

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

June 3, 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

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 [29, 14] for more information). LTCs were implicit already in [2] (cf. [14, Sec. 2.4]) and they were explicitly studied by Goldreich and Sudan [16]. By now several different constructions of LTCs are known including codes based on low-degree polynomials

*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.

over finite fields [23, 1], constructions based on PCPs of proximity/assignment testers [3, 12]¹ and sparse random linear codes [9, 20, 21]. 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 . 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 [6, 24], new families of list-decodable codes [17], to give an alternative proof (see [25]) for IP=PSPACE theorem of [26, 27] etc.

Ben-Sasson and Sudan [6] suggested to use tensor product codes as a means to construct LTCs combinatorially. A natural hope would be to expect that given a code $C \subseteq \mathbf{F}^{n_1}$, whenever the task is to test whether an input word $M \in \mathbf{F}^{n_1 \times n_1}$ is in C^2 , the tester for C^2 can choose a random row (or column) of M ; and if M was far from C^2 , its typical row/column is far from C and hence can be tested on being in C . As was shown in [30, 15, 10] this approach fails in general and is known to work only for the base codes having some non-trivial properties [13, 7, 8].

Nevertheless, Ben-Sasson and Sudan [6] 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. More formally, they showed [6, Theorem 2.6] that for every $m \geq 3$ if $\left(\frac{\Delta(C)-1}{n}\right)^m \geq \frac{7}{8}$ then C^m is locally testable using n^2 queries. Note that the blocklength of C^m is n^m and query complexity is n^2 . Hence, for example, if $m = 10$ we obtain a code with blocklength $N = n^{10}$ and query complexity $N^{0.2} = n^2$, under the assumption that $\left(\frac{\Delta(C)-1}{n}\right)^{10} \geq \frac{7}{8}$.

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 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 larger distance implies the larger underlying field. It follows that this result can not provide (via tensor products) arbitrary low sublinear query complexity (N^ϵ for every constant $\epsilon > 0$) over a fixed field.

In this paper we ask the following question: is it possible to achieve a similar result to [6] 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.

We give a positive answer on this question and show that no assumptions about the base codes (or underlying fields) are not needed. Our result does not make any assumptions about the base codes, and in particular we do not assume that the base codes involved in tensor products have very large distance and hence it holds over any fields. This contrasts with previous works on the combinatorial constructions of LTCs due to Ben-Sasson and Sudan [6] and Meir [24] which required very large base-code distance implying large field size. The constructions of best known LTCs [5, 11, 24] were obtained over the large fields (when finally, the field size can be decreased through code concatenation). Our improvement of the result of [6] implies that the construction of Meir [24] (which achieves LTCs of best known parameters) can be taken directly over any field (including the binary field). We think that this improvement has a non-negligible

¹As was pointed out in [16], 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.

role since the LTCs construction of Meir is combinatorial and the combinatorial constructions of LTCs (or PCPs) should be independent, as much as possible, of the algebraic terms such as “polynomials”, “field size”, “extension field” etc.

Given the fact that error-correcting codes play an important role in a complexity theory, and in particular, in different iterative protocols, 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 a high-rate codes which combine such properties as local testing, efficient encoding and decoding from a constant fraction of errors.

We show that a combination of our results with the results of [28, 17] implies the construction of high-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.

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 prove our main theorem (Theorem 3.2) in Section 4. Finally, in Section 5 we prove our auxiliary statements.

2 Preliminaries

Throughout this paper, \mathbf{F} is a finite field, $[n]$ denotes the set $\{1, \dots, n\}$ 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 $\text{supp}(w) = \{i | w_i \neq 0\}$, $|w| = |\text{supp}(w)|$ and $\text{wt}(w) = \frac{|w|}{n}$. We define the *distance* between two words $x, y \in \mathbf{F}^n$ to be $\Delta(x, y) = |\{i | x_i \neq y_i\}|$ 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 $\text{rate}(C) = \frac{\dim(C)}{n}$. 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 $\Delta(C) = \min_{c \in C \setminus \{0\}} \{\text{wt}(c)\}$. 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 $u = (u_1, u_2, \dots, u_n) \in \mathbf{F}^n$ and $v = (v_1, v_2, \dots, v_n) \in \mathbf{F}^n$ is defined to be $\langle u, v \rangle = \sum_{i \in [n]} u_i \cdot v_i$. We let $C^\perp = \{u \in \mathbf{F}^n | \forall c \in C : \langle u, c \rangle = 0\}$ be the dual code of C and $C_t^\perp = \{u \in C^\perp | |u| = t\}$. In a similar way we define $C_{\leq t}^\perp = \{u \in C^\perp | |u| \leq t\}$. 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 $w \in \mathbf{F}^n$ and $S = \{j_1, j_2, \dots, j_m\} \subseteq [n]$, where $j_1 < j_2 < \dots < j_m$, we let $w|_S = (w_{j_1}, \dots, w_{j_m})$ be the *restriction* of w to the subset S . We let $C|_S = \{c|_S | c \in C\}$ 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. [13, 6, 24, 8, 30]).

