \in A(R)$ (and of $j \in ([n \cdot t] \setminus A(R))$) we have at least one non-zero bit in the sequence $r|_{((j-1)t+1)}, r|_{((j-1)t+2)}, \ldots, r|_{((j-1)t+t)}$, which serves a (non-zero) message for H in the concatenation of $H^{[n]}$ and R. I.e., at least $2\delta_1$ fraction of messages (in $A(R)$ and in $[n \cdot t] \setminus A(R)$) are non-zero messages and the code C' can be viewed as a sequence of encodings of every such message by H. Thus $\frac{|c'|}{\text{blocklength}(C')} \geq \frac{1}{2} \cdot (2\delta_1) = \delta_1$ and similarly, $\frac{|c'|_{A(C')}|}{|A(C')|} \geq \frac{1}{2} \cdot (2\delta_1) = \delta_1$. The arguments above show also that $\dim(C'|_{A(C')}) = \dim(C'|_{-A(C')}) = \dim(C')$. We conclude that $A(C')$ is a δ_1-core of C'. □

Remark 9.2. We stress that the above distance amplification procedure is dependent on the choice of the permutations $\sigma_1, \sigma_2, \ldots, \sigma_{t-1}$ over $[n]$. Given these permutations, the output of the algorithm is uniquely defined. Now, in Theorem 9.1 we argued that these permutations can be found deterministically, using expanders walk (see Sections 3.2.3, 4.3, and Definition 4.10 in [27]). The crucial point is that finding "appropriate" permutations are important *only* to increase the distance of the underlying code. That means for *any* selection of the permutations $\sigma_1, \sigma_2, \ldots, \sigma_{t-1}$ over $[n]$ in the proof of Theorem 9.1, we will conclude that the resulting code C' has $\text{blocklength}(C') = 2^t \cdot \text{blocklength}(C)$, $\dim(C') = \dim(C)$ and C' is a $(q, \frac{\epsilon}{528t})$-COLTC. This observation will play an essential role in Theorem 9.3 and we will refer to the different executions of the procedure $\text{DistAmp}_{\text{t}}(\cdot)$ that actually means different executions of this procedure defined by different selection of the permutations $\sigma_1, \sigma_2, \ldots, \sigma_{t-1}$.

For example, the procedure $\text{DistAmp}_{\text{t}}(\cdot)$ can be invoked on the input code $R \subseteq \mathbb{F}_2^n$ such that the code R has no defined core and is not a COLTC. In this case, we can artificially set $A(R) = [n_R]$, i.e., to be the core containing all coordinates, and the output code C' is not guaranteed to be a COLTC; however C' will have the dimension, the blocklength and the relative distance (increased up to δ_1) as guaranteed by Theorem 9.3. Similarly, given two linear codes C_1 and C_2, the code $(\text{DistAmp}_{\text{t}}(\text{C}_1), \text{DistAmp}_{\text{t}}(\text{C}_2))$ can be obtained by the execution of $\text{DistAmp}_{\text{t}}((\text{C}_1, \text{C}_2))$, where (C_1, C_2) means that the code C_2 is appended to the code C_1.

We are ready to state and prove the main theorem of this section.

Theorem 9.3. *Let $0 < \delta_0, \delta_1 < 1/2$ be constants. Assume C is a linear code and let $A(C)$ be its δ_0^2-core. Assume also that C is a (q, ϵ)-COLTC and $C|_{A(C)} = R^{\otimes 2}$ for some linear code R such that $\delta(R) \geq \delta_0$.*

Then there exists a deterministic procedure $\text{DistAmp}'_{\text{t}}(\cdot)$ and sufficiently large constant $t > 0$ such that letting $C' = \text{DistAmp}'_{\text{t}^2}(\text{C})$ we have

- *$\dim(C') = \dim(C)$ and $\text{rate}(C') = \Omega(\text{rate}(C))$,*
- *C' is a $(q, \frac{\epsilon}{528t^2})$-COLTC,*
- *$C'|_{A(C')} = R' \otimes R'$ for some linear code R',*
- *$A(C')$ is a δ_1^2-core.*

Proof. Let $t > 0$ be the sufficiently large integer promised guaranteed by Theorem 8.4 to increase the relative distance from δ_0 to δ_1. Recall that C has a core $A(C)$ such that $C|_{A(C)} = R \otimes R$, and let

$$\text{DistAmp}'_{\text{t}}(\text{C}) = \left((\text{DistAmp}_{\text{t}}(\text{R}))^{\otimes 2}\ ,\ \text{DistAmp}_{\text{t}^2}(\text{C}|_{-\text{A(C)}})\right),$$

where $(C_1,\ C_2)$ was defined in Section 2. Let $C' = \mathrm{DistAmp}'_{\mathrm{t}}(\mathrm{C})$. Let $A(\mathrm{DistAmp}'_{\mathrm{t}}(\mathrm{C}))$ be the coordinate set of $(\mathrm{DistAmp}_{\mathrm{t}}(\mathrm{R}))^{\otimes 2}$. So, letting $R' = \mathrm{DistAmp}_{\mathrm{t}}(\mathrm{R})$ it holds that R' is a linear code, $\delta(R') \geq \delta_1$ since $\delta(R) \geq \delta_0$, and $C'|_{A(C')} = R' \otimes R'$. Thus $\delta(C'|_{A(C')}) \geq \delta_1^2$ and $\delta(C'|_{-A(C')}) = \delta(\mathrm{DistAmp}_{\mathrm{t}^2}(\mathrm{C}|_{-\mathrm{A(C)}})) \geq \delta_1^2$. It follows that $A(C')$ is a δ_1^2-core of C'.

Meir [27, Section 5] pointed out that $\mathrm{DistAmp}_{\mathrm{t}}(\cdot)$ procedure can be applied to the tensor products and preserves its structure. That means $(\mathrm{DistAmp}_{\mathrm{t}}(\mathrm{R}))^{\otimes 2}$ is equal to one of the possible outputs of $\mathrm{DistAmp}_{\mathrm{t}^2}(\mathrm{R}^{\otimes 2})$ (see Remark 9.2). This implies that

$$C' = \left(\mathrm{DistAmp}_{\mathrm{t}^2}(\mathrm{R}^{\otimes 2})\ ,\ \mathrm{DistAmp}_{\mathrm{t}^2}(\mathrm{C}|_{-\mathrm{A(C)}})\right),$$

where the intention is that such a result is possible for some of the executions of $\mathrm{DistAmp}(\cdot)$ procedure, as explained in Remark 9.2.

