Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 87 (2011)

A Combination of Testability and Decodability by Tensor Products

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

November 6, 2011

Abstract

Ben-Sasson and Sudan (RSA 2006) showed that repeated tensor products of linear codes with a very large distance are locally testable. Due to the requirement of a very large distance the associated tensor products could be applied only over sufficiently large fields. Then Meir (SICOMP 2009) used this result (as a black box) to present a combinatorial construction of locally testable codes that match best known parameters. As a consequence, this construction was obtained over sufficiently large fields.

In this paper we improve the result of Ben-Sasson and Sudan and show that for *any* linear codes the associated tensor products are locally testable. Consequently, the construction of Meir can be taken over any field, including the binary field.

Moreover, a combination of our result with the result of Spielman (IEEE IT, 1996) implies a construction of linear codes (over any field) that combine the following properties:

- have constant rate and constant relative distance;
- have blocklength n and testable with n^{ϵ} queries, for any constant $\epsilon > 0$;
- linear time encodable and linear-time decodable from a constant fraction of errors.

Furthermore, a combination of our result with the result of Guruswami et al. (STOC 2009) implies a similar corollary regarding the list-decodable codes.

1 Introduction

Over the last decades coding theory and complexity theory have benefited from numerous interesting interconnections. Recent major achievements in complexity theory, e.g., showing IP = PSPACE [39, 40, 34] and giving PCP characterization of NP [4, 3] have strongly relied on connections with coding theory either explicitly or implicitly.

Most of the well-studied and practically used codes are linear codes. A linear code $C \subseteq \mathbf{F}^n$ is a linear subspace, where n is a called the blocklength of C and $\dim(C)$ denotes the dimension of the code. The rate of the code is defined by $\mathrm{rate}(C) = \frac{\dim(C)}{n}$. The distance of the code C, denoted by $\Delta(C)$, is the minimal hamming distance between two different codewords of C. Typically, one is interested in the codes whose distance is linear to the blocklength.

[*] The research was partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 240258 and by grant number 2006104 by the US-Israel Binational Science Foundation.

ISSN 1433-8092

The central algorithmic problem in coding theory is the explicit construction of error-correcting codes with best possible parameters together with fast encoding and decoding algorithms. I.e., given a message $w \in \mathbf{F}^k$ the goal is efficiently encode this message (in the best case we have linear running time), and given a corrupted codeword the goal is efficiently decode it and obtain the original message (again, in the best case this can be done in the linear time). These features were proved to be useful also in the complexity theory.

Besides the efficient encoding/decoding algorithms we know the following well-studied properties: local testing and local decoding (correction). The combination of these properties is highly useful, e.g., PCPs based on the Hadamard code [3] relied on the fact that the Hadamard code is testable with 3 queries [31] and locally decodable (correctable) with 2 queries.

Given the fact that error-correcting codes play an important role in the complexity theory, and in particular, in different iterative protocols (see e.g., [6]), it might be helpful to develop a general scheme for constructing the error-correcting codes that combine several different properties. E.g., it might be helpful to have high-rate codes which combine such properties as local testing, efficient encoding and decoding from a constant fraction of errors. This is what we do in this paper. In the rest of the introduction we provide a brief background and explain our contribution.

Locally Testable Codes. Locally testable codes (LTCs) are error correcting codes that have a tester, which is a randomized algorithm with oracle access to the received word x. The tester reads a sublinear amount of information from x and based on this "local view" decides if $x \in C$ or not. It should accept codewords with probability one, and reject words that are far (in Hamming distance) from the code with noticeable probability.

Such codes are of interest in computer science due to their numerous connections to probabilistically checkable proofs (PCPs) and property testing (see the surveys [43, 20] for more information). LTCs were implicit already in [5] (cf. [20, Sec. 2.4]) and they were explicitly studied by Goldreich and Sudan [22]. By now several different constructions of LTCs are known including codes based on low-degree polynomials over finite fields [2, 31, 3], constructions based on PCPs of proximity/assignment testers [7, 17][1] and sparse random linear codes [13, 27, 29]. In this paper we study a different family of LTC constructions, namely, *tensor codes*. Given two linear error correcting codes $C \subseteq \mathbf{F}^{n_1}$, $R \subseteq \mathbf{F}^{n_2}$ over a finite field $\mathbf{F}$, we define their *tensor product* to be the subspace $R \otimes C \subseteq \mathbf{F}^{n_1 \times n_2}$ consisting of $n_1 \times n_2$ matrices M with entries in $\mathbf{F}$ having the property that every row of M is a codeword of R and every column of M is a codeword of C. In this case, we say that C and R are *base-codes*. If $C = R$ we use C^2 to denote $C \otimes C$ and for $i > 2$ define $C^i = C \otimes C^{i-1}$. Note that the blocklength of C^i is n_1^i.

Recently, tensor products were used to construct new families of LTCs [10, 33], new families of list-decodable codes [24], to give an alternative proof [34][2] for IP=PSPACE theorem of [39, 40] etc.

Ben-Sasson and Sudan [10] suggested to use tensor product codes as a means to construct LTCs combinatorially. Let $C \subseteq \mathbf{F}^{n_1}$ be a linear code and let us consider the following approach. Suppose that the task is to test whether an input word $M \in \mathbf{F}^{n_1 \times n_1}$ belongs to C^2, where M is far from C^2. One could expect that in this case the typical row/column of M is far from C, and hence the tester for C^2 can choose a random row (or column) of M. Then this selected row/column could be tested on being in C. However, as was shown in [44, 21, 15] this approach fails in general and is known to work only under assumptions that C has some non-trivial properties [18, 11, 12] (see also [32]).

[1] As was pointed out in [22], not all PCP constructions are known to yield LTCs, but some of them (e.g., PCPs of proximity/assignment testers) can be adapted to yield LTCs.

[2] Meir [34] showed that the "multiplication" property and the "sum-check" protocol can be designed by tensor products. We consider this surprising, since previously such features were achieved only by low degree polynomials.

In spite of this fact, Ben-Sasson and Sudan [10] showed that taking the repeated tensor products of any code $C \subseteq \mathbf{F}^n$ with sufficiently large distance results in a locally testable code with sublinear query complexity. Although it was not explicitly stated in [10], it follows that [10, Theorem 2.6] gives the following result.

Theorem 1.1 (Informal). *For every $\epsilon > 0$ there exists a sufficiently large field $\mathbf{F} = \mathbf{F}(\epsilon)$ such that letting $m = \lceil \frac{2}{\epsilon} \rceil$ for every $C \subseteq \mathbf{F}^n$, $\left(\frac{\Delta(C)-1}{n}\right)^m \geq \frac{7}{8}$ it holds that C^m is testable with N^ϵ queries, where $N = n^m$ is the blocklength of C^m.*

Let us explain some issues that remained open. First of all, it was remained unclear if the assumption about a very large distance of the base codes is necessary. Moreover, the requirement on the distance of the base code $(\Delta(C))$ is dependent on the number of tensor products (m) one should apply. Note that less query complexity (relatively to the blocklength) one should get more tensor product operations should be applied. Thus the requirement about the distance of the base code is increased when the number of queries one should get is decreased. We notice also that the requirement for larger $\Delta(C)$ implies the larger underlying field $\mathbf{F}$. As a consequence, a similar theorem could not be argued for some fixed field.

In this paper we ask the following question: is it possible to achieve a similar result to [10] but with no requirements about the base codes at all? A positive result to this question might seem surprising since it would imply that *any* linear error-correcting code can be involved in the construction of LTCs via tensor products. Nevertheless, we give a positive answer on this question and show that no assumptions about the base codes (or underlying fields) are needed. I.e., informally, we show the following result (stated formally in Theorem 3.1).

Theorem 1.2 (Informal). *For every $\epsilon > 0$ and for every field $\mathbf{F}$ letting $m = \lceil \frac{2}{\epsilon} \rceil$ it holds that for every $C \subseteq \mathbf{F}^n$ we know that C^m is testable with N^ϵ queries, where $N = n^m$ is the blocklength of C^m. The rejection probability of the tester is proportional to $\Delta(C)$.*

This contrasts with the previous works on the combinatorial constructions of LTCs due to Ben-Sasson and Sudan [10] and Meir [33] which required very large base-code distance, and as a consequence required the large field size. Moreover, the constructions of best known LTCs [9, 16, 33] were obtained over the large fields (when finally, the field size can be decreased through code concatenation). Our improvement over the result of [10] implies that the construction of Meir [33] (which achieves LTCs of best known parameters[3]) can be taken directly over any field (including the binary field). Thus our results imply that LTCs having the best known parameters can be constructed directly over any field. On the other hand, we think that this improvement has a non-negligible role since the LTCs construction of Meir [33] is combinatorial and the combinatorial constructions of LTCs (or PCPs) should be independent, as much as possible, of the algebraic terms such as “polynomials”, “fields” etc. Furthermore, our proof is much simpler than the proof provided in [10] and simultaneously we obtain some quantitative improvements in the related parameters (see Section A and in particular Remark A.6).

Efficient encoding and decoding. Let us ask the following natural question. Whether tensor products of codes can be encoded efficiently ? It is quite simple to show (Claim 3.2) that if the code C has an efficient (linear time) encoder then C^m has an efficient (linear time) encoder.

[3] The best known LTCs achieve constant relative distance, inverse poly-logarithmic rate, constant query complexity and constant rejection probability. This range of parameters was initially achieved by Dinur [16] based on the results of Ben-Sasson and Sudan [9]. Later, Meir [33], based on [10], gave a combinatorial construction of LTCs that matched the same range of parameters.

Let us turn to the decoding properties of the tensor products, e.g., the natural question here would be whether tensor products of codes preserve the decoding properties provided that the base codes are efficiently decodable. This question was studied by Gopalan et al. [24] who showed that tensor products preserve the list-decoding properties, i.e., if C is list-decodable in polynomial time then C^m is list-decodable in polynomial time.[4] Our contribution to this question is as follows. We show (Proposition 3.6) that if C is decodable from a constant fraction of errors in linear time then C^m is decodable from a constant fraction of errors in linear time.

