Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 22 (2013)

Strong LTCs with inverse poly-log rate and constant soundness

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

March 31, 2013

Abstract

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

In this paper we resolve an open question raised by Goldreich and Sudan (J.ACM 2006) and construct binary linear strong LTCs with query complexity 3, constant relative distance, constant soundness and inverse polylogarithmic rate.

Our result is based on the previous paper of the author (Viderman, ECCC TR12-168), which presented binary linear strong LTCs with query complexity 3, constant relative distance, and inverse polylogarithmic soundness and rate. We show that the "gap amplification" procedure of Dinur (J.ACM 2007) can be used to amplify the soundness of these strong LTCs from inverse polylogarithmic up to a constant, while preserving the other parameters of these codes.

Furthermore, we show that under a conceivable conjecture, there exist asymptotically good strong LTCs with poly-log query complexity.

*The research has received funding from the European Research Council as part of the ERC project CaC (grant 259426).

ISSN 1433-8092

Contents

1 Introduction

Probabilistically Checkable Proof (PCP) systems [2, 3, 31] (a.k.a. Holographic Proofs [5]) are proof systems that allow efficient probabilistic verification of a claim by reading few symbols of the proof. The celebrated PCP theorem [2, 3] is one of the main breakthrough results in complexity theory. This theorem asserts that for every language in $\mathcal{NP}$ there exists a polynomial-time PCP verifier that queries the proof in a constant number of locations. The verifier is guaranteed to always accept valid proofs of true statements, and to accept any claimed proof of false assertions with low probability. The theorem has found many applications in theoretical computer science, especially in establishing lower bounds for approximation algorithms [8, 6, 31, 40].

Informally, most of the PCP constructions were achieved using error-correcting codes, possessing nice properties. Let us first give some auxiliary definitions regarding error-correcting codes.

A code over a finite alphabet Σ is a subspace $\mathcal{C} \subseteq \Sigma^n$. A linear code over a finite field $\mathbb{F}$ is a linear subspace $\mathcal{C} \subseteq \mathbb{F}^n$. In this case, n is the blocklength of the code $\mathcal{C}$, denoted by $\text{blocklength}(\mathcal{C})$. The dimension of a linear code $\mathcal{C}$, denoted by $\dim(\mathcal{C})$, is its dimension as a vector space and is equal to $\log_{|\mathbb{F}|} |\mathcal{C}|$. The dimension of a non-linear code $\mathcal{C}$ over the alphabet Σ is defined to be $\dim(\mathcal{C}) = \log_{|\Sigma|} |\mathcal{C}|$. The rate of a code $\mathcal{C}$, denoted by $\text{rate}(\mathcal{C})$, is defined to be $\frac{\dim(\mathcal{C})}{\text{blocklength}(\mathcal{C})} = \frac{\dim(\mathcal{C})}{n}$.

We define the distance between two words $x, y \in \mathbb{F}^n$ to be $\Delta(x, y) = |\{i \mid x_i \neq y_i\}|$ and the relative distance to be $\delta(x, y) = \frac{\Delta(x,y)}{n}$. The distance of $\mathcal{C}$ is defined by $\Delta(\mathcal{C}) = \min_{x \neq y \in \mathcal{C}} \Delta(x, y)$ and its relative distance is defined by $\delta(\mathcal{C}) = \frac{\Delta(\mathcal{C})}{n}$. We note that if $\mathcal{C}$ is linear then $\Delta(\mathcal{C}) = \min_{c \in \mathcal{C} \setminus \{0\}} \{|c|\}$. One is typically interested in codes whose distance is linear to the blocklength of $\mathcal{C}$, i.e., $\Omega(n)$.

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

1.1 Locally Testable Codes

Most of the PCP constructions (e.g., [10, 19, 27, 36]) are tightly related to a special kind of error-correcting codes possessing some testability properties. These codes are called *locally testable*.

In other words, 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 [53, 33] for more information). LTCs were implicit already in [5] (cf. [33, Sec. 2.4]) and they were explicitly studied by Goldreich and Sudan [36].

By now several different constructions of LTCs are known including codes based on low-degree polynomials over finite fields and affine-invariant codes [1, 2, 26, 17, 12, 25, 38, 43, 45, 42, 52], constructions based on PCPs of proximity/assignment testers [10, 29, 27][1], sparse random linear codes [23, 44, 49] and tensor products of codes [30, 22, 21, 50, 54].

Basically, there are two kinds of LTCs: weak and strong. A code $\mathcal{C}$ is said to be (q, ϵ, ρ)-weak LTC if there exists a randomized algorithm T, called tester, that makes at most q queries to the

[1] As was pointed out in [36], 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.

input word w. If $w \in \mathcal{C}$ then T accepts w with probability 1, but if w is ρ-far from C the tester T rejects w with probability at least ϵ. Let us notice that the tester is not required to reject when $0 < \delta(w, \mathcal{C}) < \rho$. This is the reason why such codes are called *weak* LTCs.

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

Informally, we say that a code $\mathcal{C}$ is a weak LTC if it has a linear distance and there exist constants $q, \epsilon > 0$ and $\rho \leq \delta(\mathcal{C})/3$ such that C is a (q, ϵ, ρ)-weak LTC. [2] Similarly, we say that a code $\mathcal{C}$ is a strong LTC if it has a linear distance and there exist constants $q, \epsilon > 0$ such that $\mathcal{C}$ is a (q, ϵ)-strong LTC.

The best known strong LTCs are due to Goldreich and Sudan [36], who presented probabilistic construction of strong LTCs. These LTCs achieve constant query complexity, constant soundness and rate $\frac{1}{\exp(\tilde{O}(\sqrt{\log n}))}$, where n denotes the blocklength.