Finally, as was explained in Remark 9.2, amplifying the distance of the code separately on its first part of the coordinates and on its second part of the coordinates is one of the possibilities for amplifying the distance for the entire code. Thus, $C' = \mathrm{DistAmp}_{\mathrm{t}^2}(\mathrm{R}^{\otimes 2}\ ,\ \mathrm{C}|_{-\mathrm{A(C)}}) = \mathrm{DistAmp}_{\mathrm{t}^2}(\mathrm{C})$. Theorem 9.1 says that if C is a (q,ϵ)-COLTC then C' is a $(q, \frac{\epsilon}{528t^2})$-COLTC. Also, Theorem 9.1 implies that $\dim(C') = \dim(C)$ and $\mathrm{rate}(C') = \Omega(\mathrm{rate}(C))$ since t is a constant. □

10 Random Projection Procedure

We start this section by defining the random projection operation $\mathrm{RandProj}(\cdot)$. Although this procedure had appeared in [27, Section 4.2], the corresponding proofs should be revisited according to our needs, e.g., showing that the distance of the underlying code ($\delta(C)$) is preserved as well as the distance inside its core ($\delta(C|_{A(C)})$), and proving the fact that the core oriented testability (Definition 2.4) is preserved after this procedure is applied.

In words, this operation receives a linear code C, whose core ($A(C)$) is a tensor product of some linear code R. If the rate of R is sufficiently high then this operation does nothing. Otherwise, it "shortens" the core of the code such that this core remains to be a tensor product of some other linear code. I.e., this operation does not affect the code itself but redefines (probabilistically) its core.

Algorithm 1 Random Projection

Input: $C \subseteq \mathbb{F}_2^n$ such that $A(C) = [n_R] \times [n_R]$ is the core of C and $C|_{A(C)} = R \otimes R$ for the linear code $R \subseteq \mathbb{F}_2^{n_R}$.

if $\left(\mathrm{rate}(R) > \dfrac{\delta^2(R)}{100}\right)$ **then**

Output C.

else

Pick random $S_R \subseteq [n_R]$ such that $|S_R| \leq \dfrac{10 \cdot \dim(R)}{\delta^2(R)}$.

Let $C' = C$ and fix $A(C') = S_R \times S_R$.

Output C'.

Now we state Theorem 10.1. Informally, Theorem 10.1 says that with high probability the procedure $\mathrm{RandProj}(\cdot)$ outputs the same code as an input code but redefines its core such that this new core has constant rate and relative distance, and has a shape of a tensor product.

Theorem 10.1. *Let $\gamma > 0$ be a constant and $n_R \in \mathbb{N}^+$. Let $C \subseteq \mathbb{F}_2^n$ be a code and $A(C) = [n_R] \times [n_R]$ be its γ^2-core such that $C|_{A(C)} = R \otimes R$ for the linear code $R \subseteq \mathbb{F}_2^{n_R}$. Assume that C is a (q, ϵ)-COLTC.*

Then, letting $C' = \mathrm{RandProj(C)}$, with probability at least $1 - \exp(-\Omega(\dim(R)))$ we have

1. *$A(C')$ is a $(\gamma^2/4)$-core,*
2. *$C'|_{A(C')} = R' \otimes R'$ for some linear code R' such that $\delta(R') \geq \gamma/2$,*
3. *$\mathrm{rate}(C'|_{A(C')}) \geq \max\left\{\frac{\gamma^4}{100}, \mathrm{rate}(C|_{A(C)})\right\}$,*
4. *$\dim(C) = \dim(C')$, $\mathrm{blocklength}(C) = \mathrm{blocklength}(C')$ and $\Delta(C) = \Delta(C')$,*
5. *C' is a $(q, \epsilon \cdot \mathrm{rate}(C|_{A(C)}))$-COLTC.*

We prove Theorem 10.1 in Section 10.1.

10.1 Proof of Theorem 10.1

Let us consider two auxiliary statements, and then we prove Theorem 10.1. The following proposition and its proof appeared in [27]. We reproduce it here for the sake of completeness.

Proposition 10.2 ([27])**.** *Let $C \subseteq \mathbb{F}_2^n$ be a code such that $\mathrm{rate}(C) \leq \frac{\delta^2(C)}{100}$. Let $h = \frac{10 \cdot \dim(C)}{\delta^2(C)}$. Then*

$$\Pr_{S \subseteq [n], |S| \leq h}\left[\delta(C|_S) \geq \frac{\delta(C)}{2} \quad \text{and} \quad \dim(C|_S) = \dim(C)\right] \geq 1 - \exp(-\Omega(\dim(C))),$$

where the probability is taken over a uniform selection of $S \subseteq [n]$ such that $|S| \leq h$.

Proof. To prove that $C|_S$ has relative distance at least $\frac{\delta(C)}{2}$ and that $\dim(C|_S) = \dim(C)$, we use a standard probabilistic argument. Fix a non-zero codeword $c \in C$, and let $S \subseteq [n]$ be a uniformly chosen set such that $|S| \leq h$. The relative weight of c is at least $\delta(C)$, and therefore the expected relative weight of $c|_S$ is at least $\delta(C)$. Applying the Chernoff Bound, it follows that the probability that the relative weight of $c|_S$ is less than $\frac{\delta(C)}{2}$ is at most $2\exp(-\frac{1}{4} \cdot \delta^2(C) \cdot h)$.

By taking a union bound over all the codewords of C, the probability that there exists a non-zero codeword $c \in C$ such that $c|_S$ has relative weight less than $\frac{\delta(C)}{2}$ is bounded by

$$2^{\dim(C)} \cdot 2\exp(-\frac{1}{4} \cdot \delta^2(C) \cdot h) \leq 2\exp(-\Omega(\dim(C))).$$

□

Remark 10.3. One can see that Proposition 10.2 holds (up to some change in the constants) for any constant-size field. However, this Proposition becomes problematic when the field size $|\mathbb{F}|$ depends on the blocklength n. This issue was solved by [29] using smarter arguments, i.e., the statement in Proposition 10.2 is true (up to the small change in constants) for any field $\mathbb{F}$ and does not dependent on the field size.