Then, we show (Corollaries 3.10, 3.15) that a combination of our results with the results of [42, 24] implies the construction of constant-rate codes which are both testable with sublinear query complexity, linear-time encodable and efficiently decodable (or list-decodable) from the constant fraction of errors.

Tensor product of codes preserves the local correction properties. Informally, locally correctable codes (LCCs) are error-correcting codes that allow to retrieve each codeword bit using a small number number of queries even after a constant fraction of it is adversely corrupted. The most famous LCCs include Hadamard and Reed-Muller codes. Recently, Kopparty et al. [30] presented a new family of LCCs called *multiplicity codes*.

In Section 3.3 we prove that tensor product of codes preserve the local correction property. That means if C is an LCC with query complexity q then C^2 is an LCC with query complexity q^2. On the one hand, this observation discovers additional families of locally correctable codes and on the other hand, it suggests a simple way to combine two different properties: local correction and local testing. E.g., let $C \subseteq \mathbf{F}^n$ be a linear LCC with query complexity q and let $C' = C^{10} \subseteq \mathbf{F}^{n^{10}}$. Then C' has blocklength $N = n^{10}$, C' is an LCC with query complexity q^{10} and is an LTC (with query complexity $N^{0.2}$).

Organization of the paper. In the following section we provide background regarding tensor codes and locally testable codes. In Section 3 we state our main results. We state our main technical theorem (Theorem A.5) in Section A. The proof of Theorem A.5 is postponed to Section B and the proofs of auxiliary statements appear in Section C.

2 Preliminaries

Throughout this paper, $\mathbf{F}$ is a finite field, $[n]$ denotes the set $\{1, \ldots, n\}$ and $\mathbf{F}^n$ denotes $\mathbf{F}^{[n]}$. All codes discussed in this paper will be a linear. Let $C \subseteq \mathbf{F}^n$ be a linear code over $\mathbf{F}$.

For $w \in \mathbf{F}^n$ let $\mathrm{supp}(w) = \{i | w_i \neq 0\}$, $|w| = |\mathrm{supp}(w)|$ and $\mathrm{wt}(w) = \frac{|w|}{n}$. We define the *distance* between two words $x, y \in \mathbf{F}^n$ 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 a code is defined by $\Delta(C) = \min_{x \neq y \in C} \Delta(x, y)$ and its the relative distance is denoted $\delta(C) = \frac{\Delta(C)}{n}$. A $[n, k, d]_{\mathbf{F}}$-code is a k-dimensional subspace $C \subseteq \mathbf{F}^n$ of distance d. The rate of the code C is defined by $\mathrm{rate}(C) = \frac{\dim(C)}{n}$. For $x \in \mathbf{F}^n$ and $C \subseteq \mathbf{F}^n$, let $\delta(x, C) = \delta_C(x) = \min_{y \in C} \{\delta(x, y)\}$ to denote the relative distance of x from the code C. We note that $\Delta(C) = \min_{c \in C \setminus \{0\}} \{\mathrm{wt}(c)\}$. If $\delta(x, C) \geq \epsilon$ we say that x is ϵ-far from C and otherwise x is ϵ-close to C. We let $\dim(C)$ denote the dimension of C. The vector inner product between $u = (u_1, u_2, \ldots, u_n) \in \mathbf{F}^n$ and $v = (v_1, v_2, \ldots, v_n) \in \mathbf{F}^n$ is defined to be $\langle u, v \rangle = \sum_{i \in [n]} u_i \cdot v_i$. We let $C^{\perp} = \{u \in \mathbf{F}^n \mid \forall c \in C : \langle u, c \rangle = 0\}$ be the dual code of C and

[4] The main focus in [24] was done on the designing polynomial-time list-decoding algorithms and on the combinatorial bounds for the list-decoding tensor products of codes and interleaved codes.

$C_t^\perp = \left\{u \in C^\perp \mid |u| = t\right\}$. In a similar way we define $C_{\le t}^\perp = \left\{u \in C^\perp \mid |u| \le t\right\}$. For $t \in \mathbf{F}^n$ and $T \subseteq \mathbf{F}^n$ we say that $t \perp T$ if $\langle t, t' \rangle = 0$ for all $t' \in T$.

For $w \in F^n$ and $S = \{j_1, j_2, \ldots, j_m\} \subseteq [n]$, where $j_1 < j_2 < \ldots < j_m$, we let $w|_S = (w_{j_1}, \ldots, w_{j_m})$ be the *restriction* of w to the subset S. We let $C|_S = \{c|_S \mid c \in C\}$ denote the restriction of the code C to the subset S.

2.1 Tensor Product Codes

The definitions appearing here are standard in the literature on tensor-based LTCs (e. g. [18, 10, 33, 12, 44]).

For $x \in \mathbf{F}^I$ and $y \in \mathbf{F}^J$ we let $x \otimes y$ denote the tensor product of x and y (i. e., the matrix M with entries $M_{(i,j)} = x_i \cdot y_j$ where $(i, j) \in I \times J$). Let $R \subseteq \mathbf{F}^I$ and $C \subseteq \mathbf{F}^J$ be linear codes. We define the tensor product code $R \otimes C$ to be the linear space spanned by words $r \otimes c \in \mathbf{F}^{I \times J}$ for $r \in R$ and $c \in C$. Some known facts regarding the tensor products (see e. g., [18]):

- The code $R \otimes C$ consists of all $I \times J$ matrices over $\mathbf{F}$ whose rows belong to R and whose 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 $C^1 = C$ and $C^t = C^{t-1} \otimes C$ for $t > 1$. Note by this definition, $C^{2^0} = C$ and $C^{2^t} = C^{2^{t-1}} \otimes C^{2^{t-1}}$ for $t > 0$. We also notice that for a code $C \subseteq \mathbf{F}^n$ and $m \ge 1$ it holds that $\mathrm{rate}(C^m) = (\mathrm{rate}(C))^m$, $\delta(C^m) = (\delta(C))^m$ and the blocklength of C^m is n^m.

The main drawback of the tensor product operation is that this operation strongly decreases the rate and the distance of the base codes. We refer the reader to [33] which showed how one can use tensor products and avoid the decrease in the distance and the strong decrease in the rate.[5]

2.2 Locally testable codes and Robustly Testable Codes

A *standard q-query tester* for a linear code $C \subseteq \mathbf{F}^n$ is a randomized algorithm that on the input word $w \in \mathbf{F}^n$ picks non-adaptively a subset $I \subseteq [n]$ such that $|I| \le q$. Then T reads all symbols of $w|_I$ and accepts if $w|_I \in C|_I$, 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 $|I| \le q$.

Definition 2.1 (Tester of C and Test View)**.** A *q-query tester* $\mathbf{D}$ is a distribution $\mathbf{D}$ over subsets $I \subseteq [n]$ such that $|I| \le q$. Let $w \in \mathbf{F}^n$ (think of the task of testing whether $w \in C$) and let $I \subseteq [n]$ be a subset. We call $w|_I$ the *view* of a tester. If $w|_I \in C|_I$ we say that this view is *consistent* with C, or when C is clear from the context we simply say $w|_I$ is *consistent*.

Although the tester in Definition 2.1 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 $w \in C$ and

[5] Meir [33] demonstrated how one can combine the tensor product operation with two additional operations: random projections and distance amplification. In this way, on the one hand repeated tensor products could be applied, while on the other hand these supplementary operations prevent the distance loss and the strong rate reduction.

a subset $I \subseteq [n]$ is selected by the tester, the tester can output accept if $w|_I \in C|_I$ and otherwise output reject.

When considering a tensor code $C^m \subseteq \mathbf{F}^{n^m}$, an associated tester will be a distribution over subsets $I \subseteq [n]^m$.

Definition 2.2 (LTCs and strong LTCs)**.** A code $C \subseteq \mathbf{F}^n$ is a (q, ϵ, δ)-LTC if it has a q-query tester $\mathbf{D}$ such that for all $w \in \mathbf{F}^n$, if $\delta(w, C) \geq \delta$ we have $\mathop{\mathbf{Pr}}_{I \sim \mathbf{D}}[w|_I \notin C|_I] \geq \epsilon$.

A code $C \subseteq \mathbf{F}^n$ is a (q, ϵ)-strong LTC if it has a q-query tester $\mathbf{D}$ such that for all $w \in \mathbf{F}^n$, we have $\mathop{\mathbf{Pr}}_{I \sim \mathbf{D}}[w|_I \notin C|_I] \geq \epsilon \cdot \delta(w, C)$.

We notice that a (q, ϵ)-strong LTC is a $(q, \epsilon\delta, \delta)$-LTC for every $\delta > 0$. Note that given a code $C \subseteq \mathbf{F}^n$, the subset $I \subseteq [n]$ uniquely defines $C|_I$. Moreover, the linearity of C implies that $C|_I$ is a linear subspace of $\mathbf{F}^I$.

3 Main Results

We start this section by presenting our main theorem. Theorem 3.1 implies that *any linear code* over any field can be used to define a locally testable code with sublinear query complexity.

Theorem 3.1 (Main Theorem)**.** *Let $C \subseteq \mathbf{F}^n$ be a linear code and $m \geq 3$ is a constant. Then C^m is a (n^2, α_m)-strong LTC, where $\alpha_m > 0$ is a constant that depends only on m and $\delta(C)$. Note that the blocklength of C^m is n^m.*

The proof-sketch of Theorem 3.1 appears in Section C.1 and is based on the main technical theorem (Theorem A.5). We continue to the "encoding" property of tensor products. Claim 3.2 shows that if a linear code C is linear-time encodable then so is C^i for any constant i. Later we will use this claim together with Theorem 3.1 to show Corollary 3.3.

Claim 3.2. *Let $m \geq 1$ be a constant. If $C \subseteq \mathbf{F}^n$ is a linear-time encodable linear code then C^m is linear-time encodable.*

The proof of Claim 3.2 is postponed to Section C.2. Note that every linear code can be encoded is quadratic time (multiplication by a generator matrix). Now, we combine Theorem 3.1 and Claim 3.2 to show a simple construction of strong LTCs with arbitrary small sublinear query complexity and arbitrary high rate from any linear code with sufficiently high rate.