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., [13]):

- 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 .
- $\dim(R \otimes C) = \dim(R) \cdot \dim(C)$
- $\text{rate}(R \otimes C) = \text{rate}(R) \cdot \text{rate}(C)$
- $\delta(R \otimes C) = \delta(R) \cdot \delta(C)$

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 $m \geq 1$ it holds that $\text{rate}(C^m) = (\text{rate}(C))^m$, $\delta(C^m) = (\delta(C))^m$ 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 [24] which showed how to use tensor products and avoid the decrease in the distance and the strong decrease in the rate.

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 $|I| \leq q$. Then T reads all symbols of $w|_I$ and accepts if $w|_I \in C|_I$, and rejects otherwise (see [4, Theorem 2]). Hence a q -query tester can be associated with a distribution over subsets $I \subseteq [n]$ such that $|I| \leq q$.

For purposes of composition we want to define a generalized tester (Definition 2.1) which does not make queries, but selects and returns a “view” (a subset $I \subseteq [n]$) which can be considered as a code by itself ($C|_I$).

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 $|I| \leq q$. 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*.

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

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 $\Pr_{I \sim \mathbf{D}}[w|_I \notin C|_I] \geq \epsilon$.

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

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 . In the rest of this section we formally define the notion of *robustness* (Definition 2.4) as was introduced in [6]. To do that we start from the definition of *local distance* (Definition 2.3), which will be used in Definition 2.4 and later in our proofs.

Definition 2.3 (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 $\Delta(w|_I, C|_I) = \min_{c \in C} \{\Delta(w|_I, c|_I)\}$ and similarly the *relative local distance* of w from C with respect to I (relative I -distance of w from C) is $\delta(w|_I, C|_I) = \min_{c \in C} \{\delta(w|_I, c|_I)\}$.

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 [3, 11]. We are ready to provide a general definition of robustness.

Definition 2.4 (Robustness). Given a tester (i. e., a distribution) \mathbf{D} for the code $C \subseteq \mathbf{F}^n$, we let

$$\rho^{\mathbf{D}}(w) = \mathbf{E}_{I \sim \mathbf{D}} [\delta(w|_I, C|_I)]$$

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

3 Main Results

The tester we consider in this paper is the plane tester (suggested in [6]).

Definition 3.1 (Plane Tester). Let $m \geq 3$. Let $M \in \mathbf{F}^{m^m}$ be an input word and think of testing whether $M \in C^{m^m}$. The plane tester \mathcal{D} picks (non-adaptively) a random $b \in [3]$ and random $i \in [n]$, and returns (b, i) -plane (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} .

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 [6] which showed that this claim holds for base codes with a very large distance.

Theorem 3.2 (Main Theorem). *Let $C \subseteq \mathbf{F}^n$ be a linear code and $m \geq 3$. Let \mathcal{D} be the plane tester for C^m . Then*

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

The proof of Theorem 3.2 is postponed to Section 4. Theorem 3.2 extends the main result of Ben-Sasson and Sudan [6] since it implies that the m -wise tensor product of linear codes is robust for any linear base codes with constant relative distance. In particular, the tensor product can be applied over any field, including the binary field. So, as explained in the introduction, the combinatorial construction of LTCs in [24] can be taken over any field (regardless of the field size).

Ben-Sasson and Sudan [6] explained that plane testers can be composed and the robustness of the plane testers implies the strong local testability. For the sake of completeness we state this claim formally in Corollary 3.3, and provide a proof-sketch in Section 5 (see [6, 7] for more information about composition of the testers).