Assume that $C \subseteq \mathbb{F}_2^n$ is a code such that $\text{rate}(C) = \frac{\dim(C)}{n} \leq \frac{\delta^2(C)}{100}$. Notice that when one selects $S \subseteq [n]$ at random such that $|S| \leq \frac{10 \cdot \dim(C)}{\delta^2(C)}$, then $\bar{S} = [n] \setminus S$ is selected at random such that $|\bar{S}| \geq \frac{90 \cdot \dim(C)}{\delta^2(C)}$. Similarly to Proposition 10.2, one can obtain the following proposition.

Proposition 10.4. *Let $C \subseteq \mathbb{F}_2^n$ be a code such that $\text{rate}(C) \leq \frac{\delta^2(C)}{100}$. Let $h = \frac{10 \cdot \dim(C)}{\delta^2(C)}$. Then*

$$\Pr_{S \subseteq [n], |S| \leq h}\left[\delta(C|_{[n]\setminus S}) \geq \frac{\delta(C)}{2} \quad \text{and} \quad \dim(C|_{[n]\setminus S}) = \dim(C)\right] \leq 1 - \exp(-\Omega(\dim(C))),$$

where the probability is taken over a uniform selection of $S \subseteq [n]$ such that $|S| \leq h$.

We are ready to prove Theorem 10.1.

Proof of Theorem 10.1. We know that $C|_{A(C)} = R \otimes R$ and $\delta(C|_{A(C)}) = \delta^2(R)$. We also know that $A(C)$ is a γ^2-core, i.e., $\delta(C|_{A(C)}) \geq \gamma^2$. Thus $\delta(R) \geq \gamma$.

If $\text{rate}(R) > \frac{\delta^2(R)}{100}$ then $C' = C$, $A(C') = A(C)$ and the Theorem holds.

Otherwise, we have $\text{rate}(R) \leq \frac{\delta^2(R)}{100}$. Propositions 10.2 and 10.4 imply that with probability at least $1 - \exp(-\Omega(\dim(R)))$ we have

$$\delta(R|_{S_R}) \geq \frac{\gamma}{2}, \quad \delta(R|_{[n]\setminus S_R}) \geq \frac{\gamma}{2}, \quad \text{and} \quad \dim(R|_{S_R}) = \dim(R|_{[n_R]\setminus S_R}) = \dim(R).$$

Given these facts we prove all parts of Theorem 10.1.

Proof of Part 1. It follows that $A(C')$ is a $(\gamma^2/4)$-core since $\delta(C'|_{A(C')}) \geq \delta^2(R|_{S_R}) \geq \gamma^2/4$ and it can be verified that

$$\delta(C'|_{-A(C')}) \geq \min\left\{\delta^2(R|_{S_R}), \delta(R|_{S_R}) \cdot \delta(R|_{[n]\setminus S_R}), \delta^2(R|_{[n]\setminus S_R}), \delta(C|_{-A(C)})\right\} \geq \gamma^2/4.$$

Proof of Part 2. By definition of RandProj we have $C'|_{A(C')} = R|_{S_R} \otimes R|_{S_R}$ for the linear code $R|_{S_R}$ such that $\delta(R|_{S_R}) \geq \gamma/2$.

Proof of Part 3. By definition of RandProj it holds that $|S_R| \leq \frac{10 \dim(R)}{\delta^2(R)}$ and $\text{rate}(C'|_{A(C')}) \geq \left(\frac{\delta^2(R)}{10}\right)^2 \geq \frac{\gamma^4}{100}$.

Proof of Part 4. Since $C' = C$ it follows that

$$\dim(C) = \dim(C'), \quad \text{blocklength}(C) = \text{blocklength}(C') \quad \text{and} \quad \Delta(C) = \Delta(C').$$

Proof of Part 5. We know that the core of C' was modified such that $A(C') \subseteq A(C)$ and $|A(C')| \geq \text{rate}(C|_{A(C)}) \cdot |A(C)|$ since the dimension of the core stays intact. We argue that the tester for C, which is a (q, ϵ)-COLTC tester is also a $(q, \epsilon \cdot \text{rate}(C|_{A(C)}))$-COLTC tester for C'. This is true since for every $w \in \mathbb{F}_2^n$ we have $\delta(w|_{A(C)}, C|_{A(C)}) \geq \text{rate}(C|_{A(C)}) \cdot \delta(w|_{A(C')}, C|_{A(C')})$ and w is rejected by the COLTC tester of C with probability at least

$$\epsilon \cdot \delta(w|_{A(C)}, C|_{A(C)}) \geq \left(\epsilon \cdot \text{rate}(C|_{A(C)})\right) \cdot \delta(w|_{A(C')}, C|_{A(C')}).$$

Thus C' is a $(q, \epsilon \cdot \text{rate}(C|_{A(C)}))$-COLTC. □

11 Proof of Main Results

We prove Theorem 3.1.

Proof of Theorem 3.1. First we present an iterative construction of COLTCs. Then we prove that these codes have the required range of parameters.

Let $R = \mathbb{F}_2^2 = \{00, 01, 10, 11\}$ and fix a core $A(R) = \mathrm{supp}(R) = \{1, 2\}$. We let $C_1 = R^{\otimes 2}$ and note that $C_1 = C_1^{\star 2} = C_1|_{A(C_1)} = R^{\otimes 2} = \mathbb{F}_2^{2\times 2}$ because $A(R) = \mathrm{supp}(R)$. Let $\mathcal{D}_{C_1}$ be the tester for C_1 that always accepts any given word (with $0 \leq 3$ queries). Thus $\mathcal{D}_{C_1}$ always accepts the codewords of C_1 and rejects all non-codewords with probability 1 (vacuously). I.e., C_1 is a $(3, 1)$-COLTC. We have:

- $\dim(C_1) = 2^{2^1} = 4$ and $\mathrm{blocklength}(C_1) = 4$,
- $\mathrm{rate}(C_1) = \mathrm{rate}(C_1|_{A(C_1)}) = 1$,
- $\delta(C_1) = \delta^2(R) = \delta^2(R|_{A(R)}) = \frac{1}{2^2} = \frac{1}{4}$,
- $A(C_1) = A(R) \times A(R)$ is a $\frac{1}{4}$-core of C_1 and $C_1|_{A(C_1)} = R^{\otimes 2}$, and
- C_1 is a $(3, 1)$-COLTC.

Now we provide a probabilistic algorithm that constructs COLTCs.

Algorithm 2 Construction of COLTCs

Input: $m \in \mathbb{N}^+$
for every $i \in [m]$ **do**
  $C' := C_i^{\star 4}$
  $C'' := \mathrm{RandProj}(\mathrm{C}')$
  $C_{i+1} := \mathrm{DistAmp}'(\mathrm{C}'')$
Output C_m.