Corollary 3.3. *Let $\mathbf{F}$ be any field. Let $C \subseteq \mathbf{F}^n$ be a linear code and let $m \geq 3$ be a constant. Then $C^m \subseteq \mathbf{F}^{n^m}$ is a (n^2, α_m)-strong LTC, where $\alpha_m > 0$ is a constant that depends only on m and $\delta(C)$. In particular, for every $\epsilon > 0$, $m = \lceil \frac{2}{\epsilon} \rceil$, $N = n^m$ and $C \subseteq \mathbf{F}^n$ such that $\mathrm{rate}(C) \geq (1-\epsilon)^{1/m}$ we have $C^m \subseteq \mathbf{F}^N$ is a (N^ϵ, α)-strong LTC and $\mathrm{rate}(C^m) \geq 1 - \epsilon$, where $\alpha > 0$ is a constant that depends only on ϵ. Moreover, if C is a linear-time encodable then C^m is a linear-time encodable.*

Remark 3.4. We notice that there are linear error-correcting codes with arbitrary high rate that can be encodable in the linear time (see e.g., [38][6]). Thus Corollary 3.3 provides a construction of high-rate LTCs with constant relative distance and arbitrary low sublinear query complexity that can be encoded in linear time. Moreover, this construction can be taken over any field.

[6] This result improves the previous result of [26] and presents the construction of linear codes that lie close to the singleton bound, and have linear time encoding/decoding algorithms.

We also notice that any simple approach, based on the testing of (low-degree) polynomials [2], to achieve the similar result to Corollary 3.3 fails shortly. In particular, let us consider the testing of Reed-Muller codes of degree d and recall that informally, Reed-Muller codes of degree d can be tested by making $\approx 2^d$ queries. If d is large then the associated codes must be constructed over the very large field (depending on the blocklength of the code), since otherwise cannot have constant relative distance. However, if d is small then the rate of the associated code is very low. Furthermore, the *linear-time* encoding of the codes based on high-degree polynomials is problematic.

Usually, in the areas of locally testable and locally decodable codes the main interest was given to the constant query complexity. Recently, Kopparty et al. [30] showed the construction of high-rate locally decodable codes with sublinear query complexity (see [30] for the motivation behind this range of parameters). Since then, the interest to the other range of parameters, and in particular, to sublinear query complexity was increased.

We would like to stress that Corollary 3.3 is quite powerful for this range of parameters (sublinear query complexity and high rate). First of all, there are different constructions of linear-time encodable and decodable codes with constant rate and constant relative distance [25, 26, 42], and them all can be involved to define high-rate LTCs with sublinear query complexity that are linear-time encodable. The other advantage of such constructions is that the repeated tensor products of the base code are known to inherit some properties of the base codes besides local testability. E.g., Gopalan et al. [24] showed that the tensor product operation preserves list-decodability properties.

In Section 3.1 we show how the local testing with sublinear query complexity can be combined with the linear-time encoding and decoding. Then, in Section 3.2 we show that Corollary 3.3 can be combined with the result of [24] to define asymptotically good codes that can be encodable in linear time, testable with sublinear query complexity and list-decodable in polynomial time.

Tensor Products of Codes can have nice distance. As was said in Section 2.1, Meir [33] explained that one of the standard procedures for distance amplification of the code [1] can be combined together with the repeated tensor product operations. He also proved that this procedure preserves the local testability of the underlying code. The simplest way to see this is as follows. Let $\mathrm{DistAmp}(\cdot)$ be a procedure that increases the relative distance of the code $C' \subseteq \mathbf{F}_2^n$, e.g., from 0.001 to 0.49. I.e., if $\delta(C') \geq 0.001$ then $\delta(\mathrm{DistAmp}(C')) \geq 0.49$. Moreover, it holds that if C' was locally testable then $\mathrm{DistAmp}(C')$ is locally testable, where the query complexity of the code $\mathrm{DistAmp}(C')$ is increased by only a constant factor, independent on the other parameters of the code). It can be readily verified that the distance amplification procedure preserves the encoding time, and in particular, if C' was linear-time encodable then $\mathrm{DistAmp}(C')$ is linear-time encodable. Thus, one can pick any linear-time encodable code C with linear distance, obtain a linear-time encodable LTC $C' = C^{10}$ and then increase its distance by $\mathrm{DistAmp}(C')$. We refer the reader to [33, Section 4.3] for further information about distance amplification procedures and its affect on local testability.

In this paper we won't use any distance amplification procedures and restrict our attention only to the tensor product operation.

3.1 Locally testable and Linear-time decodable codes

Let us first recall the definition of decodable codes.

Definition 3.5. Let $C \subseteq \mathbf{F}^n$ be a code and let $\alpha < \delta(C)/2$. We say that C is decodable from αn errors in time T if there exists a decoder D_C which on the input word $w \in \mathbf{F}^n$ such that $\delta(w, C) \leq \alpha$ outputs $c \in C$

such that $\delta(w, c) \leq \alpha$ and its runtime is upper-bounded by T. If $T = O(n)$ we say that C is decodable in linear time.

Proposition 3.6 shows that the tensor product operation preserves the "unique-neighbor" decoding property. In particular, if C is a linear code that is linear time decodable from a constant fraction of errors then so is C^2. Hence this observation, together with a result of, e.g. [42], can result in the construction of asymptotically good locally testable codes with sublinear query complexity that can be linear-time encoded and decoded to the closest codeword after a constant fraction of errors.

Proposition 3.6. *Assume $C \subseteq \mathbf{F}^n$ is a linear code that is linear-time decodable from $\alpha \cdot n$ errors. Let $m \geq 1$ be a fixed constant. Then C^m is a linear code that is linear-time decodable from $\alpha^m \cdot n^m$ errors.*

The proof of Proposition 3.6 is postponed to Section C.3. Spielman [42] (based on [41]) was first who provided the (explicit) construction of linear codes that can be encoded in linear time and decoded in linear time from the constant fraction of errors. The construction of these codes was achieved over the binary field, but it can be easily extended to any other field as well.

Theorem 3.7 ([42])**.** *There exists an (explicit) family of linear error correcting codes $C \subseteq \mathbf{F}_2^n$ such that $\mathrm{rate}(C) = \Omega(1)$, $\delta(C) = \Omega(1)$, C is a linear-time encodable and linear-time decodable from the constant fraction of errors.*

A combination of Theorem 3.7, Proposition 3.6 and Claim 3.2 together with Theorem 3.1 results in the following corollary. Note that it considers binary linear codes.

Corollary 3.8. *For every constant $\epsilon > 0$ there exists an (explicit) family of linear error correcting codes $C \subseteq \mathbf{F}_2^N$ (obtained by tensor products on the codes from Theorem 3.7) that*

- *have rate and relative distance $\Omega_\epsilon(1)$,*
- *linear time encodable and linear time decodable from the constant fraction ($\Omega_\epsilon(1)$) of errors,*
- *are (N^ϵ, α)-strong LTCs, where $\alpha = \alpha(\epsilon) > 0$ is a constant.*

Proof. Let $\epsilon > 0$ and $m = \lceil \frac{2}{\epsilon} \rceil$. Let $C' \subseteq \mathbf{F}_2^n$ be a code from Theorem 3.7 such that $\mathrm{rate}(C') = \Omega(1)$, $\delta(C') = \Omega(1)$ and C' is linear-time encodable and decodable from the constant fraction of errors. Let $C = (C')^m$ and note that the blocklength of C is $N = n^m$. It follows that $\mathrm{rate}(C) = (\mathrm{rate}(C'))^m = \Omega_m(1)$ and $\delta(C) = (\delta(C'))^m = \Omega_m(1)$. Moreover, Claim 3.2 and Proposition 3.6 imply that C is encodable in linear time and decodable from the constant fraction ($\Omega_\epsilon(1)$) of errors in linear time. By Theorem 3.1 it holds that C is a (N^ϵ, α)-strong LTC, where α is a constant that depends on ϵ. □

The work of Spielman [42] was improved later (e.g., [26, 38]) and in particular, the construction of codes with arbitrary high rate was achieved over the fields of constant size.[7]

The next theorem is due to Guruswami and Indyk [26].[8]

Theorem 3.9 ([26])**.** *For every $\epsilon > 0$ there exist a field $\mathbf{F} = \mathbf{F}(\epsilon)$ and an (explicit) family of linear error correcting codes $C \subseteq \mathbf{F}^n$ such that $\mathrm{rate}(C) \geq 1 - \epsilon$, $\delta(C) = \Omega_\epsilon(1)$, C is a linear-time encodable and linear-time decodable from the constant fraction ($\Omega_\epsilon(1)$) of errors, where $\mathbf{F}$ is a field of constant size (independent of the blocklength).*

[7] The code constructions suggested in [26, 38] correct a larger fraction of errors than in [42], and even almost optimal given the distance parameter. However, for our result (Corollary 3.10) it is sufficient to say that the codes are decodable from the *constant fraction* of errors.

[8] The Theorem was improved later by Roth and Skachek [38].

Note that the underlying field $\mathbf{F}$ has a constant size that depends only on the parameter ϵ. Again, a combination of Theorem 3.7, Proposition 3.6 and Claim 3.2 together with Theorem 3.1 results in Corollary 3.10. Note that Corollaries 3.8 and 3.10 present a construction of error-correcting codes that combine different non-trivial and useful properties. The difference between these two corollaries is in the binary field versus a larger field and the constant rate versus arbitrary high rate.

Corollary 3.10. *For every constant $\epsilon > 0$ there exist a field $\mathbf{F}$ and an (explicit) family of linear error correcting codes $C \subseteq \mathbf{F}^N$ (obtained by tensor products on the codes from Theorem 3.9) that*

- *have rate at least $1 - \epsilon$ and relative distance $\Omega_\epsilon(1)$,*
- *linear time encodable and linear time decodable from the constant fraction ($\Omega_{\epsilon(1)}$) of errors,*
- *are (N^ϵ, α)-strong LTCs, where $\alpha = \alpha(\epsilon) > 0$ is a constant.*

Proof. The proof of this corollary is very similar to the proof of Corollary 3.8 with the difference that the base code is taken from Theorem 3.9.