Corollary 3.3. *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 .*

Corollary 3.3 implies that any linear code can be used to define a locally testable code with sublinear query complexity. Claim 3.4 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 Corollary 3.3 to show Corollary 3.5.

Claim 3.4. *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 3.4](#) is postponed to [Section 5](#). Now, we combine [Corollary 3.3](#) and [Claim 3.4](#) 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 3.5. *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 $\epsilon > 0$, $m = \lceil \frac{1}{\epsilon} \rceil$, $N = n^m$ and $C \subseteq \mathbf{F}^n$ such that $\text{rate}(C) \geq (1 - \epsilon)^{1/m}$ we have $C^m \subseteq \mathbf{F}^N$ is a (N^ϵ, α) -strong LTC and $\text{rate}(C^m) \geq 1 - \epsilon$, 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.

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. [[22](#)] showed the construction of high-rate locally decodable codes with sublinear query complexity (see [[22](#)] 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 3.5](#) 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 codes with constant rate and constant relative distance [[18](#), [19](#), [28](#)], 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 product of the base code is known to inherit some properties of the base codes besides local testability. E.g., Gopalan et al. [[17](#)] showed that the tensor product operation preserves list-decodability properties. Furthermore, we know about nice constructions of error-correcting codes that can be efficiently encoded and decoded (list-decoded) from a constant fraction of errors (see e.g., [[18](#), [19](#)]).

In [Section 3.1](#) we show how testing with sublinear query complexity can be combined with a linear time encoding and decoding. Then, in [Section 3.2](#) we show that [Corollary 3.5](#) can be combined with the result of [[17](#)] to define asymptotically good codes that can be encodable in linear time, testable with sublinear query complexity and list-decodable in polynomial time.

3.1 Linear-time decodable codes

[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 unique-neighbor decodable from a constant fraction of errors then so is C^2 . Hence this observation, together with a result of, e.g. [[28](#)], 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 neighbor 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. Then $C^2 = C \otimes C$ is a linear code that is linear-time decodable from $\frac{\alpha^2}{100} \cdot n^2$ errors.*

Proof. Let Dec_C be a linear-time decoder for the code C that can correct any $\alpha \cdot n$ errors. Note that in particular, Dec_C correct any $\alpha \cdot n$ erasures in the linear time. We define the linear-time decoder Dec_{C^2} for the code C^2 that will correct any $\frac{\alpha^2}{100} \cdot n^2$ errors.

To do this, let $M \in \mathbf{F}^{n \times n}$ be an input word. The decoder Dec_{C^2} will decode every row of M using Dec_C and every column of M using Dec_C . Note that every entry of M is contained in (exactly) one row

and one column. Call the entry (i, j) of M an inconsistent if row decoding gives to $M|_{(i,j)}$ a different value from column decoding, and otherwise the entry is called consistent.

We call the row (column) of M bad if it contains at least αn inconsistent entries. Let Bad_r be a number of bad rows and Bad_c be a number of bad columns. It holds that $Bad_r \cdot \alpha n \leq \frac{\alpha^2 n^2}{100}$ and hence $Bad_r \leq \alpha n/100$. Similarly, $Bad_c \leq \alpha n/100$.

The decoder Dec_{C^2} removes all bad rows and bad columns that have at least $\alpha n/2$ inconsistent entries and obtains a large submatrix of size at least $(1 - \alpha/100)n \times (1 - \alpha/100)n$. It is easy to see that all consistent entries in the above large submatrix were decoded correctly.

In the last step, the decoder Dec_{C^2} decodes, using Dec_C , every row of the large submatrix of M (of size at least $(1 - \alpha/100)n \times (1 - \alpha/100)n$) from at most $\alpha n/100$ erasures and obtains a submatrix of size at least $(1 - \alpha/100)n \times n$. Now, it decodes every column of the submatrix to the full matrix. It can be easily verified that the decoder Dec_{C^2} obtains a correct codeword of C^2 and runs in linear time. \square

While the results of [28] were improved, for our purpose (Corollary 3.8) this result is sufficient.

Theorem 3.7 ([28]). *There exists an (explicit) family of linear error correcting codes $C \subseteq \mathbb{F}_2^n$ such that $\text{rate}(C) = \Omega(1)$, $\delta(C) = \Omega(1)$, 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 3.4 results in the following corollary.