Later, other constructions of LTCs [19, 27, 50] succeeded to obtain the rate $\frac{1}{\text{polylog}(n)}$ together with constant query complexity and soundness, however these codes were weak LTCs. It can be verified that every strong LTC is also a weak LTC, but some weak LTCs are not strong LTCs [55]. So, strong LTCs are strictly stronger objects than weak LTCs. In the journal version of [36], the authors pointed out that all known LTCs that achieve inverse polylogarithmic rate are weak LTCs, and asked about the existence of strong LTCs with polylogarithmic rate [36, Section 6]. As was pointed out by Goldreich [32], strong LTCs correspond to proximity oblivious testers [35] whereas weak LTCs are even weaker than ordinary testers, i.e., the testers for weak LTCs are supposed to work only for a fixed value of the proximity parameter.

The previous paper of the author [55] showed a probabilistic construction of binary linear 3-query strong LTCs with inverse polylogarithmic rate, inverse polylogarithmic soundness and constant relative distance. In this paper (Section 1.3), we show how to amplify the soundness parameter of these codes from inverse polylogarithmic to constant, while preserving the other parameters of these codes, therefore resolving an open question raised by Goldreich and Sudan [36]. To increase the soundness parameter we apply the gap amplification technique of Dinur [27].

An interesting point is that the gap amplification was known to improve the soundness parameter of weak LTCs [27, 50], however it was not known to preserve the strong testability requirement, where all non-codewords are rejected with corresponding probability and not only words that are sufficiently far from the code. In more details, the gap amplification procedure outputs a code accompanied with a probabilistically checkable proof that could be translated to a weak LTC.

In [55] we conjectured that it should be possible to modify this procedure to preserve this stronger property. Surprisingly, it turns out that no modification is needed (besides adapting the gap amplification to preserve the linearity of the underlying codes, as was done in [50, Section 6.4]). In Section 1.3 we present formally our main result (Theorem 1.4) and explain the ideas that lead to its proof.

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

1.1.1 Asymptotically good LTCs

The main open question in the area of LTCs is whether there exists a family of asymptotically good LTCs with constant query complexity and soundness, i.e., LTCs over a constant size alphabet that have constant query complexity, constant soundness parameter, constant rate and constant relative distance [36]. A possible approach to refute the existence of such codes was suggested in [24]. In fact, [24] conjectures that such codes do not exist and proves this conjecture under quite "strong" assumptions. It is worth to mention that during last years a non-trivial effort was made in studying the limitations of LTCs, and in particular: LTCs testable with 2 queries [11, 39, 48, 47] (which is a severe restriction), random low density parity check (LDPC) codes [15], cyclic codes [4], symmetric codes [16, 20, 37, 46], LTCs with small redundancy among its tests [13] and dense LTCs [28, 24]. Nevertheless, it seems that we are very far from resolving this problem.

Let us suppose that asymptotically good LTCs with constant query complexity and constant soundness do not exist. In this case, the most intriguing question would be "What are the best LTCs we could obtain?". To address this question we should decide how to compare different LTCs. Informally, in this subject and in the area of error-correcting codes in general, we always require a constant relative distance since otherwise even a tiny fraction of errors could modify one codeword into another. Hence we want a constant relative distance and do not allow to relax this requirement. Given that we consider only LTCs with constant relative distance, we have 3 parameters that describe the "goodness" of LTCs: the query complexity, the soundness parameter and the rate.

Constant Soundness. It is not hard to show that LTCs with sub-constant soundness parameter ϵ and query complexity q could be converted to LTCs with soundness $\frac{1}{2}$ and query complexity $q \cdot \lceil \frac{1}{\epsilon} \rceil$ (see Claim 6.2). Hence, for the sake of this discussion we can require constant soundness parameter and compare different LTCs only according to their query complexity and the rate.

Constant Query Complexity. Recall that in Theorem 1.4 we show that when query complexity is required to be constant, the rate can be inverse polylogarithmic. Informally, under assumption that asymptotically good LTCs with constant query complexity and soundness do not exist, this is the best achievable rate when query complexity, relative distance and soundness parameter are required to be constant.

Constant Rate. Indeed, one of the most natural questions is what is the minimal query complexity if the rate and the soundness parameter of an LTC are required to be constant as well as the relative distance. In other words, what is the minimal query complexity required for the asymptotically good code to be testable. [3] For the current state of the art, we know that for every constant $\epsilon > 0$ there exist asymptotically good strong LTCs with query complexity n^{ϵ} and constant soundness parameter, where n is the blocklength of the code [18, 21, 54].

In Section 1.3.1 we show that under a conceivable conjecture there exist asymptotically good strong LTCs with query complexity $\text{polylog}(n)$ and constant soundness parameter. Informally, this is the minimal query complexity we can hope for, under conjecture that asymptotically good LTCs with constant query complexity and soundness do not exist [24].

[3] We think that this question is pretty much known in the area, but we do not aware if it was explicitly asked in the literature.

1.2 Preliminaries

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

For the distribution $\mathcal{D}$ over the subsets of $[n]$ we let $\mathcal{D}(I)$ to denote the probability that a subset $I \subseteq [n]$ is selected by $\mathcal{D}$ and $\mathrm{supp}(\mathcal{D}) = \{I \subseteq [n] \mid \mathcal{D}(I) > 0\}$. For $i \in [n]$ we let $N_{\mathcal{D}}(i) = \{I \in \mathrm{supp}(\mathcal{D}) \mid i \in I\}$.

Now we define testers and LTCs (see [36, 55] for the justification of this definition).