Let $\epsilon > 0$ and $m = \lceil \frac{2}{\epsilon} \rceil$. Let $C' \subseteq \mathbf{F}^n$ be a code from Theorem 3.9 such that $\mathrm{rate}(C') \geq (1-\epsilon)^{1/m}$, and recall that $\delta(C') = \Omega_\epsilon(1)$, C' is linear-time encodable and decodable from the constant fraction of errors. Let $C = (C')^m$ and note that the blocklength of C is $N = n^m$. It follows that $\mathrm{rate}(C) = (\mathrm{rate}(C'))^m \geq 1 - \epsilon$ and $\delta(C) = (\delta(C'))^m = \Omega_\epsilon(1)$. Moreover, Claim 3.2 and Proposition 3.6 imply that C is encodable in linear time and decodable from the constant fraction of errors in linear time. By Theorem 3.1 it holds that C is a (N^ϵ, α)-strong LTC, where α is a constant that depends on ϵ. □

3.2 Locally testable and List-decodable codes

In this section we recall some constructions of the list-decodable codes. We start by defining the list-decodable codes.

Definition 3.11 (List-decodable codes)**.** A code C is a (α, L)-list decodable if for every word $w \in \mathbf{F}^n$, $\delta(w, C) \leq \alpha$ we have $|\{c \in C \mid \delta(c, w) \leq \alpha\}| \leq L$. The code is said to be (α, L)-list decodable in time T if there exists algorithm which on the input $w \in \mathbf{F}^n$ such that $\delta(w, C) \leq \alpha$ outputs all codewords $c \in C$ such that $\delta(c, w) \leq \alpha$ (at most L codewords).

Guruswami et al. [24] showed that the list-decodability is preserved in the tensor product operation. More formally, they showed the following theorem stated in [24, Theorem 5.7].

Theorem 3.12 ([24])**.** *Let $\mathbf{F}$ be a finite field and $q = |\mathbf{F}|$. Given two linear codes $C_1, C_2 \subseteq \mathbf{F}^n$, for every $\epsilon > 0$, the number of codewords of $C_2 \otimes C_1$ within distance $\eta^* = \min(\delta_1\eta_2, \delta_2\eta_1) - 3\epsilon$ of any received word is bounded by $l(C_2 \otimes C_1, \eta^*) \leq 4q^{\frac{1}{4\delta_1^2\epsilon^2}\ln\frac{8l_1(\eta_1)}{\epsilon}\ln\frac{8l_2(\eta_2)}{\epsilon}}$.*

Further, if C_1 and C_2 can be list decoded in polynomial time up to error rates η_1, η_2 and C_2 is a linear code, then $C_2 \otimes C_1$ can be list decoded in polynomial time up to error rate η^*. Specifically, if T denotes the time complexity of list decoding C_1 and C_2, then the running time of the list decoding algorithm for $C_2 \otimes C_1$ is $O(4q^{\frac{1}{4\delta_1^2\epsilon^2}\ln\frac{8l_1(\eta_1)}{\epsilon}\ln\frac{8l_2(\eta_2)}{\epsilon}} \cdot Tn_1n_2)$.

Then, Gopalan et al. used Theorem 3.12 to conclude the following theorem, appearing in [24, Theorem 5.8].

Theorem 3.13 ([24]). *Let C be a linear code with distance δ, list decodable up to an error rate η. For every $\delta > 0$, the m-wise tensor product code C^m can be list decoded up to an error rate $\delta^{m-1}\eta - \epsilon$ with a list size $\exp((O(\frac{\ln l(\eta)/\epsilon}{\epsilon^2}))^m)$. Moreover, if $m \geq 1$ is constant and C is polynomial-time list decodable then the runtime of the list decoding algorithm for C^m is polynomial (depending on m).*

The next fact is known due to the several constructions of list-decodable codes ([25, 37]).

Fact 3.14. There exist linear error-correcting codes (over any field) of constant rate and constant relative distance that can be encoded in linear time and list-decoded in polynomial time.

We use the combination of Theorem 3.13, Fact 3.14, Claim 3.2 and Corollary 3.3 to conclude Corollary 3.15 which shows that the tensor products can be used to combined local testability and list-decoding properties.

Corollary 3.15. *Let $\mathbf{F}$ be any field. For every constant $\epsilon > 0$ there exists a code $C \subseteq \mathbf{F}^N$ such that $C = (C')^{(\lceil 2/\epsilon \rceil)}$, where $C' \subseteq \mathbf{F}^n$ is a linear code, $\mathrm{rate}(C') = \Omega(1)$, $\delta(C') = \Omega(1)$ and C' is (ρ, L)-list decodable in polynomial time ($\rho, L > 0$ are constants).*

- *C is a (N^ϵ, α)-strong LTC, where $\alpha = \alpha(\epsilon) > 0$ is a constant,*
- *C is linear time encodable and list-decodable (constant list size) in polynomial time from the constant fraction of errors (depending on ϵ),*
- *$\mathrm{rate}(C) = \Omega_\epsilon(1)$ and $\delta(C) = \Omega_\epsilon(1)$.*

Proof. The proof of Corollary 3.15 is similar to the proof of Corollary 3.10. The base-code C' can be taken any linear code that match the requirements written in Fact 3.14. Then, letting $m = \lceil \frac{2}{\epsilon} \rceil$ it holds that $C = (C')^m$ is the required code due to Claim 3.2,Theorem 3.13 and Theorem 3.1. □

3.3 Tensor Products preserve Local Correction properties

Informally, locally decodable codes (LDCs) allow to recover each message entry with high probability by reading only a few entries of the codeword even if a constant fraction of it is adversely corrupted. These codes are related to private information retrieval protocols, initiated by [14]. The best construction of LDCs was initiated by the breakthrough results of Yekhanin [45] who showed a (conditional) subexponential construction of 3-query LDCs. Later Efremenko [19] showed unconditional subexponential construction of LDCs. Gopalan [23] showed that these codes can be considered as a sub-family of Reed-Muller codes. On the other hand, locally correctable codes (LCCs) are error-correcting codes that allow to retrieve each codeword bit using a small number number of queries even after a constant fraction of it is adversely corrupted. The well-known LCCs include the Hadamard and Reed-Muller codes. Recently, Kopparty et al. [30] presented a new family of LCCs with constant rate and sublinear number of queries.

In this section we explain that the tensor product of codes preserves the local decoding (correction) properties as well as local testability. Although it is not hard to see that the repeated tensoring preserves these properties, this fact seems to have remained unnoticed.

Let us start from the formal definition of LCCs.

Definition 3.16 (LCCs). Let $\mathcal{C} \subseteq \mathbf{F}^n$ be a code. Then $\mathcal{C}$ is a (q, ϵ, δ)-LCC if there exists a self-corrector ($\mathbb{SC}$) that reads at most q entries and the following condition holds:

- For all $c \in \mathcal{C}$, $i \in [n]$ and $\hat{c} \in F^n$ such that $\Delta(c, \hat{c}) \leq \delta n$ we have $\mathbf{Pr}\left[\mathbb{SC}^{\hat{c}}[i] = c_i\right] \geq 1 - \epsilon$, i.e., with probability at least $1 - \epsilon$ entry c_i will be recovered correctly.

The parameter q is known as the query complexity, ϵ is the error probability of the self-corrector and δ is the distance threshold.

Sometimes, the LCCs are defined using the requirement $\mathbf{Pr}\left[\mathbb{SC}^{\hat{c}}[i] = c_i\right] \geq \dfrac{1}{|\mathbf{F}|} + \epsilon$ to stress that the success probability of the self-corrector should be higher than trivial $(1/|\mathbf{F}|)$. In this paper, we use the requirement $\mathbf{Pr}\left[\mathbb{SC}^{\hat{c}}[i] = c_i\right] \geq 1 - \epsilon$ to treat the ϵ as the error probability.

While the definition of LCCs may seem similar to the definition of LDCs these types of codes are different, and as was pointed out, e.g., in [28] every LCC is an LDC with the same parameters but some LDCs are not LCCs. The standard range of parameters for an LCC is related to $q, \epsilon, \delta > 0$ are constants.

Remark 3.17. We stress that the error probability ϵ may be arbitrary small when the distance threshold δ converges to 0. For example, the well-known Hadamard code is a $(2, \epsilon = 2\delta, \delta)$-LCC and hence ϵ can be picked as arbitrary small constant.

The following proposition shows that the tensor product preserves the local correction property. It can be readily verified that a similar statement holds for the locally decodable codes for the case of systematic codes (the codes whose first codeword entries are message symbols)[9].

Proposition 3.18. *Let $C \subseteq \mathbf{F}^n$ be a code. If C is a (q, ϵ, δ)-LCC then $C^2 = C \otimes C$ is a $(q^2, (q+1) \cdot \epsilon), \delta^2)$-LCC.*

The proof of Proposition 3.18 appears in Section C.4. The following corollary summarizes the effect of repeated tensoring on an LCC. We recall that as was said in Remark 3.17, the error probability ϵ can be arbitrary small when δ is taken arbitrary small.

Corollary 3.19. *Let $C \subseteq \mathbf{F}^n$ be a code and $i \geq 1$ be an integer. If C is a (q, ϵ, δ)-LCC such that $(\Pi_{j=1}^{j=i}(q^{2^{j-1}} + 1))\epsilon < \frac{|\mathbf{F}|-1}{|\mathbf{F}|}$ then $C^{2^i} \subseteq \mathbf{F}^{n^{2^i}}$ is a $(q^{2^i}, (\Pi_{j=1}^{j=i}(q^{2^{j-1}} + 1))\epsilon, \delta^{2^i})$-LCC.*

In Corollary 3.19 we required that the error probability of the obtained LCCs will be below $\frac{|\mathbf{F}|-1}{|\mathbf{F}|}$, which is trivial. In this way, the LCCs can be involved in the tensor products resulting in the code which are both locally testable and locally correctable, similarly to the combination of local testability and efficient decodability (see Section 3.1).