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

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

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 3.9 (List-decodable codes). A code C is a (α, L) -list decodable if for every word $w \in \mathbb{F}^n$, $\delta(w, C) \leq \alpha$ 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 $w \in \mathbb{F}^n$ 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. [17] showed that the list-decodability is preserved in the tensor product operation. More formally, they showed the following theorem stated in [17, Theorem 5.7].

Theorem 3.10 ([17]). *Let \mathbb{F} be a finite field and $q = |\mathbb{F}|$. Given two linear codes $C_1, C_2 \subseteq \mathbb{F}^n$, 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 efficiently list decoded 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 efficiently 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 $O(4q^{\frac{1}{4\delta_1^2 \epsilon^2} \ln \frac{8l_1(\eta_1)}{\epsilon} \ln \frac{8l_2(\eta_2)}{\epsilon}} \cdot T n_1 n_2)$.*

Then, Gopalan et al. used Theorem 3.10 to conclude the following theorem, appearing in [17, Theorem 5.8].

Theorem 3.11 ([17]). *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 running time 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.

Fact 3.12. There exist linear error-correcting codes 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 3.11, Fact 3.12, Claim 3.4 and Corollary 3.5 to conclude the following corollary.

Corollary 3.13. *Let \mathbf{F} be any field. For every constant $\epsilon > 0$ there exists a code $C' \subseteq \mathbf{F}^N$ such that $C' = C^m$, where $C \subseteq \mathbf{F}^n$ is a linear code, $\text{rate}(C) = \Omega(1)$, $\delta(C) = \Omega(1)$ and C is (ρ, L) -list decodable in polynomial time.*

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

4 Proof of Theorem 3.2

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 . Throughout this paper we assume that $m \geq 3$ and for the case of $m = 2$ we refer a reader to [7, 8, 13, 30, 15, 10, 10]). We start this section by defining the concepts of points, lines and planes (some of the terms were defined following [6]).

4.1 Preliminary notations: Points, Lines and Planes

A point in such a code can be associated with an m -tuple (i_1, i_2, \dots, i_m) such that $i_j \in [n]$. Next we define an axis parallel line, or shortly, a line which can be associated with a subset of points. For $b \in [m]$ and $i \in [n]$ we say that l is a $(b, (i_1, i_2, \dots, i_{b-1}, i_{b+1}, \dots, i_m))$ -line if

$$l = \{(i_1, i_2, \dots, i_{b-1}, i, i_{b+1}, \dots, i_m) \mid \text{for all } j \in [m] \setminus \{b\} \text{ we have } i_j = i_j\}.$$

Note that $(b, (i_1, i_2, \dots, i_{b-1}, i_{b+1}, \dots, i_m))$ -line is parallel to the b -th axis. A line l contains a point p if $p \in l$. Note that a $(b, (i_1, i_2, \dots, i_{b-1}, i_{b+1}, \dots, i_m))$ -line contains a point $p = (j_1, i_2, \dots, j_m)$ 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 say that pl is a (b, i) -plane if

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

A (b, i) -plane contains a point $p = (i_1, i_2, \dots, i_m)$ if $i_b = i$, i.e., b -th coordinate of the point is i . A (b, i) -plane contains a line l if it contains all points of the line. We say that two (different) planes are intersected if

both planes contain at least one common point. Note that two (different) planes: (b_1, i_1) -plane and (b_2, i_2) -plane are intersected iff $b_1 \neq b_2$, moreover, they are intersected on all points $p = (i_1, \dots, i_m)$ such that $i_1 = i_{b_1}$ and $i_2 = i_{b_2}$, i.e., are intersected on n^{m-2} points.

Assume that pl_1 is a (b_1, i_1) -plane and pl_2 is a (b_2, i_2) -plane such that $b_1 < b_2$ (in particular $b_1 \neq b_2$). Let $pl_1 \cap pl_2 = \{(i_1, \dots, i_m) \mid i_{b_1} = i_1, i_{b_2} = i_2\}$ be an intersection of two planes and $C^m|_{pl_1 \cap pl_2}$ be a code C^m restricted to the points in $pl_1 \cap pl_2$. Note that $\delta(C^m|_{pl_1 \cap pl_2}) = \delta(C^{m-2}) = \delta(C)^{m-2}$.