Definition 1.1 (LTCs and Testers)**.** A q-query tester for a code $\mathcal{C} \subseteq \mathbb{F}^n$ is a distribution $\mathcal{D}$ over subsets $I \subseteq [n]$ such that $|I| \leq q$. A q-query tester $\mathcal{D}$ is a (q, ϵ, ρ)-weak tester if for all $w \in \mathbb{F}^n$, $\delta(w, \mathcal{C}) \geq \rho$ we have $\Pr_{I\sim\mathcal{D}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon$. A q-query tester $\mathcal{D}$ is a (q, ϵ)-strong tester if for all $w \in \mathbb{F}^n$ we have $\Pr_{I\sim\mathcal{D}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon \cdot \delta(w, \mathcal{C})$.

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

Remark 1.2. Although the tester in Definition 1.1 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 $w \in \mathcal{C}$ and a subset $I \subseteq [n]$ is selected by the tester, the tester can output accept if $w|_I \in \mathcal{C}|_I$ and otherwise output reject. In this manner, the tester always accepts the codewords of $\mathcal{C}$.

1.3 Main Results

In this paper we resolve the following question raised by Goldreich and Sudan [36].

Question 1.3 ([36])**.** *Are there exist constants $q \in \mathbb{N}^+$, $d, \epsilon, \gamma > 0$ and a constant size alphabet Σ such that for infinitely many $n \in \mathbb{N}^+$ we have a code $C \subseteq \Sigma^n$, where*

- *C is a (q, ϵ)-strong LTC,*
- *$\delta(C) \geq \gamma$ and $\mathrm{rate}(C) \geq \frac{1}{\log^d(n)}$.*

Although the requested range of parameters was achieved for the weak LTCs [19, 27, 50], strong LTCs with these parameters were not obtained and this question remained to be a basic open question in the area of LTCs.

Our main theorem (Theorem 1.4) answers affirmatively on Question 1.3.

Theorem 1.4 (Main Theorem)**.** *There exist constants $d, \epsilon, \gamma > 0$ such that for infinitely many $n \in \mathbb{N}^+$ we have a linear code $C \subseteq \mathbb{F}_2^n$, where*

- C *is a* $(3, \epsilon)$*-strong LTC,*
- $\delta(C) \geq \gamma$ *and* $\text{rate}(C) \geq \frac{1}{\log^d n}$.

We notice that 3 queries are necessary to test non-trivial linear codes [11].[4]

The key ideas behind the proof of Theorem 1.4. The proof of Theorem 1.4 contains three stages.

Relaxed LTCs. First, we present in Section 2 a new notion of *relaxed LTCs*. Intuitively, relaxed LTCs have two kind of coordinates: those with good testability and those which worse (but non-trivial) testability (see Definition 2.2). Then, we present the first observation of this work (Observation 2.4) and its corollary (Corollary 2.5) in Section 2 saying that such relaxed LTCs can be easily converted to strong LTCs. Hence, all we need to resolve Question 1.3 is to construct relaxed LTCs with a corresponding range of parameters.

Relaxed LTCs to start with. We want to construct sufficiently nice relaxed LTCs. To achieve the required relaxed LTCs, our starting point is the main result of [55].[5] However, we cannot use directly the codes and the testers as that were suggested in [55], i.e., they should be slightly modified before the use. So, in Section 3 we recall the main result of [55] and make some immediate corollaries to conclude the relaxed LTCs (with inverse polylogarithmic soundness) we will use as a starting point in the proof of Theorem 1.4.

Gap Amplification can be applied to relaxed LTCs. We recall the well known "gap amplification" technique of Dinur [27] in Section 4. In Section 5 we show that the gap amplification and in particular, its version corresponding to linear codes [50] (see also [14]) can be applied to the linear relaxed LTCs to obtain linear relaxed LTCs with higher first soundness parameter (see Definition 2.2). The crucial observation here is that while the first soundness parameter is amplified by the gap amplification procedure, the second soundness parameter of these relaxed LTCs will not be reduced too much. This observation gives us a possibility to apply the gap amplification many times and to obtain linear relaxed LTCs, where the first soundness parameter is constant and the second soundness parameter is inverse polylogarithmic. Finally, these relaxed LTCs can be converted to the strong LTCs with a constant soundness and inverse polylogarithmic rate using Corollary 2.5.

1.3.1 Asymptotically Good LTCs with poly-log queries

We start this section by introducing a specific kind of junta with respect to a linear code. Intuitively, an (n', h)-junta with respect to a linear code $C \subseteq \mathbb{F}_2^n$ is a junta of size n' such that every code symbol outside this junta is determined by at most h code symbols of the junta.

Definition 1.5 (Junta)**.** Let $C \subseteq \mathbb{F}_2^n$ be a linear code and $T \subseteq [n]$ be a subset. We say that T is an (n', h)-junta with respect to C if $n' = |T|$ and for every $j \in [n] \setminus T$ we have $u_j \in C^\perp$ such that $j \in \text{supp}(u)$, $\text{supp}(u_j) \setminus \{j\} \subseteq T$ and $|\text{supp}(u_j) \setminus \{j\}| \leq h$.

[4] By "non-trivial" codes we mean codes with a constant relative distance and non-constant dimension.

[5] The codes presented in [55] were very similar to the codes of [50].

We notice that every linear code C has a $(\dim(C), \dim(C))$-junta. To see this assume without loss of generality that the generating matrix of C has a systematic form[6] and let $T = [\dim(C)]$.

Recall that Theorem 1.4 shows the existence of strong LTCs with constant query complexity, soundness, relative distance and inverse polylogarithmic rate. The following conjecture argues that strong LTCs with poly-log query complexity and inverse poly-log rate can be accompanied with a $(O(\dim(C)), \text{polylog}(n))$-junta.