We want to prove that the output code of Algorithm 2, denoted by C_m, satisfies the required properties. To do that we prove that for all $i \in [m]$ it holds that C_i satisfy the following properties, where we fix $\gamma = \frac{1}{100}$.

- $\dim(C_i) = 2^{2^i}$,
- $\mathrm{rate}(C_i) \geq \frac{1}{\mathrm{polylog}(n_i)}$ and $\mathrm{rate}(C_i|_{A(C_i)}) = \Omega(1)$,
- $\delta(C_i) \geq \gamma^2$ and $\delta(C_i|_{A(C_i)}) \geq \gamma^2$,
- $A(C_i)$ is a γ^2-core and $C_i|_{A(C_i)} = R_i^{\otimes 2}$ for some linear code R_i, and
- $C_i \subseteq \mathbb{F}_2^{n_i}$ is a $(3, \frac{1}{\mathrm{polylog}(n_i)})$-COLTC.

We have showed that this is true for the code $C_1 = C_1^{\star 2}$. Let us prove that for all $i \in [m]$ the code C_i satisfies these properties.

Dimension. By induction, for all $i \in [m-1]$ we have $\dim(C_{i+1}) = \dim(C_i^{\star 4}) = \dim^2(C_i) = (2^{2^i})^2 = 2^{2^{i+1}}$ (see Claim 7.6) since the operation $(\star 4)$ squares the dimension, while the operations RandProj$(\cdot)$ and DistAmp$'(\cdot)$ do not change the dimension of the underlying code (see Theorems 10.1 and 9.3). Notice that this imply in particular, if $\dim(C_i) = k$ then $i = \log\log k$.

The rate of $C_i|_{A(C_i)}$ and of C_i. For all $i \in [m]$ it can be verified that $\text{rate}(C_i|_{A(C_i)}) = \frac{\dim(C_i)}{|A(C_i)|}$ is a fixed constant. This is true since it was a fixed constant in C_1, and it is preserved to be a fixed constant using RandProj$(\cdot)$ operation, while DistAmp$'(\cdot)$ and the star product $(\star 4)$ can decrease this rate only by a constant (see Claim 7.6). Each iteration this rate is restored by RandProj$(\cdot)$ operation to be a fixed constant (see Theorem 10.1).

Given that $\text{rate}(C_i|_{A(C_i)})$ is preserved to be a fixed constant for all $i \in [m]$, it holds that every iteration $\text{rate}(C_i)$ is decreased by a fixed multiplicative constant after the star product $(\star 4)$ (see Claim 7.6) and the distance amplification procedure DistAmp$'(\cdot)$, while RandProj$(\cdot)$ does not affect the rate of the input code. Thus $\text{rate}(C_i) \geq \frac{1}{\text{polylog}(n_i)}$.

Distance of C_i and of $C_i|_{A(C_i)}$. It can be easily verified that for every $i \in [m]$ it holds that $\delta(C_i) \geq \gamma^2$ and $\delta(C_i|_{A(C_i)}) \geq \gamma^2$. This is true since given that it is satisfied for a code C_i we conclude that the distance and the core distance of the code after the star product and after the random projection is reduced only by a multiplicative fixed constant (see Claim 7.6 and Theorem 10.1), while after the distance amplification procedure the distance and the core distance of the code are increased up to γ^2 (Theorem 9.3).

$A(C_i)$ is a γ^2-core and $C_i|_{A(C_i)} = R_i^{\otimes 2}$ for some linear code R_i. The facts that $\delta(C_i) \geq \gamma^2$, $\delta(C_i|_{A(C_i)}) \geq \gamma^2$ and $A(C_i)$ is a core of C_i after each step, imply that $A(C_i)$ is a γ^2-core of C_i. Moreover, the star product, the random projection and the distance amplification procedures (see Theorems 10.1 and 9.3) preserve the fact that $C_i|_{A(C_i)}$ is always a two-wise tensor product of some linear code.

For example, if $C_i|_{A(C_i)} = R_i^{\otimes 2}$ then $C_i^{\star 2} = C_i$ and $C_i^{\star 4}|_{A(C_i^{\star 4})} = R_i^{\otimes 4} = (R_i')^{\otimes 2}$ for $R_i' = R_i^{\otimes 2}$.

$C_i \subseteq \mathbb{F}_2^{n_i}$ is a $(3, \frac{1}{\text{polylog}(n_i)})$-COLTC. Corollary 7.9 implies that the star product $(\star 4)$ decreases the soundness parameter of a COLTC only by a fixed constant, while the query complexity stays intact. Similarly, the random projection and the distance amplification procedures (Theorems 10.1 and 9.3) reduce the soundness parameter only by a fixed constant, but remain the query complexity unaffected. Since the code C_1 is a $(3,1)$-COLTC, we conclude that for every $i \in [m]$ it follows that $C_i \subseteq \mathbb{F}_2^{n_i}$ is a $(3, \frac{1}{\text{polylog}(n_i)})$-COLTC. □

11.1 Proof of Corollary 3.2

Proof of Corollary 3.2. Let $C_m \subseteq \mathbb{F}_2^n$ be a linear $(3, \frac{1}{\text{polylog}(n)})$-COLTC as guaranteed by Theorem 3.1. Let T_m be the associated 3-query tester for C_m. We know that this tester always accepts all codewords of C_m and there exists a constant $d \in \mathbb{N}$ such that every word $w \notin C_m$ is rejected by T_m with probability at least $\frac{1}{\log^d n} \cdot \delta(w, C_m)$. We let T_m' be the new tester that invokes the tester T_m on the input word $\log^d n$ times and rejects if and only if at least one invocation of the tester T_m

rejected. Clearly, the tester T'_m has query complexity $3 \cdot \log^d n$ and always accepts all codewords of C_m.

Let $w \notin C_m$ be an input word. The tester T'_m accepts w with probability at most $\left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\log^d n}$ and rejects w with probability at least $1 - \left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\log^d n}$. We argue that $1 - \left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\log^d n} > \frac{1}{2} \cdot \delta(w, C_m)$ and this yields the Corollary.