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) -plane, i.e., to all points of the plane. We say that $M_{(b,i)}$ is a (b, i) -plane of M . Similarly, for the point $p = (i_1, \dots, i_m)$ 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 .

4.2 The proof itself

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

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

$$\rho^{\mathcal{D}}(M) = \mathbf{E}_{pl \sim \mathcal{D}} [\delta(M|_{pl}, r(pl))]. \quad (4.1)$$

We say that the (b_1, i_1) -plane and the (b_2, i_2) -plane disagree on the point $p = (i_1, \dots, i_m)$ if (b_1, i_1) -plane and (b_2, i_2) -plane are intersected, both contain the point p and $r(pl_1)|_p \neq r(pl_2)|_p$. We say that two planes disagree on the line l if both planes are intersected, both contain the line l and $r(pl_1)|_l \neq r(pl_2)|_l$.

Note that if (b_1, i_1) -plane pl_1 and (b_2, i_2) -plane pl_2 are intersected and disagree on at least one point then letting $reg = pl_1 \cap pl_2$ we have $r(pl_1)|_{reg} \neq r(pl_2)|_{reg}$ and moreover, $\delta(r(pl_1)|_{reg}, r(pl_2)|_{reg}) \geq (\delta(C))^{m-2}$. This is true since $r(pl_1)|_{reg}, r(pl_2)|_{reg} \in C^{m-2}$ are non-equal codewords of 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 at least two planes 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 plane pl such that $r(pl)|_p \neq M|_p$. Intuitively, a point p is almost fixed if all planes containing this point agree on this point but “think” that its value in M ($M|_p$) should be changed (to $r(pl)|_p$).

We let $ToFix = \{p = (i_1, i_2, \dots, i_m) \mid p \text{ is almost fixed}\}$ and let $NumToFix = |ToFix|$.

Proposition 4.1. *It holds that $\rho^{\mathcal{D}}(M) \geq \frac{wt(E)}{m} + \frac{NumToFix}{n^m}$.*

Proof. Equation 4.1 says that $\rho^{\mathcal{D}}(M)$ is a relative distance of a typical plane of M (which is a word in $\mathbf{F}^{n^{m-1}}$) from C^{m-1} . Note that for every point $p = (i_1, \dots, i_m)$: 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 planes pl of M we have $(M|_{pl})|_p \neq r(pl)|_p$. Now, every point p is contained in m different planes. Hence if $E|_p \neq 0$ then for at least one plane pl (of m planes containing the point p) it holds that $r(pl)|_p \neq M|_p$.

Hence a relative distance between a typical plane (pl) of M and $r(pl)$ is at least $\frac{wt(E)}{m} + \frac{NumToFix}{n^m}$. \square

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

Definition 4.2 (Heavy lines and planes). A line l of E is called heavy if $|E|_l \geq \delta(C) \cdot n$. A plane (b, i) of E is called heavy if $|E|_{(b,i)} \geq \frac{(\delta(C) \cdot n)^{m-1}}{2}$.

Lemma 4.3 is our main observation in the proof of Theorem 3.2. It says that any non-zero element of E is located in some heavy plane of E .

Lemma 4.3 (Main Lemma). *Let $p = (i_1, i_2, \dots, i_m)$ be a point such that $E_p \neq 0$. Then p is contained in some heavy plane of E .*

The proof of Lemma 4.3 is postponed to Section 4.2.1. Corollary 4.4 shows that it is sufficient to remove at most $\text{wt}(E) \cdot (\delta(C))^{m-1} / 2 \cdot \frac{n}{2}$ planes from E to get a zero submatrix.

Corollary 4.4. *There exists $S_1, \dots, S_m \subseteq [n]$ such that $n - |S_1| + n - |S_2| + \dots + n - |S_m| \leq \frac{2|E|}{(\delta(C)n)^{m-1}} \cdot m$ and letting $S = S_1 \times S_2 \times \dots \times S_m$ we have $E|_S = 0$.*

Proof. Let $\text{HeavyPlanes} = \{(b, i) \mid (b, i) \text{ is a heavy plane}\}$ to be a subset of pairs associated with heavy planes. For $b \in [m]$ let $\overline{S}_b = \{i \in [n] \mid (b, i) \in \text{HeavyPlanes}\}$ and $S_b = [n] \setminus \overline{S}_b$.