Conjecture 1.6 (strong LTC with a junta)**.** *There exists a linear code $C \subseteq \mathbb{F}_2^n$ (for arbitrary large $n \in \mathbb{N}^+$) such that C is a $(\text{polylog}(n), \frac{1}{2})$-strong LTC, $\delta(C) = \Omega(1)$, $\text{rate}(C) \geq \frac{1}{\text{polylog}(n)}$ and a $(\Theta(\dim(C)), \text{polylog}(n))$-junta T with respect to C.*

Remark 1.7. The construction of strong LTCs presented in [55, Corollary 3.2] seems close to resolve Conjecture 1.6, but doesn't resolve it. Informally, this construction was obtained by execution $\Theta(\log\log(n))$ iterations (see Remark 3.2) and gave a $(\text{polylog}(n), \frac{1}{2})$-strong LTC C with $\delta(C) = \Omega(1)$ and $\text{rate}(C) \geq \frac{1}{\text{polylog}(n)}$. Each iteration 3 procedures were applied: the star product, the distance amplification and the random projection. A natural candidate for a $(\Theta(\dim(C)), \text{polylog}(n))$-junta would be the core $A(C)$ of the code $C \subseteq \mathbb{F}_2^n$ constructed in [55], which had a size $|A(C)| = \Theta(\dim(C))$, i.e., $\text{blocklength}(C|_{A(C)}) = \Theta(\dim(C))$. The problem is that only 2 procedures: the star product and the distance amplification preserved the required property, i.e., there exists a fixed constant $r \in \mathbb{N}^+$ such that if a core $A(C)$ of the input code C is a $(\Theta(\dim(C)), h)$-junta then the core $A(C')$ of the code C' obtained by these procedures is a $(\Theta(\dim(C)), r \cdot h)$-junta. Unfortunately, the random projection procedure does not preserve this property. However, if there exists a way to make this procedure preserving the "junta" property as another two procedures, then after the execution of $\Theta(\log\log(n))$ iterations we would get not only a $(\text{polylog}(n), \frac{1}{2})$-strong LTC C, but also a $(\Theta(\dim(C)), \text{polylog}(n))$-junta $A(C)$.

Under Conjecture 1.6 it is not hard to prove the existence of asymptotically good strong LTCs with polylogarithmic query complexity.

Theorem 1.8 (Asymptotically good LTCs with poly-log queries)**.** *Under Conjecture 1.6, there exists a linear $(\text{polylog}(n'), \frac{1}{2})$-strong LTC $C' \subseteq \mathbb{F}_2^{n'}$ (for arbitrary large n') such that $\delta(C') = \Omega(1)$ and $\text{rate}(C') = \Omega(1)$.*

The proof of Theorem 1.8 appears in Section 6.

2 Relaxed LTCs

Before we present Observation 2.4, we recall some concept used in [55].

Definition 2.1 (A core of the code)**.** Let $C \subseteq \Sigma^n$ be a code. A core of the code C, denoted by $A(C)$, is a nonempty subset of $[n]$ such that if $A(C) \neq [n]$ then any assignment to the entries of $A(C)$ uniquely determines the entries of $[n] \setminus A(C)$ and vice versa. I.e., if $A(C) \neq [n]$ then for any $c \in C$ there is no $c' \in C$ such that $c|_{A(C)} = c'|_{A(C)}$ and $c|_{[n]\setminus A(C)} \neq c'|_{[n]\setminus A(C)}$, or $c|_{[n]\setminus A(C)} = c'|_{[n]\setminus A(C)}$ and $c|_{A(C)} \neq c'|_{A(C)}$.

Clearly, there might be many options for $A(C)$, and in this case we fix only one such option. If $A(C) = [n]$ then for any $w, w' \in \Sigma^n$ we let $\delta(w|_{[n]\setminus A(C)}, w'|_{[n]\setminus A(C)}) = \delta(w|_{[n]\setminus A(C)}, C|_{[n]\setminus A(C)}) = 0$.

[6] Such generating matrix yields codewords whose first $\dim(C)$ symbols are message symbols.

Our first novelty is the following concept of a relaxed LTC (rLTC).

Definition 2.2 (Relaxed LTC)**.** A q-query tester $\mathcal{D}$ is a $(q, \epsilon_1, \epsilon_2)$-rLTC tester for a linear code $C \subseteq \mathbb{F}^n$ with a core $A(C)$, if for every $w \in \mathbb{F}^n$ there exists $c \in C$ such that $\mathop{\mathbf{Pr}}_{I\sim\mathcal{D}}[w|_I \notin C|_I] \geq \max\left\{\epsilon_1 \cdot \delta(w|_{A(C)}, c|_{A(C)}), \epsilon_2 \cdot \delta(w|_{-A(C)}, c|_{-A(C)})\right\}$. A code $C \subseteq \mathbb{F}^n$ with a core $A(C)$ is a $(q, \epsilon_1, \epsilon_2)$-*rLTC* if it has a $(q, \epsilon_1, \epsilon_2)$-rLTC tester.

The parameter q is called the query complexity, ϵ_1 is called the first soundness parameter and ϵ_2 is called the second soundness parameter.

Intuitively, think that ϵ_1 is a constant, but ϵ_2 is sub-constant.

Remark 2.3. We note that if $C \subseteq \mathbb{F}^n$ is a (q, ϵ)-strong LTC and $\mathcal{D}$ is its tester, then setting $A(C) = [n]$ it holds that C is a $(q, \epsilon, 1)$-rLTC with regards to the same tester $\mathcal{D}$ because for every $w \in \mathbb{F}^n$ we have

$$\mathop{\mathbf{Pr}}_{I\sim\mathcal{D}}[w|_I \notin C|_I] \geq \epsilon \cdot \delta(w, C) = \max\left\{\epsilon \cdot \delta(w|_{[n]}, C|_{[n]}), 1 \cdot 0\right\} =$$

$$= \max\left\{\epsilon \cdot \delta(w|_{A(C)}, C|_{A(C)}), 1 \cdot \delta(w|_{-A(C)}, C|_{-A(C)})\right\}.$$

Our first observation in this work is that a relaxed LTC with sub-constant second soundness parameter can be easily converted to a strong LTC with a constant soundness.