Acknowledgements

The author thanks Eli Ben-Sasson for many invaluable discussions about the “robustness” concept and the possible connections to the work [36], mentioned in Remark A.7. We would like to thank Or Meir for helpful discussions. The author thanks Ronny Roth for pointers to the literature.

We thank the anonymous referees for valuable comments on an earlier version of this article.

[9] Any linear code can be viewed as systematic by picking an appropriate generator matrix.

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] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing reed-muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005.

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

[4] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.

[5] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proc. 23rd STOC*, pages 21–31. ACM, 1991.

[6] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[7] 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.

[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 *STOC*, 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 Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim, editors, *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 Maria J. Serna, Ronen Shaltiel, Klaus Jansen, and José D. P. Rolim, editors, *APPROX-RANDOM*, volume 6302 of *Lecture Notes in Computer Science*, pages 420–433. Springer, 2010.

[14] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. *JACM: Journal of the ACM*, 45, 1998.

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

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

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

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

[19] Klim Efremenko. 3-query locally decodable codes of subexponential length. In Michael Mitzenmacher, editor, *STOC*, pages 39–44. ACM, 2009.

[20] Oded Goldreich. Short locally testable codes and proofs (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, (014), 2005.

[21] Oded Goldreich and Or Meir. The Tensor Product of Two Good Codes Is Not Necessarily Robustly Testable. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(062), 2007.

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

[23] Parikshit Gopalan. A note on efremenko's locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, (069), 2009.

[24] Parikshit Gopalan, Venkatesan Guruswami, and Prasad Raghavendra. List decoding tensor products and interleaved codes. In Michael Mitzenmacher, editor, *STOC*, pages 13–22. ACM, 2009.

[25] Venkatesan Guruswami and Piotr Indyk. Linear time encodable and list decodable codes. In *STOC*, pages 126–135. ACM, 2003.

[26] Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005.

[27] Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *FOCS*, pages 590–600. IEEE Computer Society, 2007.

[28] Tali Kaufman and Michael Viderman. Locally Testable vs. Locally Decodable Codes. In Maria J. Serna, Ronen Shaltiel, Klaus Jansen, and José D. P. Rolim, editors, *APPROX-RANDOM*, volume 6302 of *Lecture Notes in Computer Science*, pages 670–682. Springer, 2010.

[29] Swastik Kopparty and Shubhangi Saraf. Local list-decoding and testing of random linear codes from high error. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 417–426. ACM, 2010.

[30] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. In *ECCC - TR10-148*, 2010.

[31] M.Blum, M.Luby, and R.Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *JCSS: Journal of Computer and System Sciences*, 47, 1993.

[32] Or Meir. On the rectangle method in proofs of robustness of tensor products. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(061), 2007.

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

[34] Or Meir. IP = PSPACE using Error Correcting Codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:137, 2010.

[35] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5), 2010.

[36] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *STOC*, pages 475–484, 1997.

[37] Ron M. Roth. *Introduction to coding theory*. Cambridge University Press, 2006.

[38] Ron M. Roth and Vitaly Skachek. Improved Nearly-MDS Expander Codes. *IEEE Transactions on Information Theory*, 52(8):3650–3661, 2006.

[39] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[40] A. Shen. IP = PSPACE: Simplified proof. *J. ACM*, 39(4):878–880, 1992.

[41] Michael Sipser and Daniel A. Spielman. Expander Codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996. Preliminary version appeared in FOCS 1994.

[42] Daniel A. Spielman. Linear-time Encodable and Decodable Error-Correcting Codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996. Preliminary version appeared in STOC 1995.

[43] Luca Trevisan. Some Applications of Coding Theory in Computational Complexity. In *ECCC: Electronic Colloquium on Computational Complexity, technical reports*, 2004.

[44] Paul Valiant. The Tensor Product of Two Codes Is Not Necessarily Robustly Testable. In *APPROX-RANDOM*, volume 3624 of *Lecture Notes in Computer Science*, pages 472–481. Springer, 2005.

[45] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1), 2008.

A Main Technical Theorem — Theorem A.5

We start this section by defining the notion of *robustness* (Definition A.2) as was introduced in [10]. To do that we start from the definition of *local distance* (Definition A.1), which will be used in Definition A.2 and later in our proofs.

Definition A.1 (Local distance)**.** Let C be a code and $w|_I$ be the view on the coordinate set I obtained from the word w. The *local distance* of w from C with respect to I (also called the I-distance of w from C) is $\Delta\left(w|_I, C|_I\right) = \min_{c \in C} \left\{\Delta\left(w|_I, c|_I\right)\right\}$ and similarly the *relative local distance* of w from C with respect to I (relative I-distance of w from C) is $\delta(w|_I, C|_I) = \min_{c \in C} \left\{\delta(w|_I, c|_I)\right\}$.

Informally, robustness implies that if a word is far from the code then, on average, a test's view is far from any consistent view that can be accepted on the same coordinate set I. This notion was defined for LTCs following an analogous definition for PCPs [7, 16]. We are ready to provide a general definition of robustness.

Definition A.2 (Robustness)**.** Given a tester (i. e., a distribution) $\mathbf{D}$ for the code $C \subseteq \mathbf{F}^n$, 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 $w \in \mathbf{F}^n$ it holds that $\rho^{\mathbf{D}}(w) \geq \rho^{\mathbf{D}}(C) \cdot \delta_C(w)$.

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 $\rho^{\mathbf{D_n}}(C_n) \geq \alpha$.

In the rest of the section we consider the "hyperplane tester" defined in the work of Ben-Sasson and Sudan [10], which generalized in some sense the work of Raz and Safra [36]. To do this let us define two auxiliary notations: points and hyperplanes. A point in m-dimensional cube can be associated with an m-tuple $(i_1, i_2, ..., i_m)$ such that $i_j \in [n]$. We say that τ is a (b, i)-hyperplane if

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

Definition A.3 (Hyperplane Tester)**.** Let $m \geq 3$. Let $M \in \mathbf{F}^{n^m}$ be an input word and think of testing whether $M \in C^{n^m}$. 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 $M|_{(b,i)}$). Note that if M is a candidate word to be in C^m then $M|_{(b,i)}$ is a candidate word to be in C^{m-1}.

For the first reading we suggest to think about the binary field $\mathbf{F} = \mathbf{F}_2$ and $m = 3$, and look on the matrix $M \in \mathbf{F}^{n^m}$ as on the boolean 3-dimensional cube. Throughout this paper we assume that $m \geq 3$ and for the case of $m = 2$ we refer a reader to [11, 12, 18, 44, 21, 15, 15]).

Now we state the main result of Ben-Sasson and Sudan [10].

Theorem A.4 ([10])**.** *Let $C \subseteq \mathbf{F}^n$ be a linear code and $m \geq 3$. Let $\mathcal{D}$ be the hyperplane tester for C^m. If $\left(\frac{\Delta(C)-1}{n}\right)^m \geq \frac{7}{8}$ then*

$$\rho^{\mathcal{D}}(C^m) \geq 2^{-16}.$$

Now we state our main technical theorem which says that the tensor product of any base code (with constant relative distance) is robustly testable. This extends the result of [10] (Theorem A.4) which showed that this claim holds for base codes with a very large distance.[10]

Theorem A.5 (Main Technical Theorem)**.** *Let $C \subseteq \mathbf{F}^n$ be a linear code and $m \geq 3$. Let $\mathcal{D}$ be the hyperplane tester for C^m. Then*

$$\rho^{\mathcal{D}}(C^m) \geq \frac{(\delta(C))^m}{2m^2}.$$

[10] We notice that a similar requirement for the very large distance/field was done in the work of Raz and Safra [36], although due to the different reasons.

The proof of Theorem A.5 is postponed to Section B. Ben-Sasson and Sudan [10] explained that hyperplane testers can be composed and the robustness of the hyperplane testers implies the strong local testability. So, Theorem A.5 is our main step to conclude Theorem 3.1, and we provide a proof-sketch in Section C.1 (see [10, 11] for more information about composition of the testers).[11]

Remark A.6. First, we note that Theorem A.5 can be extended straightforward to the tensor products of different linear base codes, i.e, $C_1 \otimes C_2 \otimes \ldots \otimes C_m$, where the codes C_j might have different blocklength.

We also note that $\rho^{\mathcal{D}}(C^m)$ in Theorem A.5 is lower-bounded by the expression depending on m. Thus one could think that for the "large" values of m this bound becomes bad. This issue can be easily solved using the next observation: $C^m = C^{m_1} \otimes C^{m_2} \otimes C^{m_3}$, where $m_1 + m_2 + m_3 = m$. E.g., C^{10} can be viewed as $C^3 \otimes C^3 \otimes C^4$, which is a 3-wise tensor product of $C_1 = C^3$, $C_2 = C^3$ and $C_3 = C^4$, i.e., $m = 3$. In this case, one can work with $m = 3$ such that the hyperplane tester selects the local views (hyperplanes) that will be tested recursively on the membership to $C^{m_i} \otimes C^{m_j}$.

Finally, we note that Theorem A.5 achieves quantitative improvement versus Theorem A.4. E.g., taking $m = 3$ and $(\delta(C))^m \geq 7/8$ (as required by [10]) Theorem A.4 guarantees that $\rho^{\mathcal{D}}(C^m) \geq 2^{-16} \approx 0.000015$, while our result (Theorem A.5) guarantees that $\rho^{\mathcal{D}}(C^m) \geq \frac{7}{8 \cdot 18} \approx 0.048611$. We also notice that our proof of Theorem A.5 is simpler than the proof of Theorem A.4 in [10].

Remark A.7. We would like to notice that our proof of Theorem A.5 might be very interesting with regards to the PCPs with subconstant error probability due to Raz and Safra [36], improved later in the breakthrough results of Moshkovitz and Raz [35]. One of the main ingredients in these works is a tight analysis related to the tester similar to the "hyperplane tester" in Definition A.3. The main difference was that in [36, 35] the underlying code was low-degree polynomial over the large field, and as a consequence one could select much more different "hyperplanes" than in the tensor product of general codes (see [36]).