We claim that $|\text{HeavyPlanes}| \leq \frac{2|E|}{(\delta(C)n)^{m-1}} \cdot m$. This is true since every heavy plane contains at least $\frac{(\delta(C)n)^{m-1}}{2}$ 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) planes. Thus $n - |S_1| + n - |S_2| + \dots + n - |S_m| = \sum_{b \in [m]} |\overline{S}_b| \leq \frac{2|E|}{(\delta(C)n)^{m-1}} \cdot m$.

Now, note that Lemma 4.3 implies that for every point $p = (i_1, i_2, \dots, i_m)$ such that $E_p \neq 0$ is contained in some heavy plane, i.e., in some plane from HeavyPlanes . Hence if all heavy planes are removed from E we obtain a zero submatrix. So, it follows that $E|_S = 0$. \square

Proposition 4.5 says that if after removing small fraction of planes from M we obtain a submatrix that is close to the legal submatrix then M is close to C^m .

Proposition 4.5. *Let $S_1, S_2, \dots, S_m \subseteq [n]$ be such that $n - |S_1| + n - |S_2| + \dots + n - |S_m| \leq \tau n < \delta(C) \cdot n$ and let $S = S_1 \times S_2 \times \dots \times S_m$. Let $C' = C|_{S_1} \otimes C|_{S_2} \otimes \dots \otimes C|_{S_m}$. Recall that $M|_S$ is a submatrix of M obtained by removing at most τ -fraction of planes. Assume that $\Delta(M|_S, C') \leq \alpha \cdot n^m$. Then $\delta(M, C^m) \leq \tau + \alpha$.*

The proof of Proposition 4.5 appears in Section 4.2.2. Let us prove Theorem 3.2.

Proof of Theorem 3.2. By Proposition 4.1 we have $\rho^{\mathcal{D}}(M) \geq \frac{\text{wt}(E)}{m} + \frac{\text{NumToFix}}{n^m}$. If $\text{wt}(E) \geq \frac{(\delta(C))^m}{2m}$ then we are done. Otherwise, assume that $\text{wt}(E) < \frac{(\delta(C))^m}{2m}$.

Corollary 4.4 implies that it is sufficient to remove at most $\frac{2|E|}{(\delta(C)n)^{m-1}} \cdot m < \delta(C) \cdot n$ planes from E to get a zero submatrix. Proposition 4.5 implies that $\delta(M, C^m) \leq \frac{2 \text{wt}(E)}{(\delta(C))^{m-1}} \cdot m + \frac{\text{NumToFix}}{n^m}$.

Let $\beta = \frac{2m^2}{(\delta(C))^{m-1}}$. Then $\rho^{\mathcal{D}}(M) \cdot \beta \geq (\frac{\text{wt}(E)}{m} + \frac{\text{NumToFix}}{n^m}) \cdot \beta \geq \delta(M, C^m)$ and $\rho^{\mathcal{D}}(M) \geq \frac{(\delta(C))^{m-1}}{2m^2}$. \square

4.2.1 Proof of Main Lemma 4.3

In this section we prove Lemma 4.3.

Proof of Main Lemma 4.3. By definition of E we know that there are (at least) two planes that disagree on the point p . Assume without loss of generality (symmetry) that the planes $p_1 = (1, i_1)$ and $p_2 = (2, i_2)$ disagree on the point p . We will prove that either p_1 is a heavy plane or p_2 is a heavy plane.

Consider the intersection of p_1 and p_2 , i.e., $reg = pl_1 \cap pl_2 = \{(i_1, i_2, j_3, j_4, \dots, j_m) \mid j_k \in [n]\}$. Note that $p \in reg$. Let ln be a line, which is parallel to the axis 3 and contains a point p (recall that $m \geq 3$). Then the planes p_1 and p_2 disagree on this line (since they disagree on the point p contained in the line ln), i.e., $r(p_1)|_{ln} \neq r(p_2)|_{ln}$. But $r(p_1)|_{ln}, r(p_2)|_{ln} \in C$ by definition. This implies that $\Delta(r(p_1)|_{ln}, r(p_2)|_{ln}) \geq \delta(C) \cdot n$, i.e., for at least $\delta(C) \cdot n$ points $p \in ln$ it holds that $r(p_1)|_p \neq r(p_2)|_p$.