Observation 2.4 (A conversion of rLTCs to strong LTCs)**.** *Let $q \geq 2$ and $C \subseteq \mathbb{F}^n$ be a linear $(q, \epsilon_1, \epsilon_2)$-rLTC with a core $A(C)$. Then there exists a linear $(q, \epsilon_1/6)$-strong LTC $C' \subseteq \mathbb{F}^{n'}$, where $n \leq n' \leq \frac{12}{\epsilon_2} \cdot n$, $\dim(C') = \dim(C)$, $\mathrm{rate}(C') \geq \frac{\epsilon_2}{12} \cdot \mathrm{rate}(C)$ and $\delta(C') \geq 0.9 \cdot \delta(C|_{A(C)})$. Moreover, the construction of C' from C is explicit and done in time $O(n')$.*

Proof. Let $\mathcal{D}$ be the corresponding tester for C. Without loss of generality assume that $A(C)$ is the first $|A(C)|$ indexes, i.e., $A(C) = [|A(C)|]$. If $A(C) = [n]$ then C is a (q, ϵ_1)-strong LTC with regards to the same tester $\mathcal{D}$ and we are done. Thus we assume for the rest of the proof that $A(C) \subsetneq [n]$ and thus $n - |A(C)| \neq 0$.

Let $h = \left\lceil \frac{10}{\epsilon_2} \cdot \frac{n}{|A(C)|} \right\rceil$ and note that $h \geq 1$. Then $\frac{10}{\epsilon_2} \cdot n \leq h \cdot |A(C)| \leq \frac{11}{\epsilon_2} \cdot n$. Now, let C' be a linear code obtained from C by concatenating the symbols of $C|_{A(C)}$ to C exactly h times. I.e., for every $c' \in C'$ we have $c' = (c, (\overbrace{c|_{A(C)}, c|_{A(C)}, \ldots, c|_{A(C)}}^{h}))$ for some $c \in C$. In particular, $C = C'|_{[n]}$ and for every $j \in [h]$ we have $C'|_{n+(j-1)\cdot|A(C)|+[|A(C)|]} = C|_{A(C)}$. We set $A(C') = A(C)$.

We notice that $\dim(C') = \dim(C)$ and the blocklength of C' is $n' = n + h \cdot |A(C)| \leq n + \frac{11}{\epsilon_2} \cdot n \leq \frac{12}{\epsilon_2} \cdot n$. It can be verified that $\delta(C') \geq \frac{(h+1)\cdot|A(C)|\cdot\delta(C|_{A(C)})}{n'} \geq 0.9 \cdot \delta(C|_{A(C)})$. Note also that $\mathrm{rate}(C') = \frac{\dim(C)}{n'} \geq \frac{\epsilon_2}{12} \cdot \mathrm{rate}(C)$.

Let $\mathcal{D}'$ be a tester for C' that on the input word $w \in \mathbb{F}^{(n')}$:

- picks $r \in \{0, 1\}$
- if $r = 1$ then picks random $j_1 \in A(C)$ and $j_2 \in [h]$ and outputs $\{j_1, n + (j_2 - 1) \cdot |A(C)| + j_1\}$
- otherwise ($r = 2$), samples $\mathcal{D}$ on $w|_{[n]}$ and returns its output

We argue that C' is a $(q, \epsilon_1/6)$-strong LTC with respect to its tester $\mathcal{D}'$. Clearly, $\mathcal{D}'$ is a q-query tester. Let $w \in \mathbb{F}^{n'}$ be an input word. We show that

$$\mathop{\mathbf{Pr}}_{I\sim\mathcal{D}'}\left[w|_I \notin C'|_I\right] \geq \frac{\epsilon_1}{6} \cdot \delta(w, C').$$

Let β be the fraction of bits in $w|_{[n']\setminus[n]}$ that is not equal to the corresponding bits of $w|_{A(C)}$, i.e., $\beta = \mathop{\mathbf{Pr}}_{j_1\in A(C), j_2\in[h]}\left[w|_{j_1} \neq w|_{n+(j_2-1)\cdot|A(C)|+j_1}\right]$. We know that

$$\mathop{\mathbf{Pr}}_{I\sim\mathcal{D}'}\left[w|_I \notin \mathcal{C}'|_I\right] \geq \frac{\beta}{2} + \frac{\mathbf{Pr}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin C|_I\right]}{2}. \tag{1}$$

By assumption, C is a $(q, \epsilon_1, \epsilon_2)$-rLTC and hence there exists a codeword $c \in C$ such that $\mathop{\mathbf{Pr}}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin \mathcal{C}|_I\right] \geq \max\left\{\epsilon_1 \cdot \delta((w|_{[n]})|_{A(C)}, c|_{A(C)}), \epsilon_2 \cdot \delta((w|_{[n]})|_{-A(C)}, c|_{-A(C)})\right\}$. Therefore

$$\delta((w|_{[n]})|_{A(C)}, c|_{A(C)}) \leq \frac{\mathbf{Pr}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin \mathcal{C}|_I\right]}{\epsilon_1} \tag{2}$$

and

$$\delta((w|_{[n]})|_{-A(C)}, c|_{-A(C)}) \leq \frac{\mathbf{Pr}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin \mathcal{C}|_I\right]}{\epsilon_2}. \tag{3}$$