Now, the intriguing detail in the proof of [36] was that the proof was achieved via reduction to the graphs and the analysis of the obtained graph. Ben-Sasson and Sudan [10] followed the proof-style (on the high level) of [36] and in particular, they also defined a reduction from the code and the hyperplanes to the graph and analyzed the graph. However, in our proof we do not go to the graphs and analyzed the code and the hyperplanes directly, and this is one of the reasons behind the improvement in Theorem A.5 versus Theorem A.4 of [10].

B Proof of Theorem A.5

Throughout this paper we assume that $C \subseteq \mathbf{F}^n$ is a linear code. We shall consider an m-wise tensor product, i.e., $C^m \subseteq \mathbf{F}^{n^m}$. Note that the blocklength of C^m is n^m. For simplicity we recommend to the reader to think about the case where $m = 3$, $\mathbf{F} = \mathbf{F}_2$ and then every word in $\mathbf{F}^{n^m}$ can be viewed as a boolean 3-dimensional cube.

We start this section by defining the concepts of points, lines and hyperplanes (some of the terms were defined following [10]).

B.1 Preliminary notations: Points, Lines and Hyperplanes

Recall that a point in the m-dimensional cube can be associated with an m-tuple $(i_1, i_2, \ldots, i_m)$ such that $i_j \in [n]$. Next we define the axis parallel line, or shortly, the line which can be associated with a subset of

[11] Given Theorem A.5, the proof of a similar statement to Theorem 3.1 can be found in [10]. For the sake of completeness we provide the proof-sketch for Theorem 3.1 in Section C.1.

points. 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

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

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. 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 $i_k = j_k$. Two (different) lines intersects on the point p if both lines contain the point p.

We recall that τ is a (b, i)-hyperplane if

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

A (b, i)-hyperplane contains the point $p = (i_1, i_2, \ldots, i_m)$ if $i_b = i$, i.e., the b-th coordinate of the point p is i. A (b, i)-hyperplane contains a line l if it contains all points of the line. We say that two (different) hyperplanes are intersected if both hyperplanes contain at least one common point. Note that two (different) hyperplanes: (b_1, i_1)-hyperplane and (b_2, i_2)-hyperplane are intersected iff $b_1 \neq b_2$, moreover, they are intersected on all points $p = (i_1, \ldots, i_m)$ such that $i_1 = i_{b_1}$ and $i_2 = i_{b_2}$, i.e., are intersected on n^{m-2} points.

Assume that τ_1 is a (b_1, i_1)-hyperplane and τ_2 is a (b_2, i_2)-hyperplane such that $b_1 < b_2$ (in particular $b_1 \neq b_2$). Let $\tau_1 \cap \tau_2 = \{(i_1, \ldots, i_m) \mid i_{b_1} = i_1, i_{b_2} = i_2\}$ be an intersection of two hyperplanes and $C^m|_{tau_1 \cap tau_2}$ be a code C^m restricted to the points in $\tau_1 \cap \tau_2$. Note that $\delta(C^m|_{tau_1 \cap tau_2}) = \delta(C^{m-2}) = \delta(C)^{m-2}$.

Given a word $M \in \mathbf{F}^{n^m}$, $b \in [m]$ and $i \in [n]$ we let $M|_{(b,i)}$ be a restriction of M to the (b, i)-hyperplane, i.e., to all points of the hyperplane. We say that $M|_{(b,i)}$ is a (b, i)-hyperplane of M. Similarly, for the point $p = (i_1, \ldots, i_m)$ 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 say that $M|_l$ is a line l of M.

B.2 The proof

Let $M \in \mathbf{F}^{n^m}$ be an input word. We prove that $\rho^{\mathcal{D}}(M) \geq \frac{(\delta(C))^{m-1}}{2m^2} \cdot \delta(M, C^m)$.

For every hyperplane τ of M let $r(\tau)$ be the closest codeword of C^{m-1} to $M|_\tau$ (if there are more than one such codewords fix any of them arbitrarily). Intuitively, the hyperplane τ of M "thinks" that the symbols of $M|_\tau$ should be changed to $r(\tau)$. In this sense every hyperplane of M has its own "opinion". Then we have

$$\rho^{\mathcal{D}}(M) = \mathop{\mathbf{E}}_{\tau \sim \mathcal{D}}[\delta(M|_\tau, r(\tau))]. \tag{B.1}$$

We say that the (b_1, i_1)-hyperplane and the (b_2, i_2)-hyperplane disagree on the point $p = (i_1, \ldots, i_m)$ if both hyperplanes contain the point p and $r(\tau_1)|_p \neq r(\tau_2)|_p$. We say that two hyperplanes disagree on the line l if both hyperplanes contain the line l and $r(\tau_1)|_l \neq r(\tau_2)|_l$.

Note that if (b_1, i_1)-hyperplane τ_1 and (b_2, i_2)-hyperplane τ_2 are intersected and disagree on at least one point then letting $reg = \tau_1 \cap \tau_2$ we have $r(\tau_1)|_{reg} \neq r(\tau_2)|_{reg}$ and moreover, $\delta(r(\tau_1)|_{reg}, r(\tau_2)|_{reg}) \geq (\delta(C))^{m-2}$. This is true since $r(\tau_1)|_{reg} \neq r(\tau_2)|_{reg} \in C^{m-2}$ and $\delta(C^{m-2}) = (\delta(C))^{m-2}$.

Let $E \in \mathbf{F}_2^{n^m}$ be a binary matrix such that $E|_p = 1$ if there are at least two hyperplanes which disagree on the point p, and otherwise $E|_p = 0$. For the point p we say that the point is almost fixed if $E|_p = 0$ but p is contained in some hyperplane τ such that $r(\tau)|_p \neq M|_p$. Intuitively, a point p is almost fixed if all hyperplanes containing this point agree on this point but "think" that its value in M ($M|_p$) should be changed (to $r(\tau)|_p$).

We let $ToFix = \{p = (i_1, i_2, \ldots, i_m) \mid p$ is almost fixed$\}$ and let $NumToFix = |ToFix|$. Recall that $\mathrm{wt}(E) = \frac{|E|}{n^m}$, i.e., $\mathrm{wt}(E)$ is the relative weight of the matrix E.

Proposition B.1. *It holds that* $\rho^{\mathcal{D}}(M) \geq \frac{\mathrm{wt}(E)}{m} + \frac{NumToFix}{n^m}$.

Proof. Equation B.1 says that $\rho^{\mathcal{D}}(M)$ is a relative distance of a typical hyperplane of M (which is a word in $\mathbf{F}^{n^{m-1}}$) from C^{m-1}. Note that for every point $p = (i_1, \ldots, i_m)$: if $E|_p \neq 0$ then $p \notin NeedToFix$. That means for every point p at most one condition is satisfied: $E|_p \neq 0$ or $p \in NeedToFix$.

Note also that for every point $p \in NeedToFix$, for all hyperplanes τ of M we have $(M|_\tau)|_p \neq r(\tau)|_p$. Now, every point p is contained in m different hyperplanes. Hence if $E|_p \neq 0$ then for at least one hyperplane τ (of m hyperplanes containing the point p) it holds that $r(\tau)|_p \neq M|_p$.

Hence a relative distance between a typical hyperplane (τ) of M and $r(\tau)$ is at least $\frac{\mathrm{wt}(E)}{m} + \frac{NumToFix}{n^m}$. □

Next we define an important concept of “heavy hyperplanes (lines)” in the inconsistency matrix E. Intuitively, a heavy hyperplane (line) of the matrix E is a plane (line) where many inconsistencies occur, i.e., many non-zero symbols.

Definition B.2 (Heavy lines and hyperplanes)**.** A line l of E is called heavy if $|E|_l| \geq \delta(C) \cdot n$. A (b, i)-hyperplane of E is called heavy if $|E|_{(b,i)}| \geq \frac{(\delta(C)\cdot n)^{m-1}}{2}$.

Lemma B.3 is our main observation in the proof of Theorem A.5. It says that any non-zero element of E is located in some heavy hyperplane of E. This lemma plays a crucial role since it gives us an understanding of how the inconsistent points of the input matrix are distributed. Again, as was pointed in Remark A.7, this lemma can be of independent interest due to the plausible connections to [36, 35].

Lemma B.3 (Main Lemma)**.** *Let $p = (i_1, i_2, \ldots, i_m)$ be a point such that $E_p \neq 0$. Then p is contained in some heavy hyperplane of E.*

The proof of Lemma B.3 is postponed to Section B.2.1. Using Lemma B.3 it is quite simple to prove Corollary B.4 which shows that it is sufficient to remove at most $\frac{2|E|}{(\delta(C)\cdot n)^{m-1}} \cdot m$ hyperplanes from E to get a zero submatrix.

Corollary B.4. *There exists $S_1, \ldots, S_m \subseteq [n]$ such that $n - |S_1| + n - |S_2| + \ldots + n - |S_m| \leq \frac{2|E|}{(\delta(C)\cdot n)^{m-1}} \cdot m$ and letting $S = S_1 \times S_2 \times \ldots \times S_m$ we have $E|_S = 0$.*

The proof of Corollary B.4 appears in Section B.2.2. Proposition B.5 says that if after removing a small fraction of hyperplanes from M we obtain a submatrix that is close to the legal submatrix then M is close to C^m.

Proposition B.5. *Let $S_1, S_2, \ldots, S_m \subseteq [n]$ be such that $n - |S_1| + n - |S_2| + \ldots + n - |S_m| \leq \gamma n < \delta(C) \cdot n$ and let $S = S_1 \times S_2 \times \ldots \times S_m$. Let $C' = C|_{S_1} \otimes C|_{S_2} \otimes \ldots \otimes C|_{S_m}$. Recall that $M|_S$ is a submatrix of M obtained by removing at most $\gamma \cdot n$ hyperplanes. Assume that $\Delta\left(M|_S, C'\right) \leq \alpha \cdot n^m$. Then $\delta(M, C^m) \leq \gamma + \alpha$.*

The proof of Proposition B.5 appears in Section B.2.3. Let us prove Theorem A.5.