Let $BadPoints = \{p \in ln \mid p_1 \text{ and } p_2 \text{ disagree on } p\}$. Note that $|BadPoints| \geq \delta(C) \cdot n$. Note that $|BadPoints| \geq \delta(C) \cdot n$. Let $BadPlanes = \{(3, i) - \text{plane} \mid i \in [n], \exists p \in BadPoints \text{ s.t. } p \in (3, i) - \text{plane}\}$. Note that $|BadPlanes| \geq \delta(C) \cdot n$.

We claim that for every $plane \in BadPlanes$ we have that either $plane$ disagrees with p_1 on some point $p \in BadPoints$ or with p_2 on some point $p \in BadPoints$. Hence at least one of p_1, p_2 disagrees with at least $\frac{1}{2} \cdot |BadPlanes| \geq \frac{1}{2} \cdot \delta(C)n$ planes from $BadPlanes$. Without loss of generality assume that p_1 disagrees with at least $\frac{1}{2} \cdot \delta(C) \cdot n$ planes from $BadPlanes$.

Let $BadPlanes_{p_1} = \{plane \in BadPlanes \mid plane \text{ disagrees with } p_1\}$. Note that all planes from $BadPlanes$ are non-intersecting and thus all planes from $BadPlanes_{p_1}$ are non-intersecting. Every plane $pl \in BadPlanes_{p_1}$ disagrees with the plane p_1 on some point and hence disagree on at least $(\delta(C)n)^{m-2}$ points in their intersection region $(pl \cap p_1)$ since $r(pl)|_{pl \cap p_1} \neq r(p_1)|_{pl \cap p_1} \in C^{m-2}$.

Let $total = \{p = (i_1, j_2, \dots, j_m) \mid \exists plane \in BadPlanes_{p_1} \text{ s.t. } p \in p_1 \cap plane, r(plane)|_p \neq r(p_1)|_p\}$. 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 planes in $BadPlanes_{p_1}$ are non-intersecting.

Hence the plane p_1 disagree with other planes in at least $\frac{(\delta(C) \cdot n)^{m-1}}{2}$ points (on the plane). Thus $E|_{p_1}$ has at least $\frac{(\delta(C) \cdot n)^{m-1}}{2}$ non-zero symbols. We conclude that p_1 is a heavy plane of E and the point p is contained in the plane p_1 . \square

4.2.2 Proof of Proposition 4.5

In this section we prove Proposition 4.5.

Proof of Proposition 4.5. Note that for every $i \in [n]$ we have $|S_i| > n - \delta(C) \cdot n$. The following simple claim was proven in [6, 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 4.5. 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 a codeword of C^m by changing at most τ -fraction of symbols (since all symbols outside the submatrix $M|_S$ are at most τ -fraction of all symbols). We conclude that $\delta(M, C^m) \leq \tau + \alpha$. \square

5 Proofs of Auxiliaries Corollaries and Claims

In this section we first proof Corollary 3.3.

Proof Sketch: For $i \geq 3$ let \mathcal{D}_i be the plane 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} . Hence \mathcal{D}_{m-1} can be invoked on the local view of \mathcal{D}_m , etc. So, the testers $\mathcal{D}_m, \mathcal{D}_{m-1}, \dots, \mathcal{D}_3$ can be composed to result in a n^2 -query tester \mathcal{D}_{comp} for the code C^m .

The robustness of the composed tester will be $\rho^{\mathcal{D}_{comp}}(C^m) \geq \rho^{\mathcal{D}_m}(C^m) \cdot \rho^{\mathcal{D}_{m-1}}(C^{m-1}) \cdot \dots \cdot \rho^{\mathcal{D}_3}(C^3)$. 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 $(\rho^{\mathcal{D}_m}(C^m) \cdot \rho^{\mathcal{D}_{m-1}}(C^{m-1}) \cdot \dots \cdot \rho^{\mathcal{D}_3}(C^3)) \cdot \delta$ far from C^2 .

Theorem 3.2 says that $\rho^{\mathcal{D}_m}(C^m) \geq \frac{(\delta(C))^m}{2m^2}$. 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 .