Let $c' \in C'$ be the corresponding codeword to c (i.e., $c'|_{[n]} = c$). Then (by (1), (2) and (3))

$$\begin{aligned}
\delta(w, c') &\leq \beta + \frac{\delta((w|_{[n]})|_{A(C)}, c|_{A(C)}) \cdot |A(C)| \cdot (h+1) + \delta((w|_{[n]})|_{-A(C)}, c|_{-A(C)}) \cdot (n - |A(C)|)}{n'} \leq \\
&\leq \beta + \frac{(1/\epsilon_1) \cdot |A(C)| \cdot (h+1) + (1/\epsilon_2) \cdot n}{n'} \cdot \mathop{\mathbf{Pr}}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin \mathcal{C}|_I\right] \leq \\
&\leq \beta + \frac{(1/\epsilon_1) \cdot \frac{12}{\epsilon_2} \cdot n + (1/\epsilon_2) \cdot n}{n'} \cdot \mathop{\mathbf{Pr}}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin \mathcal{C}|_I\right] \leq \\
&\leq \beta + \frac{(1/\epsilon_1) \cdot \frac{12}{\epsilon_2} \cdot n + (1/\epsilon_2) \cdot n}{(10/\epsilon_2) \cdot n} \cdot \mathop{\mathbf{Pr}}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin \mathcal{C}|_I\right] \leq \\
&\leq \beta + \frac{3}{\epsilon_1} \cdot \mathop{\mathbf{Pr}}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin \mathcal{C}|_I\right] \leq \frac{6}{\epsilon_1} \cdot \left(\frac{\beta}{2} + \frac{\mathbf{Pr}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin C|_I\right]}{2}\right) \leq \frac{6}{\epsilon_1} \cdot \mathop{\mathbf{Pr}}_{I\sim\mathcal{D}'}\left[w|_I \notin C'|_I\right].
\end{aligned}$$

That means $\delta(w, c') \leq \frac{6}{\epsilon_1} \cdot \mathbf{Pr}_{I\sim\mathcal{D}'}[w|_I \notin C'|_I]$, where $c' \in C'$. We conclude that

$$\mathop{\mathbf{Pr}}_{I\sim\mathcal{D}'}\left[w|_I \notin \mathcal{C}'|_I\right] \geq \frac{\epsilon_1}{6} \cdot \delta(w, C').$$

This proves that C' is a $(q, \epsilon_1/6)$-strong LTC with respect to its tester $\mathcal{D}'$, and completes the proof of Observation 2.4. □

Although Observation 2.4 might seem naive, it implies the following corollary that will play a crucial role in the proof of Theorem 1.4.

Corollary 2.5. *Assume that for constants $q \geq 2, \epsilon > 0$ and infinitely many $n \in \mathbb{N}^+$ we have a linear code $C \subseteq \mathbb{F}_2^n$ with a core $A(C)$ such that C is a $(q, \epsilon, \frac{1}{\text{polylog}(n)})$-rLTC, $\delta(C|_{A(C)}) = \Omega(1)$ and $\text{rate}(C) = \frac{1}{\text{polylog}(n)}$. Then, there exists $C' \subseteq \mathbb{F}_2^{n'}$ such that $n \leq n' \leq n \cdot \text{polylog}(n)$, C' is a $(q, \epsilon/6)$-strong LTC, $\delta(C') = \Omega(1)$ and $\text{rate}(C') = \frac{1}{\text{polylog}(n')}$ (i.e., Question 1.3 is solved).*

Proof. The construction of the required C' follows immediately from Observation 2.4. □

3 The main result of [55] and its corollaries

In this section we recall the main result of [55]. Then we make some corollaries that will be used later.

Theorem 3.1 ([55]). *For some constant $d \in \mathbb{N}^+$ and infinitely many $n \in \mathbb{N}^+$ there exists a linear code $C \subseteq \mathbb{F}_2^n$ and its tester $\mathcal{D}$ such that*

- *C is a $(3, \frac{1}{\log^d n})$-strong LTC with respect to $\mathcal{D}$,*
- *$\delta(C) = \Omega(1)$,*
- *$\mathrm{rate}(C) = \frac{1}{\log^d n}$,*
- *$|\mathrm{supp}(\mathcal{D})| \leq n \log^d n$ and for every $u \in \mathrm{supp}(\mathcal{D})$ it holds that $\mathcal{D}(u) \leq \frac{\log^d n}{n}$, and*
- *for every $i \in [n]$ we have $|N_{\mathcal{D}}(i)| \leq \log^d n$.*

Remark 3.2. Although in [55] two last bullets were not proved, but one could verify that these bullets hold. The construction in [55] begins from a constant blocklength code C_1 and contained 3 procedures: the star product, the random projection and the distance amplification. These 3 procedures were applied iteratively $\Theta(\log \log n)$ times. Each iteration i is executed on the code C_i that had a blocklength n_i and a tester $\mathcal{D}_i$. The output of each iteration i is the code C_{i+1}.

Initially, the base code $C_1 \subseteq \mathbb{F}_2^{n_1}$ and its tester $\mathcal{D}_1$ satisfied the last two bullets with respect to its blocklength $n_1 = O(1)$. I.e., $|\mathrm{supp}(\mathcal{D}_1)| \leq n_1 \cdot O(1)$ and for every $u \in \mathrm{supp}(\mathcal{D}_1)$ it holds that $\mathcal{D}_1(u) \leq \frac{O(1)}{n_1}$. Moreover, for every $i \in [n_1]$ we have $|N_{\mathcal{D}_1}(i)| \leq O(1)$.

Each iteration, the star product, the random projection and the distance amplification procedures were applied. The random projection does not affect the properties listed in these bullets, but only rearranges the coordinates of the given code in some way. The star product and the distance amplification procedure do affect the properties listed in these bullets, but only by fixed multiplicative constants.