It holds that $\left(\left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\log^d n}\right)^{\frac{1}{\delta(w,C_m)}} = \left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\frac{\log^d n}{\delta(w,C_m)}} \leq e^{-1} < \frac{1}{2}$ and $1 - \left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\frac{\log^d n}{\delta(w,C_m)}} > \frac{1}{2}$. On the other hand,

$$1 - \left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\frac{\log^d n}{\delta(w,C_m)}} \leq \frac{1}{\delta(w,C_m)} \cdot \left(1 - \left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\log^d n}\right),$$

where we used the fact that $1 - p^l \leq l \cdot (1-p)$ for $p \leq 1$ and $l \in \mathbb{N}^+$.[7] The required inequality is obtained by replacing p with $\left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\log^d n}$ and m with $\frac{1}{\delta(w,C_m)}$ (we assume without loss of generality that $\frac{1}{\delta(w,C_m)}$ is an integer because otherwise we could use $\left\lceil \frac{1}{\delta(w,C_m)} \right\rceil$).

We conclude that $\frac{1}{2} < \frac{1}{\delta(w,C_m)} \cdot \left(1 - \left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\log^d n}\right)$ and

$$\frac{\delta(w,C_m)}{2} < 1 - \left(1 - \frac{\delta(w,C_m)}{\log^d n}\right)^{\log^d n}.$$

□

References

[1] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509, 1992.

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

[3] Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos A. Kiwi, and Madhu Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory*, 42(6):1781–1795, 1996.

[4] Mihir Bellare and Madhu Sudan. Improved non-approximability results. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing (STOC), 23-25 May 1994, Montréal, Québec, Canada*, pages 184–193. ACM, 1994.

[7] The fact is true since $1 - p^l = (1-p) \cdot (1 + p + p^2 + \ldots + p^{l-1}) \leq (1-p) \cdot l$ for $p \leq 1$ and $l \in \mathbb{N}^+$.

[5] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding. *SIAM Journal on Computing*, 36(4):889–974, 2006.

[6] Eli Ben-Sasson, Oded Goldreich, and Madhu Sudan. Bounds on 2-Query Codeword Testing. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 2764 of *Lecture Notes in Computer Science*, pages 216–227. Springer, 2003.

[7] Eli Ben-Sasson, Venkatesan Guruswami, Tali Kaufman, Madhu Sudan, and Michael Viderman. Locally Testable Codes Require Redundant Testers. *SIAM J. Comput*, 39(7):3230–3247, 2010.

[8] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF Properties Are Hard to Test. *SIAM Journal on Computing*, 35(1):1–21, 2005.

[9] Eli Ben-Sasson and Madhu Sudan. Simple PCPs with poly-log rate and query complexity. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC), Baltimore, MD, USA, May 22-24, 2005*, pages 266–275. ACM, 2005.

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

[11] Eli Ben-Sasson and Michael Viderman. Composition of Semi-LTCs by Two-Wise Tensor Products. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 5687 of *Lecture Notes in Computer Science*, pages 378–391. Springer, 2009.

[12] Eli Ben-Sasson and Michael Viderman. Tensor Products of Weakly Smooth Codes are Robust. *Theory of Computing*, 5(1):239–255, 2009.

[13] Eli Ben-Sasson and Michael Viderman. Low rate is insufficient for local testability. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 6302 of *Lecture Notes in Computer Science*, pages 420–433. Springer, 2010.

[14] Eli Ben-Sasson and Michael Viderman. Towards lower bounds on locally testable codes via density arguments. *Computational Complexity*, 21(2):267–309, 2012.

[15] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, December 1993.

[16] Don Coppersmith and Atri Rudra. On the Robust Testability of Product of Codes. *Electronic Colloquium on Computational Complexity (ECCC)*, (104), 2005.

[17] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12:1–12:44, June 2007.

[18] Irit Dinur and Omer Reingold. Assignment Testers: Towards a Combinatorial Proof of the PCP Theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006.

[19] Irit Dinur, Madhu Sudan, and Avi Wigderson. Robust Local Testability of Tensor Products of LDPC Codes. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 4110 of *Lecture Notes in Computer Science*, pages 304–315. Springer, 2006.

[20] Zeev Dvir and Amir Shpilka. Locally Decodable Codes with Two Queries and Polynomial Identity Testing for Depth 3 Circuits. *SIAM J. Comput*, 36(5):1404–1434, 2007.

[21] Oded Goldreich. Short Locally Testable Codes and Proofs (Survey). *Electronic Colloquium on Computational Complexity (ECCC)*, (014), 2005.

[22] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006.

[23] Oded Goldreich and Or Meir. The tensor product of two good codes is not necessarily robustly testable. *Inf. Process. Lett*, 112(8-9):351–355, 2012.

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

[25] Tali Kaufman, Simon Litsyn, and Ning Xie. Breaking the epsilon-soundness bound of the linearity test over GF(2). *SIAM J. Comput*, 39(5):1988–2003, 2010.

[26] Tali Kaufman and Michael Viderman. Locally Testable vs. Locally Decodable Codes. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 6302 of *Lecture Notes in Computer Science*, pages 670–682. Springer, 2010.

[27] Or Meir. Combinatorial Construction of Locally Testable Codes. *SIAM J. Comput*, 39(2):491–544, 2009.

[28] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC), 1997*, pages 475–484. ACM, 1997.

[29] Shubhangi Saraf and Sergey Yekhanin. Noisy Interpolation of Sparse Polynomials, and Applications. In *IEEE Conference on Computational Complexity*, pages 86–92. IEEE Computer Society, 2011.

[30] Luca Trevisan. Some Applications of Coding Theory in Computational Complexity, September 23 2004.

[31] Paul Valiant. The Tensor Product of Two Codes Is Not Necessarily Robustly Testable. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 3624 of *Lecture Notes in Computer Science*, pages 472–481. Springer, 2005.

[32] Michael Viderman. A combination of testability and decodability by tensor products. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 7408 of *Lecture Notes in Computer Science*, pages 651–662. Springer, 2012.

A Repetition for strong LTCs and COLTCs

In this section we prove Proposition A.3 showing that a repetition of codeword entries preserve the local testability. Similar statements appeared in [13] for the case of weak LTCs. For the sake of completeness, we provide the proofs in this section, while modifying them to fit for strong LTCs and COLTCs.

For $S \subseteq [m]$ and $i \in \mathbb{N}$ let $S + i = i + S = \{s + i \mid s \in S\}$. For $w \in F^m$ and $t \in \mathbb{N}$ let $w^{(t)} \in \mathbb{F}^{mt}$ be the concatenation of w to itself t times. I.e., for every $i \in \{0, 1, \ldots, t-1\}$ we have $w^{(t)}|_{([m]+(i \cdot m))} = w$. To do this we first define projected testers.