Now, let $\alpha_m = \rho^{\mathcal{D}_{comp}}(C^m)$ and note that the query complexity of \mathcal{D}_{comp} is n^2 . Let $M \in \mathbf{F}^{n^m}$ such that $\delta(M, C^m) = \delta$ and think of testing whether M in C^m . We argue that $\Pr_{I \sim \mathcal{D}_{comp}}[M|_I \in C^m|_I] \geq \alpha_m$. This is true since otherwise, $\Pr_{I \sim \mathcal{D}_{comp}}[M|_I \notin C^m|_I] < \alpha_m$, and then the robustness $\rho^{\mathcal{D}_{comp}}(C^m) < \alpha_m \cdot 1$ (even assuming that whenever $M|_I \notin C^m|_I$ we have $\delta(M|_I, C^m|_I) = 1$). Contradiction.

This proves that C^m is a (n^2, α_m) -strong LTC. \square

Now we prove Claim 3.4.

Proof of Claim 3.4. 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 $C = \{E_C(x) \mid x \in \mathbf{F}^k\}$. 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., $C^i = \{E_{C^i}(x) \mid x \in \mathbf{F}^{k^i}\}$. 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 $k \times k^{i-1}$ and the encoder E_{C^i} will first encode (by the encoder $E_{C^{i-1}}$) every row of the matrix, obtaining the matrix $k \times n^{i-1}$. Then E_{C^i} will encode every column of the obtained matrix to get a codeword of C^i . The running time of the encoder C^i is $k \cdot ((i-1) \cdot n^{i-2}T) + n^{i-1}T \leq ((i-1) \cdot n^{i-1}T) + n^{i-1}T = i \cdot n^{i-1} \cdot T$. \square

Acknowledgements

We thank Eli Ben-Sasson and Or Meir for helpful discussions. The author thanks also Eli Ben-Sasson for many invaluable discussions regarding the ‘‘robustness’’ concept.

References

- [1] 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.
- [2] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proc. 23rd STOC*, pages 21–31. ACM, 1991.

- [3] 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.
- [4] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF Properties Are Hard to Test. *SIAM Journal on Computing*, 35(1):1–21, 2005.
- [5] Eli Ben-Sasson and Madhu Sudan. Simple PCPs with poly-log rate and query complexity. In *STOC*, pages 266–275. ACM, 2005.
- [6] Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Struct. Algorithms*, 28(4):387–402, 2006.
- [7] 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.
- [8] Eli Ben-Sasson and Michael Viderman. Tensor Products of Weakly Smooth Codes are Robust. *Theory of Computing*, 5(1):239–255, 2009.
- [9] 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.
- [10] Don Coppersmith and Atri Rudra. On the Robust Testability of Product of Codes. *Electronic Colloquium on Computational Complexity (ECCC)*, (104), 2005.
- [11] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12:1–12:44, June 2007.
- [12] Irit Dinur and Omer Reingold. Assignment Testers: Towards a Combinatorial Proof of the PCP Theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006.
- [13] 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.
- [14] Oded Goldreich. Short locally testable codes and proofs (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, (014), 2005.
- [15] 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.
- [16] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *Journal of the ACM*, 53(4):558–655, July 2006.
- [17] Parikshit Gopalan, Venkatesan Guruswami, and Prasad Raghavendra. List decoding tensor products and interleaved codes. In Michael Mitzenmacher, editor, *STOC*, pages 13–22. ACM, 2009.
- [18] Venkatesan Guruswami and Piotr Indyk. Linear time encodable and list decodable codes. In *STOC*, pages 126–135. ACM, 2003.

- [19] Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005.
- [20] Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *FOCS*, pages 590–600. IEEE Computer Society, 2007.
- [21] 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.
- [22] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. In *ECCC - TR10-148*, 2010.
- [23] M.Blum, M.Luby, and R.Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *JCSS: Journal of Computer and System Sciences*, 47, 1993.
- [24] Or Meir. Combinatorial Construction of Locally Testable Codes. *SIAM J. Comput.*, 39(2):491–544, 2009.
- [25] Or Meir. $IP = PSPACE$ using Error Correcting Codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:137, 2010.
- [26] Adi Shamir. $IP = PSPACE$. *J. ACM*, 39(4):869–877, 1992.
- [27] A. Shen. $IP = PSPACE$: Simplified proof. *J. ACM*, 39(4):878–880, 1992.
- [28] 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.
- [29] Luca Trevisan. Some Applications of Coding Theory in Computational Complexity. In *ECCC: Electronic Colloquium on Computational Complexity, technical reports*, 2004.
- [30] 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.