Proof of Theorem A.5. By Proposition B.1 we have $\rho^{\mathcal{D}}(M) \geq \frac{\mathrm{wt}(E)}{m} + \frac{NumToFix}{n^m}$. If $\mathrm{wt}(E) \geq \frac{(\delta(C))^m}{2m}$ then we are done. Otherwise, assume that $\mathrm{wt}(E) < \frac{(\delta(C))^m}{2m}$.

Corollary B.4 implies that it is sufficient to remove at most $\frac{2|E|}{(\delta(C)n)^{m-1}} \cdot m < \delta(C) \cdot n$ hyperplanes from E to get a zero submatrix. Proposition B.5 implies that $\delta(M, C^m) \leq \frac{2\,\mathrm{wt}(E)}{(\delta(C))^{m-1}} \cdot m + \frac{NumToFix}{n^m}$.

Let $\beta = \frac{2m^2}{(\delta(C))^{m-1}}$. Then, by Proposition B.1 we have $\rho^{\mathcal{D}}(M) \cdot \beta \geq (\frac{\mathrm{wt}(E)}{m} + \frac{NumToFix}{n^m}) \cdot \beta \geq \delta(M, C^m)$ and $\rho^{\mathcal{D}}(M) \geq \frac{(\delta(C))^{m-1}}{2m^2} \cdot \delta(M, C^m)$. □

B.2.1 Proof of Main Lemma B.3

In this section we prove Lemma B.3.

Proof of Main Lemma B.3. By definition of E 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. We will prove that either τ_1 is a heavy hyperplane or τ_2 is a heavy hyperplane.

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 axis 3 and contains a 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 $\Delta\left(r(\tau_1)|_l, r(\tau_2)|_l\right) \geq \delta(C) \cdot n$, 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 $r(\tau)|_{\tau \cap \tau_1} \neq r(\tau_1)|_{\tau \cap \tau_1} \in C^{m-2}$.

Let $total = \{p = (i_1, j_2, \ldots, j_m) \mid \exists \tau \in BadPlanes_{\tau_1} \text{ s.t. } p \in p_1 \cap \tau, r(\tau)|_p \neq r(\tau_1)|_p\}$. 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). Thus $E|_{\tau_1}$ has at least $\frac{(\delta(C) \cdot n)^{m-1}}{2}$ non-zero symbols. We conclude that τ_1 is a heavy hyperplane of E and the point p is contained in the hyperplane τ_1. □

Remark B.6. We notice that the proof of Lemma B.3 shows even a stronger claim than it is needed. Namely, it shows that if two different hyperplanes τ_1 and τ_2 disagree on some point then at least one of them disagree with a lot of different non-intersecting hyperplanes, and as a consequence, is heavy. This lemma can be easily reformulated and shown for the low-degree test analysis in [36], and it remains an interesting question whether this will affect the work of [36].

B.2.2 Proof of Corollary B.4

Let us prove Corollary B.4.

Proof of Corollary B.4. Let $HeavyPlanes = \{(b, i) \mid (b, i)$ is a heavy hyperplane$\}$ to be a subset of pairs associated with heavy hyperplanes. For $b \in [m]$ let $\overline{S_b} = \{i \in [n] \mid (b, i) \in HeavyPlanes\}$ and $S_b = [n] \setminus \overline{S_b}$.

We claim that $|HeavyPlanes| \leq \frac{2|E|}{(\delta(C)\cdot n)^{m-1}} \cdot m$. This is true since every heavy hyperplane contains at least $\frac{(\delta(C)\cdot n)^{m-1}}{2}$ non-zero elements of E and the total number of non-zero elements of E is $|E|$. Furthermore, every non-zero element of E is contained in at most m (heavy) hyperplanes. Thus $n - |S_1| + n - |S_2| + \ldots + n - |S_m| = \sum_{b\in[m]} |\overline{S_b}| \leq \frac{2|E|}{(\delta(C)n)^{m-1}} \cdot m$.

Now, note that Lemma B.3 implies that every point $p = (i_1, i_2, \ldots, i_m)$ such that $E|_p \neq 0$ is contained in some heavy hyperplane, i.e., in some hyperplane of $HeavyPlanes$. Hence if all heavy hyperplanes are removed from E we obtain a zero submatrix. So, it follows that $E|_S = 0$. □

B.2.3 Proof of Proposition B.5

In this section we prove Proposition B.5.

Proof of Proposition B.5. Note that for every $i \in [n]$ we have $|S_i| > n - \delta(C) \cdot n$. The following simple claim was proven in [10, Proposition 3.1]. For the sake of completeness we provide its proof.

Every codeword c' of C' can be extended to a unique codeword c of C^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 to C' is bijection, because both codes are of dimension $(\dim(C))^m$. Thus, every word in C' has a unique preimage in C.

We turn to prove Proposition B.5. We know that M can be modified in at most α-fraction of points $p \in S$ to get $M|_S \in C'$. Then, by the claim above, M can be modified (outside the submatrix $M|_S$) to get a codeword of C^m, by changing at most γ-fraction of symbols (since all symbols outside the submatrix $M|_S$ are at most γ-fraction of all symbols). We conclude that $\delta(M, C^m) \leq \gamma + \alpha$. □

C Proofs of Auxiliaries Claims and Propositions

C.1 Proof Sketch of Theorem 3.1

Let us start from the following simple claim proved in [10]. For the sake of completeness we give its proof in this paper.

Claim C.1. *Let $C \subseteq \mathbf{F}^n$ be a code and assume that $\mathcal{D}$ is its q-query tester such that $\rho^{\mathcal{D}}(C) \geq \alpha$. Then C is a (q, α)-strong LTC.*

Proof. Recall that $\mathcal{D}$ can be associated to a distribution over subsets $I \subseteq [n]$ such that $|I| \leq q$. It is sufficient to prove that for every $w \in \mathbf{F}^n$ we have $\mathbf{Pr}_{I\sim\mathcal{D}}[w|_I \notin C|_I] \geq \alpha \cdot \delta(w, C)$.

Fix any $w \in \mathbf{F}^n$. Note that for $I \subseteq [n]$ if $w|_I \in C|_I$ then $\delta(w|_I, c|_I) = 0$ and if $w|_I \notin C|_I$ then $\delta(w|_I, c|_I) \leq 1$. Hence $\alpha \cdot \delta(w, C) \leq \rho^{\mathcal{D}}(C) \cdot \delta(w, C) \leq \mathbf{E}_{I\sim\mathcal{D}}[\delta(w|_I, C_I)] \leq \mathbf{Pr}_{I\sim\mathcal{D}}[w|_I \notin C|_I]$. □

Using Claim C.1 we prove Theorem 3.1.

Proof Sketch: For $i \geq 3$ let $\mathcal{D}_i$ be the hyperplane tester for the code C^i. Note that the tester $\mathcal{D}_m$ returns a local view that is a candidate to be in the code C^{m-1}. We first explain a simple way to compose the testers and then show how to improve this.

Note that $\mathcal{D}_{m-1}$ can be invoked on the local view of $\mathcal{D}_m$, etc. So, the testers $\mathcal{D}_m, \mathcal{D}_{m-1}, \ldots, \mathcal{D}_3$ can be composed to result in an n^2-query tester $\mathcal{D}_{comp}$ for the code C^m.

The robustness of the composed tester will be $\rho^{\mathcal{D}_{comp}}(C^m) \geq \rho^{\mathcal{D}_m}(C^m) \cdot \rho^{\mathcal{D}_{m-1}}(C^{m-1}) \cdot \ldots \cdot \rho^{\mathcal{D}_3}(C^3)$. To see this let $w \in \mathbf{F}^{n^m}$ be a word such that $\delta(w, C^m) = \delta$. Then the local view of the tester $\mathcal{D}_m$ is expected to be $\rho^{\mathcal{D}_m}(C^m) \cdot \delta$ far from C^{m-1}. When $\mathcal{D}_{m-1}$ will be invoked, its local view will be $\rho^{\mathcal{D}_m}(C^m) \cdot \rho^{\mathcal{D}_{m-1}}(C^{m-1}) \cdot \delta$ far from C^{m-2}, etc. Finally, the local view of $\mathcal{D}_3$ will be $(\rho^{\mathcal{D}_m}(C^m) \cdot \rho^{\mathcal{D}_{m-1}}(C^{m-1}) \cdot \ldots \cdot \rho^{\mathcal{D}_3}(C^3)) \cdot \delta$ far from C^2.

Theorem A.5 says that for every $i \geq 3$ we have $\rho^{\mathcal{D}_m}(C^m) \geq \frac{(\delta(C))^m}{2m^2}$. Hence for constant $m \geq 3$ it holds that $\rho^{\mathcal{D}_{comp}}(C^m) > 0$ is a constant that depends only on $\delta(C)$ and m.

Recall that the query complexity of $\mathcal{D}_{comp}$ is n^2. Claim C.1 implies that C^m is a $(n^2, \rho^{\mathcal{D}_{comp}}(C^m))$-strong LTC.

Now, let us show a more efficient way to compose the testers. Without loss of generality let us assume that $m/3$ is an integer, otherwise we would use $\lfloor m/3 \rfloor$ and $\lceil m/3 \rceil$. Then we have $C^m = C^{m/3} \otimes C^{m/3} \otimes C^{m/3}$, i.e., C^m is a 3-wise tensor product of $C^{m/3}$ with itself. Hence we can test it using a tester with robustness $\frac{(\delta(C^{m/3}))^3}{2 \cdot 3^2} = \frac{(\delta(C))^m}{18}$. The local view produced by this tester will be a candidate to be the codeword of $C^{m/3} \otimes C^{m/3} = C^{2m/3}$. I.e., we decreased the tensor degree of the underlying code from m to $2m/3$ in the single step. It follows that after $\log_{(3/2)} \frac{m}{2}$ steps we obtain the local view that is a candidate to be the codeword of C^2 that is entirely read by the composed tester. The robustness of this composed tester is

$$\frac{(\delta(C))^m}{18} \cdot \frac{(\delta(C))^(2m/3)}{18} \cdot \frac{(\delta(C))^(4m/9)}{18} \cdot \ldots \cdot \frac{(\delta(C))^2}{18} \geq \frac{(\delta(C))^{2m}}{18^{\log_{1.5} m}}.$$

□

C.2 Proof of Claim 3.2

Proof of Claim 3.2. Let $k = \dim(C)$. Let E_C be an encoder for the code C, which receives a message $x \in \mathbf{F}^k$ and outputs a codeword $E_C(x) \in C$ such that $C = \left\{E_C(x) \mid x \in \mathbf{F}^k\right\}$. Assume that E_C has running time $T = O(k)$. Note that this implies that $n \leq T = O(k)$ since the blocklength can not exceed the running time of the encoder.