More formally, there exists a fixed constant $h > 0$ such that the following occur. Suppose that in the iteration i for some $h_i > 0$ we have the code $C_i \subseteq \mathbb{F}^{n_i}$ and its tester $\mathcal{D}_i$ such that $\mathrm{supp}(\mathcal{D}_i) \leq h_i \cdot n_i$, for every $u \in \mathrm{supp}(\mathcal{D}_i)$ it holds that $\mathcal{D}_i(u) \leq \frac{h_i}{n_i}$, and for every $j \in [n_i]$ it holds that $|N_{\mathcal{D}_i}(j)| \leq h_i$. Then, after the star product (or distance amplification) is applied, resulting in the code $C_{i+1} \subseteq \mathbb{F}^{n_{i+1}}$ and its tester $\mathcal{D}_{i+1}$, we have $\mathrm{supp}(\mathcal{D}_{i+1}) \leq h \cdot h_i \cdot n_{i+1}$, for every $u \in \mathrm{supp}(\mathcal{D}_{i+1})$ it holds that $\mathcal{D}_{i+1}(u) \leq \frac{h \cdot h_i}{n_{i+1}}$, and for every $j \in [n_{i+1}]$ it holds that $|N_{\mathcal{D}_{i+1}}(j)| \leq h \cdot h_i$.

Therefore, after $\Theta(\log \log n)$ iterations we obtain the code $C \subseteq \mathbb{F}^n$ and its tester $\mathcal{D}$ such that $|\mathrm{supp}(\mathcal{D})| \leq n \cdot \mathrm{polylog}(n)$, for every $u \in \mathrm{supp}(\mathcal{D})$ it holds that $\mathcal{D}(u) \leq \frac{\mathrm{polylog}(n)}{n}$, and for every $j \in [n]$ it holds that $|N_{\mathcal{D}}(j)| \leq \mathrm{polylog}(n)$.

We pay attention that one can turn the strong LTCs of Theorem 3.1 to the strong LTCs with a uniform distribution over the tests, and the soundness parameter, roughly speaking, will be preserved.

Corollary 3.3. *For some constant $d \in \mathbb{N}^+$ and infinitely many $n \in \mathbb{N}^+$ there exist a linear code $C' \subseteq \mathbb{F}_2^n$ and its tester $\mathcal{D}'$ which is a uniform distribution over $\mathrm{supp}(\mathcal{D}')$ such that*

- *C' is a $(3, \frac{1}{\log^d n})$-strong LTC with respect to $\mathcal{D}'$,*

- $\delta(C') = \Omega(1)$,
- $\mathrm{rate}(C') \geq \frac{1}{\log^d n}$,
- $|\mathrm{supp}(\mathcal{D}')| \leq n \log^d n$, *and*
- *for every* $i \in [n]$ *we have* $|N_{\mathcal{D}'}(i)| \leq \log^d n$.

Proof. Corollary 3.3 follows from Theorem 3.1 by letting $C' = C$ and $\mathcal{D}'$ be a uniform distribution over $\mathrm{supp}(\mathcal{D})$. Note that $\mathrm{supp}(\mathcal{D}') = \mathrm{supp}(\mathcal{D})$ and for every $u \in \mathrm{supp}(\mathcal{D}')$ we have

$$\mathcal{D}'(u) = \frac{1}{|\mathrm{supp}(\mathcal{D})|} \geq \frac{1}{n \log^d n} = \frac{1}{\log^{2d} n} \cdot \frac{\log^d n}{n} \geq \frac{1}{\log^{2d} n} \cdot \mathcal{D}(u).$$

Then, for any $w \in \mathbb{F}_2^n$ we have

$$\mathop{\mathbf{Pr}}_{u \sim \mathcal{D}'}[\langle u, w \rangle \neq 0] \geq \frac{1}{\log^{2d} n} \cdot \mathop{\mathbf{Pr}}_{u \sim \mathcal{D}}[\langle u, w \rangle \neq 0] \geq \frac{1}{\log^{2d} n} \cdot \frac{1}{\log^d n} \cdot \delta(w, C) = \frac{1}{\log^{3d} n} \cdot \delta(w, C).$$

□

Now, in Corollary 3.4 we show that the 3-query strong LTCs over $\mathbb{F}_2$ from Corollary 3.3 can be easily converted to the 2-query rLTCs over $\mathbb{F}_2^3$ with a similar range of parameters.[7] This conversion is standard (for the case of LTCs, PCPs and assignment testers) and was explained, e.g., in [27, 50].

Corollary 3.4. *For some constant* $d \in \mathbb{N}^+$ *and infinitely many* $n'' \in \mathbb{N}^+$ *there exist a code* $C'' \subseteq (\mathbb{F}_2^3)^{(n'')}$ *and its tester* $\mathcal{D}''$ *which is a uniform distribution over* $\mathrm{supp}(\mathcal{D}'')$ *such that*

- C'' *is linear over* $\mathbb{F}_2$,
- C'' *is a* $(2, \frac{1}{3\log^d n''}, \frac{1}{6\log^{2d} n''})$*-rLTC with respect to its core* $A(C'')$ *and* $\mathcal{D}''$,
- $\delta(C''|_{A(C'')}) = \Omega(1)$,
- $\mathrm{rate}(C'') \geq \frac{1}{2\log^{2d} n''}$,
- $|\mathrm{supp}(\mathcal{D}'')| \leq 3 \cdot n''$, *and*
- *for every* $i \in [n'']$ *we have* $|N_{\mathcal{D}''}(i)| \leq \log^d n''$.

Proof. Corollary 3.3 implies the existence of a constant $d \in \mathbb{N}^+$, a linear code $C' \subseteq \mathbb{F}_2^n$ for arbitrary large n and its tester $\mathcal{D}'$ such that C' is a $(3, \frac{1}{\log^d n})$-strong LTC with respect to $\mathcal{D}'$, $\delta(C') = \Omega(1)$, $\mathrm{rate}(C') \geq \frac{1}{\log^d n}$, $|\mathrm{supp}(\mathcal{D}')| \leq n \log^d n$, and for every $i \in [n]$ we have $|N_{\mathcal{D}'}(i)| \leq \log^d n$.