Definition A.1 (Repetition Code)**.** Let $R \subseteq \mathbb{F}^m$ be a linear code and $t > 0$. We say that $C \subseteq \mathbb{F}^{(im)}$ is the *t-repetition* of R if every codeword of C is a codeword of R repeated t times. Formally, $c \in C$ if and only if $c = r^{(t)}$ for some $r \in R$.

If $A(R)$ is a core of R, then $A(C)$ is a core for C such that $C|_{A(C)}$ is a t-repetition $R|_{A(R)}$. In particular, it follows that $C|_{-A(C)}$ is a t-repetition $R|_{-A(R)}$.

Notice that the linearity of R implies the linearity of its repetition. The following simple claim argues that the repetition does not affect the distance of the code.

Claim A.2. *Let $t \in \mathbb{N}^+$ and $R \subseteq \mathbb{F}^m$ be a linear code. If $C \subseteq \mathbb{F}^{(mt)}$ is a t-repetition of R then $\delta(C) = \delta(R)$. Similarly, we have $\delta(C|_{A(C)}) = \delta(R|_{A(R)})$ and $\delta(C|_{-A(C)}) = \delta(R|_{-A(R)})$.*

Now we state Proposition A.3.

Proposition A.3. *Let $R \subseteq \mathbb{F}^m$ be a linear code and $t > 0$ be an integer. Let $C \subseteq \mathbb{F}^{(mt)}$ be a (t)-repetition of R. Then,*

- *If R is a (q, ϵ)-strong LTC then C is a $(q, \epsilon/4)$-strong LTC. Moreover, if R is a (q, ϵ)-COLTC then C is a $(q, \epsilon/8)$-COLTC.*
- *If C is a (q, ϵ)-strong LTC then R is a (q, ϵ)-strong LTC. Moreover, if C is a (q, ϵ)-COLTC then R is a (q, ϵ)-COLTC.*

To prove Proposition A.3 we need to define *projected testers*. Let $R \subseteq \mathbb{F}^m$ be a linear code and $C \subseteq \mathbb{F}^{(mt)}$ be its t-repetition. Given a tester $\mathcal{T}_C$ for C we define its *projected tester* by the distribution obtained from picking tests $(I \bmod m)$ where $I \sim \mathcal{T}_C$.

Proof of Proposition A.3. Note that $C|_{[m]} = R$.

Part 1. Assume that R is a (q, ϵ)-strong LTC and let $\mathcal{T}_R$ be a (q, ϵ)-strong tester for R. We define the following tester $\mathcal{T}_C$ for C.

- Flip a coin
- **If** "heads,"
  - pick $j \in [m]$ and $\ell_1 \in [t-1]$ independently at random,
  - output $I = \{j, j + m \cdot \ell_1\}$ (note that $I \subseteq [mt]$);
- **Else** pick $I \sim \mathcal{T}_R$ and output I (note that $I \subseteq [m]$).

We argue that $\mathcal{T}_C$ is a $(q, \epsilon/4)$-strong tester for C. Let $w \in \mathbb{F}^{(mt)}$ be a word. If $\delta(w|_{[m]}, C|_{[m]}) = \delta(w|_{[m]}, R) \geq \delta(w, C)/2$ we are done, since

$$\mathop{\mathbf{Pr}}_{I\sim\mathcal{T}_C}[w|_I \notin C|_I] \geq \frac{1}{2} \cdot \mathop{\mathbf{Pr}}_{I\sim\mathcal{T}_R}\left[(w|_{[m]})|_I \notin R|_I\right] \geq \frac{\delta(w, C) \cdot \epsilon}{4}.$$

Otherwise we have $\delta(w|_{[m]}, C|_{[m]}) = \delta(w|_{[m]}, R) < \delta(w, C)/2$. But then

$$\mathop{\mathbf{E}}_{j\in[t-1]}\left[\delta(w|_{\{jm+1,\ldots,(j+1)m\}}, w|_{[m]})\right] \geq \delta(w, C) - \delta(w, C)/2 = \delta(w, C)/2.$$

Hence $\mathop{\mathbf{Pr}}_{I\sim\mathcal{T}_C}[w|_I \notin C|_I] \geq \frac{1}{2}\mathop{\mathbf{Pr}}_{j\in[m],\ell_1\in[t-1]}\left[w|_{\{j,j+m\cdot\ell_1\}} \notin C|_{\{j,j+m\cdot\ell_1\}}\right] \geq \delta(w, C)/4$. Thus C is a $(q, \epsilon/4)$-strong LTC.

Now, assume that R is a (q, ϵ)-COLTC and let $\mathcal{T}_R$ be a (q, ϵ)-COLTC tester for R. We define the following tester $\mathcal{T}_C$ for C.

- Flip a coin
- **If** "heads,"
  - pick random $r \in \{1, 2\}$
    - If $r = 1$ then check the repetition for $A(R)$: pick $j \in A(R)$ and $\ell_1 \in [t-1]$ independently at random,
    - Else ($r = 2$) then check the entire repetition: pick $j \in [m]$ and $\ell_1 \in [t-1]$ independently at random.
  - Output $I = \{j, j + m \cdot \ell_1\}$ (note that $I \subseteq [mt]$);
- **Else** pick $I \sim \mathcal{T}_R$ and output I (note that $I \subseteq [m]$).

We argue that $\mathcal{T}_C$ is a $(q, \epsilon/8)$-COLTC tester for C. Let $w \in \mathbb{F}^{(mt)}$ be a word. If $\delta_{A(R)}(w|_{[m]}, C|_{[m]}) = \delta_{A(R)}(w|_{[m]}, R) \geq \delta_{A(C)}(w, C)/2$ we are done, since

$$\mathop{\mathbf{Pr}}_{I\sim\mathcal{T}_C}[w|_I \notin C|_I] \geq \frac{1}{2} \cdot \mathop{\mathbf{Pr}}_{I\sim\mathcal{T}_R}\left[(w|_{[m]})|_I \notin R|_I\right] \geq \frac{\delta_{A(R)}(w|_{[m]}, R) \cdot \epsilon}{2} \geq \frac{\delta_{A(C)}(w, C) \cdot \epsilon}{4}.$$