For every $i \geq 1$ we define E_{C^i} to be the encoder for C^i, i.e., $C^i = \left\{E_{C^i}(x) \mid x \in \mathbf{F}^{k^i}\right\}$. We will argue that the running time of E_{C^i} is $i \cdot n^{i-1} \cdot T$. Since $n \leq T = O(k)$ we will conclude that for any constant $i \geq 1$ the running time of E_{C^i} is linear (in k^i).

We prove the claim by induction on i. The encoder $E_C = E_{C^1}$ was defined and its running time is $T = 1 \cdot n^{1-1} \cdot T$. Assume that we defined the encoder $E_{C^{i-1}}$ for the code C^{i-1} and its running time is $(i-1) \cdot n^{(i-1)-1} \cdot T$.

Let us define the encoder E_{C^i} for the code C^i. Note that the code C^i has message length k^i and its blocklength is n^i. Hence the message $x \in \mathbf{F}^{k^i}$ can be viewed as a matrix $k \times k^{i-1}$. So, we assume that $x \in \mathbf{F}^{k \times k^{i-1}}$. Note that every row of x belongs to $\mathbf{F}^{k^{i-1}}$.

The encoder E_{C^i} will first encode (by the encoder $E_{C^{i-1}}$) every row of the matrix x, obtaining the matrix $x' \in \mathbf{F}^{k \times n^{i-1}}$. The runtime of this step is $k \cdot ((i-1) \cdot n^{i-2}T)$. Then E_{C^i} will encode every column of the obtained matrix x' to get a codeword of C^i, and the runtime of this step is $n^{i-1}T$.

Hence the runtime of the encoder E_{C^i} is $k \cdot ((i-1) \cdot n^{i-2}T) + n^{i-1}T \leq ((i-1) \cdot n^{i-1}T) + n^{i-1}T = i \cdot n^{i-1} \cdot T$, where we used the fact that $k \leq n$. □

C.3 Proof of Proposition 3.6

Proof of Proposition 3.6. Recall that $C^m = C^{m-1} \otimes C$. It is sufficient to prove by induction on $j = 1, \dots, m$ that C^j is linear-time decodable from $\alpha^j n^j$ errors. For $j = 1$ the claim holds since $C^1 = C$. Let Dec_C be a linear-time decoder for the code C that can correct any $\alpha \cdot n$ errors. Assume that C^{j-1} is linear-time decodable from $\alpha^{j-1}n^{j-1}$ errors and let $Dec_{C^{j-1}}$ be its decoder.

We prove that $C^j = C^{j-1} \otimes C$ is linear-time decodable from $\alpha^j n^j$ errors. We define the linear-time decoder Dec_{C^j} for the code C^j that will correct any $\alpha^j \cdot n^j$ errors. Let $M \in \mathbf{F}^{n \times n^{j-1}}$ be an input word such that $\delta(M, C^j) \leq \alpha^j$.[12] The decoder Dec_{C^j} decodes every row of M (using $Dec_{C^{j-1}}$) to obtain the matrix $X_1 \in \mathbf{F}^{n \times n^{j-1}}$. Then $Dec_{C^{j-1}}$ decodes every column of X_1 (using Dec_C) to obtain the matrix $X_2 \in \mathbf{F}^{n \times n^{j-1}}$. Finally, the decoder outputs X_2.

Clearly, the runtime of Dec_{C^j} is $O(n^j)$, i.e., linear to the blocklength of C^j. Assume $X \in C^j$ is the closest codeword, i.e., $\delta(M, X) \leq \alpha^j$. We argue that $X_2 = X$, i.e., the decoder outputs the closest codeword.

For every $(a, b) \in [n] \times [n^{j-1}]$ such that $M|_{(a,b)} \neq X|_{(a,b)}$ we say that (a, b) is an error of M. Let

$$Bad_r = \left\{ i \in [n] \mid \delta(M|_{\{i\} \times [n^{j-1}]}, X|_{\{i\} \times [n^{j-1}]}) > \alpha^{j-1} \right\}$$

be the set of rows containing more than αn errors. Since $\delta(M, X) \leq \alpha^j$ we conclude that $|Bad_r| < \alpha n$.

Note that if $i \in [n] \setminus Bad_r$ then the i-th row of X_1 is equal to the i-th row of X, because $Dec_{C^{j-1}}$ corrects up to $\alpha^{j-1}n^{j-1}$ errors. That means less than αn rows of X_1 are different from the corresponding rows of X. It follows that every column of X_1 is α-close to the corresponding column of X, i.e., for every $a \in [n^{j-1}]$ we have $\delta(X_1|_{[n] \times \{a\}}, X|_{[n] \times \{a\}}) < \alpha$. Moreover, every column of X belongs to C by definition. We conclude that for every $j \in [n]$ the decoder Dec_C on the input $X_1|_{[n] \times \{j\}}$ will output $X|_{[n] \times \{j\}}$. This implies that $X_2 = X$.

This completes the induction and the proof of the proposition. □

C.4 Proof of Proposition 3.18

Proof of Proposition 3.18. Let $\mathbb{SC}$ be the self-corrector for the code C and assume without loss of generality that $\mathbb{SC}$ always queries exactly q queries. Let $M \in \mathbf{F}^{n \times n}$ be an input word (we view M as a matrix $n \times n$) and let $(i, j) \in [n] \times [n]$ be an input entry coordinate, the local-corrector for C^2 should retrieve. Assume that $\delta(M, C^2) \leq \delta^2$ and let $X \in C^2$ be the closest codeword, i.e., $\delta(M, X) \leq \delta^2$.

We turn to describe the self-corrector for the code C^2 and recall that the inputs are the matrix M and the coordinate (i, j).

1. Invoke $\mathbb{SC}$ on the row i of M, i.e., $M_{\{i\} \times [n]}$ to retrieve the entry (i, j). Call this the first invocation of $\mathbb{SC}$.

[12] We can view the matrix $M \in \mathbf{F}^{n^j}$ as a matrix in $\mathbf{F}^{n \times n^{j-1}}$.

2. For every queried coordinate (i, j_k) by the self-corrector $\mathbb{SC}$ (in the first invocation) return to it $\mathbb{SC}^{M|_{[n]\times\{j_k\}}}[(i, j_k)]$ as an answer instead instead of $M|_{(i,j_k)}$, i.e., return the output of the self-corrector $\mathbb{SC}$ on the column j_k of M and the input coordinate (i, j_k).

3. After q queries in the stage 2 were obtained, return the output of the first invocation of $\mathbb{SC}$.

Clearly, the self-corrector for C^2 queries at most q^2 entries. In the rest of the proof we prove that with probability at least $1 - (q + 1) \cdot \epsilon$ the self-corrector for C^2 outputs $X|_{(i,j)}$.

Assume that the first invocation of $\mathbb{SC}$ queried the coordinates $(i, j_1), (i, j_2), \ldots, (i, j_q)$. Note that this coordinate are on the row i of M.

Recall that to receive the value for the entry (i, j_1) the self-corrector $\mathbb{SC}$ was invoked on the column j_1 of M, i.e., on the vector $M|_{[n]\times\{j_1\}}$ and received the predicted value for the entry (i, j_1) (using q queries). In the similar way, to receive the value for the entry (i, j_1) the self-corrector $\mathbb{SC}$ was invoked on the column j_2 of M etc. Finally, after q^2 queries the values for the entries $(i, j_1), (i, j_2), \ldots, (i, j_q)$ are retrieved and using these values the first invocation of the self-corrector $\mathbb{SC}$ predicts the value for the entry (i, j).

We turn to analyze the error probability of retrieving the entry (i, j). Let us call the column k of M bad if $\delta(M|_{[n]\times\{k\}}, X|_{[n]\times\{k\}}) > \delta$, i.e., the column k of M has more than δ-fraction of noise. Since $\delta(M, X) \leq \delta^2$ the number of bad columns is upper-bounded by $\lfloor \delta n \rfloor$. Let $f = \lfloor \delta n \rfloor$ and assume without loss of generality that columns indexed by $\{1, 2, \ldots, f\}$ are bad, while all other columns of M are good.

Now, let x be the i-th row of X and note that $x \in C$. Note that for every $\hat{x} \in \mathbf{F}^n$ such that $\operatorname{supp}(x - \hat{x}) \subseteq [f]$ we have that the error probability of $\mathbb{SC}$ to retrieve correctly any entry of $\hat{x}$ is at most ϵ. That means regardless of the values of entries indexed by $[f]$, the self-corrector succeeds with probability at least $1 - \epsilon$.

Now, let us turn to our analysis of the retrieving the entry (i, j) from M and recall that the self-corrector for C^2 achieves this via retrieving the entries $(i, j_1), (i, j_2), \ldots, (i, j_q)$ via columns indexed by $\{j_1, j_2, \ldots, j_q\}$. The central point is that regardless of whether $\{j_1, \ldots, j_q\} \cap [f] = \emptyset$ or not the error probability of the retrieving (i, j) is upper-bounded by $q \cdot \epsilon + \epsilon$. This is true since the self-corrector uses at most q good columns, and for each good column the error probability in the retrieving the appropriate entry (i, j_l) is bounded by ϵ. Hence the total probability to error on at least one good column is at most $q \cdot \epsilon$. On the other side, the values retrieved from the bad columns (indexed by $[f]$) are irrelevant as was explained above. Given the fact that the self-corrector for C^2 retrieved correctly the entries from all good columns it queried, its error probability is at most ϵ.

Thus the total error probability of the self-corrector for C^2 is at most $q \cdot \epsilon + \epsilon$. This completes the proof of the Proposition. □