Otherwise we have $\delta(w|_{[m]}, C|_{[m]}) = \delta(w|_{[m]}, R) < \delta_{A(C)}(w, C)/2$. But then

$$\mathop{\mathbf{E}}_{j\in[t-1]}\left[\delta(w|_{jm+A(R)}, w|_{A(R)})\right] \geq \delta_{A(C)}(w, C) - \delta_{A(C)}(w, C)/2 = \delta_{A(C)}(w, C)/2.$$

or

$$\mathop{\mathbf{E}}_{j\in[t-1]}\left[\delta(w|_{jm+[m]}, w|_{[m]})\right] \geq \delta_{A(C)}(w, C) - \delta_{A(C)}(w, C)/2 = \delta_{A(C)}(w, C)/2.$$

Assume without loss of generality the first case. Then,

$$\mathop{\mathbf{Pr}}_{I\sim\mathcal{T}_C}[w|_I \notin C|_I] \geq \frac{1}{2} \cdot \frac{1}{2} \cdot \mathop{\mathbf{Pr}}_{j\in A(R),\ell_1\in[t-1]}\left[w|_{\{j,j+m\cdot\ell_1\}} \notin C|_{\{j,j+m\cdot\ell_1\}}\right] \geq \delta_{A(C)}(w, C)/8.$$

Thus C is a $(q, \epsilon/8)$-COLTC.

This completes the proof of the first part and now we deal with the second part.

Part 2. Assume that C is a (q,ϵ)-strong LTC and let $\mathcal{T}_C$ be its (q,ϵ)-strong tester. Let $\mathcal{T}_R$ be a projected tester of $\mathcal{T}_C$. Note that $\mathcal{T}_R$ is a distribution over subsets $I \subseteq [m]$ such that $|I| \leq q$.

We argue that $\mathcal{T}_R$ is a (q,ϵ)-strong tester for R. Let $w \in \mathbb{F}^m$ be a word. Assume by way of contradiction that $\underset{I\sim\mathcal{T}_R}{\mathbf{Pr}}[w|_I \notin R|_I] < \epsilon \cdot \delta(w,R)$. Notice that $\delta(w^{(t)},C) = \delta(w,R)$. We have $\underset{I\sim\mathcal{T}_C}{\mathbf{Pr}}\left[w^{(t)}|_I \notin C|_I\right] < \epsilon \cdot \delta(w,R)$ since if for $I \subset [tm]$ it holds that $w^{(t)}|_I \notin C|_I$ then $w|_{I \bmod m} \notin R$. We conclude that $\mathcal{T}_C$ is not a (q,ϵ)-strong tester for C. Contradiction.

The case, where C is a (q,ϵ)-COLTC, has almost the same proof. Let $\mathcal{T}_C$ be the (q,ϵ)-COLTC tester for C. Let $\mathcal{T}_R$ be a projected tester of $\mathcal{T}_C$. Fix any $w \in \mathbb{F}^m$ and note that $\delta_{A(C)}(w^{(t)},C) = \delta_{A(R)}(w,R)$, and as was said, for $I \subset [tm]$ it holds that $w^{(tm)}|_I \notin C|_I$ then $w|_{I \bmod m} \notin R$. So, the rejection probability of $\mathcal{T}_R$ on w is at least the rejection probability of $\mathcal{T}_C$ on $w^{(t)}$. We conclude that $\mathcal{T}_R$ is the (q,ϵ)-COLTC tester for R. □

B Weak LTCs vs. strong LTCs

In this section we show that some weak LTCs are not strong LTCs (Proposition B.1), but all strong LTCs are weak LTCs (Claim B.4). Although these claims are folklore in the area of LTCs, we do not aware of a formal proof of Proposition B.1 in the literature. Hence we state and prove Proposition B.1 and Claim B.4 in Sections B.1 and B.2, respectively.

B.1 Some weak LTCs are not strong LTCs

In this section we state and prove Proposition B.1.

Proposition B.1. *There exists a binary linear code $C \subseteq \mathbb{F}_2^{n+1}$ for arbitrary large n such that $\delta(C) = \Omega(1)$, C is a $(3,\epsilon,\rho)$-weak LTC for constants $\rho = \delta(C)/10$ and $\epsilon > 0$, but C is not a (q',ϵ')-strong LTC for any constants $q',\epsilon' > 0$.*

To prove Proposition B.1 we need to state and prove two auxiliary claims.

We start by recalling one of the results in the work of Ben-Sasson et al. [8] and refer a reader to [7] for the detailed explanation of this result and its corollaries. The work [8] showed that without loss of generality a q-query tester for a linear code C associated with a distribution $\mathcal{D}$ over $C^\perp_{\leq q}$. This tester picks $u \in_{\mathcal{D}} C^\perp_{\leq q}$ and accepts the input word w if and only if $\langle w,u\rangle = 0$. Since $|\operatorname{supp}(u)| \leq q$ the tester needs to query w in at most q coordinates in order to compute $\langle w,u\rangle$. Hence the result of Ben-Sasson et al. [8] gives us the following claim.

Claim B.2. *Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code. If $\operatorname{span}(\mathcal{C}^\perp_{\leq q}) \neq \mathcal{C}^\perp$ then $\mathcal{C}$ is not a (q,ϵ)-strong LTC for any $\epsilon > 0$.*

Proof. The fact that $\operatorname{span}(\mathcal{C}^\perp_{\leq q}) \neq \mathcal{C}^\perp$ implies the existence of $u \in \mathcal{C}^\perp$ such that $u \notin \operatorname{span}(\mathcal{C}^\perp_{\leq q})$. Thus there exists a word $w \in \mathbb{F}^n$ such that $\langle u',w\rangle = 0$ for all $u' \in \mathcal{C}^\perp_{\leq q}$ but $\langle u,w\rangle \neq 0$. So, $w \notin \mathcal{C}$ and $\delta(w,\mathcal{C}) > 0$.

Assume by contradiction that $\mathcal{C}$ is a (q,ϵ)-strong LTC for some $\epsilon > 0$. Then $\mathcal{C}$ has a corresponding tester T.

However, the work of Ben-Sasson et al. [8] implies that any tester for $\mathcal{C}$ picks a dual codeword of weight at most q and always accepts when the inner product between the input word and the

selected dual codeword is 0. Thus any tester for C will accept the word w with probability 1, i.e., the rejection probability is 0. However, the tester T is supposed to reject the word w with probability at least $\epsilon \cdot \delta(w, \mathcal{C}) > 0$. Contradiction. □

Now we recall the following claim of Kaufman and Viderman [26].

Claim B.3 ([26])**.** *Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code such that $\dim(\mathcal{C}) = \omega(\log(n))$. Then there exists $w \in \mathbb{F}^n$ such that $\Delta(w, \mathcal{C}^\perp) = \omega(1)$.*

Proof. For integer $R \in \mathbb{N}$ let $V(n, R) = \sum_{i=0}^{R} \binom{n}{i} \cdot (|\mathbb{F}|-1)^i$ be the volume of a sphere in $\mathbb{F}^n$ of radius R. Let $k = \dim(C) = \omega(\log(n))$ and $S = C^\perp$. Then $\dim(S) = n - k$ and $|S| = |\mathbb{F}|^{n-k} = |\mathbb{F}|^n/|\mathbb{F}|^k$. Recall that a covering radius of a code S is $R_S = \max_{w \in \mathbb{F}^n} \Delta(w, S)$, i.e., the largest Hamming distance of any word in $\mathbb{F}^n$ from S. Note that if R_S is constant then $V(n, R_S)$ is polynomial in n and vice versa, if $V(n, R_S)$ is super-polynomial in n then R_S goes to infinity with n. Assume by a way of contradiction that there exists a constant $t > 0$ such that for all $w \in \mathbb{F}^n$ we have $\Delta(w, S) \leq t$, i.e., $R_S \leq t = O(1)$.

The covering radius bound[8] states that

$$|S| \cdot V(n, R_S) \geq |\mathbb{F}|^n.$$

But then $V(n, R_S) \geq |\mathbb{F}|^k$, where $k = \omega(\log(n))$. Hence $V(n, R_S)$ must be super-polynomial in n, and $R_S = \omega(1)$. Contradiction. □

We are ready to prove Proposition B.1.

Proof of Proposition B.1. Let $\mathcal{C}' \subseteq \mathbb{F}_2^n$ be a binary linear code such that C' is a $(3, \epsilon')$-strong LTC for constant $\epsilon' > 0$, $\dim(\mathcal{C}) = \omega(\log n)$ and $\Delta(\mathcal{C}) = \Omega(n)$ (such codes exist, e.g., [24]). Let $\mathcal{T}'$ be the corresponding tester for $\mathcal{C}'$. Claim B.3 implies that there exists $u \in \mathbb{F}^n$ such that $\Delta(u, (\mathcal{C}')^\perp) = \omega(1)$.

Let $\mathcal{C} \subseteq F_2^{n+1}$ be a binary linear code such that $\mathcal{C}|_{[n]} = \mathcal{C}'$ and for every $c \in \mathcal{C}$ we have $c_{(n+1)} = \langle u, c|_{[n]} \rangle$, i.e., the first n coordinates of the code $\mathcal{C}$ are identical to the code $\mathcal{C}$ and the last bit of the code $\mathcal{C}$ is a sum of the bits indexed by $\mathrm{supp}(u)$. Note that $\Delta(\mathcal{C}) \geq \Delta(\mathcal{C}')$ and $\delta(\mathcal{C}) \geq \frac{n}{n+1} \cdot \delta(C) \geq 0.99\delta(C)$ for $n \geq 99$.

Let $q', \epsilon' > 0$ be any constants and assume by contradiction that $\mathcal{C}$ is a (q', ϵ')-strong LTC. Let $v \in \mathbb{F}_2^{n+1}$ be such that $\mathrm{supp}(v) = \mathrm{supp}(u) \cap \{n+1\}$. Note that since the underlying field $\mathbb{F}_2$ is the binary field, the vector v is defined. Notice that by construction it holds that $v \in \mathcal{C}^\perp$ and for every $v' \in \mathcal{C}^\perp$ we have $\Delta(v, v') = \omega(1) > q'$. It follows that $v' \notin \mathrm{span}(\mathcal{C}^\perp_{\leq q'})$,. i.e, $\mathrm{span}(\mathcal{C}^\perp_{\leq q'}) \neq \mathcal{C}^\perp$. Then by Claim B.2 we conclude that $\mathcal{C}$ is not (q', ϵ')-strong LTC. Contradiction.

However, $\mathcal{C}$ is a $(3, \epsilon' \cdot \delta(\mathcal{C})/11, \delta(\mathcal{C})/10)$-weak LTC. To see this, let $\mathcal{T}$ be the tester for $\mathcal{C}$ that on the input word $w \in \mathbb{F}_2^{n+1}$ invokes the tester $\mathcal{T}'$ on $w|_{[n]}$, and accepts if and only if $\mathcal{T}'$ accepted. Clearly, $\mathcal{T}$ makes at most 3 queries as $\mathcal{T}'$. If $w \in \mathcal{C}$ then $\mathcal{T}$ accepts with probability 1, since $w|_{[n]} \in \mathcal{C}'$ and thus $\mathcal{T}'$ accepts $w|_{[n]}$ with probability 1. Finally, if $\delta(w, \mathcal{C}) \geq \delta(\mathcal{C})/10$ then $\delta(w|_{[n]}, \mathcal{C}') \geq \delta(\mathcal{C})/11$. Thus the tester $\mathcal{T}'$ rejects $w|_{[n]}$ with probability at least $\epsilon' \cdot \delta(\mathcal{C})/11$. That means the tester $\mathcal{T}$ rejects w with probability at least $\epsilon \cdot \delta(\mathcal{C})/11$.

We conclude that $\mathcal{C}$ is a $(3, \epsilon, \rho)$-weak LTC, where $\epsilon = \epsilon' \cdot \delta(\mathcal{C})/11$ and $\rho = \delta(\mathcal{C})/10$. □

[8] For any code $\mathcal{C} \subseteq \mathbb{F}^n$ (whether linear or not) the covering bound states that the covering radius R of $\mathcal{C}$ relates to n and $|\mathcal{C}|$ by $|\mathcal{C}| \cdot V(n, R) \geq |\mathbb{F}|^n$.

B.2 All strong LTCs are weak LTCs

Claim B.4. *If a code $C \subseteq \mathbb{F}_2^n$ is a (q, ϵ)-strong LTC then C is a $(q, \epsilon \cdot \rho, \rho)$-weak LTC for every $\rho > 0$.*

Proof. The claim holds since there exists a q-query tester T that always accepts all $w \in C$ and rejects all $w \notin C$ with probability at least $\epsilon \cdot \delta(w, C)$. In particular, if $\delta(w, C) \geq \rho$ then T rejects w with probability at least $\epsilon \cdot \rho$. □

ECCC ISSN 1433-8092

http://eccc.hpi-web.de