Let every element of $\mathbb{F}_2^3$ be associated with a tuple of 3 bits. Let also every bit (an element of $\mathbb{F}_2$) be viewed as an element of $\mathbb{F}_2^3$. Let us create a separate code symbol $X_{(i_1,i_2,i_3)}$ for every 3-query test $\{i_1, i_2, i_3\}$ of the original code (similarly, we create $X_{(i_1,i_2)}$ and $X_{(i_1)}$ for every 2-query and 1-query tests). Call the new code C''. A new tester $\mathcal{D}''$ for C'' samples an original tester $\mathcal{D}'$, and if $\mathcal{D}'$ queries i_1, i_2, i_3, then the tester $\mathcal{D}''$ queries the entry $X_{(i_1,i_2,i_3)}$ and a random coordinate

[7] During this paper we associate $\mathbb{F}_2^3$ with $\mathbb{F}_{2^3}$.

$i_j \in \{i_1, i_2, i_3\}$, and accepts iff three bits of $X_{(i_1,i_2,i_3)}$ are summed to 0 and the corresponding bit of $X_{(i_1,i_2,i_3)}$ is equal to i_j.

We let $A(C'') = [n]$ and note that $C'' \subseteq (\mathbb{F}_2^3)^{(n'')}$. One can easily verify that C'' is linear over $\mathbb{F}_2$, $\delta(C''|_{A(C'')}) = \delta(C') = \Omega(1)$, $\mathrm{rate}(C'') = \frac{\dim(C'')}{n''} = \frac{\dim(C')}{n''} \geq \frac{\dim(C')}{n+n\log^d n} \geq \frac{1}{2\log^{2d} n''}$, $|\operatorname{supp}(\mathcal{D}'')| \leq 3 \cdot n''$, where $n'' \leq n + n \cdot \log^d n$. Moreover, for every $i \in [n'']$ we have $|N_{\mathcal{D}''}(i)| \leq \log^d n''$.

We prove that C'' is a $(2, \frac{1}{3\log^d n''}, \frac{1}{\log^{2d} n''})$-rLTC with respect to its core $A(C'')$ and $\mathcal{D}''$. Let $w \in (\mathbb{F}_2^3)^{(n'')}$. Note that $w|_{[n]}$ can be viewed as a word in $\mathbb{F}_2^n$. Since C' is a $(3, \frac{1}{\log^d n})$-strong LTC with respect to $\mathcal{D}'$, there exists $c' \in C'$ such that

$$\mathop{\mathbf{Pr}}_{I \sim \mathcal{D}'}\left[(w|_{[n]})|_I \notin C'|_I\right] \geq \frac{1}{\log^d n} \cdot \delta(w|_{[n]}, c').$$

Let $c'' \in C''$ be a corresponding codeword to c', i.e., $c''|_{[n]} = c''|_{A(C'')} = c'$. Then,

$$\mathop{\mathbf{Pr}}_{I \sim \mathcal{D}''}\left[w|_I \notin C''|_I\right] \geq \frac{1}{3} \cdot \mathop{\mathbf{Pr}}_{I \sim \mathcal{D}'}\left[(w|_{[n]})|_I \notin C'|_I\right] \geq \frac{1}{3\log^d n} \cdot \delta(w|_{[n]}, c') \geq \frac{1}{3\log^d n''} \cdot \delta(w|_{A(C'')}, c''|_{A(C'')}),$$

where we used the fact that every constraint selected by $\mathcal{D}'$ contains at most 3 symbols (e.g., $\{i_1, i_2, i_3\}$ for $i_1, i_2, i_3 \in [n]$), and hence if it is not satisfied ($w|_{\{i_1,i_2,i_3\}} \notin C'|_{\{i_1,i_2,i_3\}}$), then at least one of the corresponding 3 constraints selected by $\mathcal{D}''$ is not satisfied. Hence the first soundness parameter is decreased by $\frac{1}{3}$.

On the other hand,

$$\delta(w|_{-A(C'')}, c''|_{-A(C'')}) \leq 3 \cdot \mathop{\mathbf{Pr}}_{I \sim \mathcal{D}''}\left[w|_I \notin C''|_I\right] + (\log^d n) \cdot \delta(w|_{[n]}, c''|_{[n]}).$$

To see this, we recall that the code C'' was constructed from C' by adding the entries indexed by $[n''] \setminus [n]$ that simulates the constraints checked by $\mathcal{D}'$. Thus an entry of $w|_{-A(C'')}$ should be modified only if it contains a non-consistent value, i.e., it does not correspond to the symbols contained by the constraint it simulates (the fraction of such entries is at most $3 \cdot \mathbf{Pr}_{I \sim \mathcal{D}''}[w|_I \notin C''|_I]$) or this entry simulates a constraint containing a symbol of $w|_{[n]}$ that should be modified (the fraction of such entries is at most $(\log^d n) \cdot \delta(w|_{[n]}, c''|_{[n]})$). Here we used the fact that a change of a symbol indexed by $[n]$ of C'' might yield a change of at most $\log^d n$ entries (that simulates constraints) indexed by $[n''] \setminus [n]$ since for every $i \in [n]$ we have $|N_{\mathcal{D}'}(i)| \leq \log^d n$ with respect to the code C' and its tester $\mathcal{D}'$.

Furthermore,

$$3 \cdot \mathop{\mathbf{Pr}}_{I \sim \mathcal{D}''}\left[w|_I \notin C''|_I\right] + (\log^d n) \cdot \delta(w|_{[n]}, c''|_{[n]}) \leq$$

$$3 \cdot \mathop{\mathbf{Pr}}_{I \sim \mathcal{D}''}\left[w|_I \notin C''|_I\right] + (\log^d n) \cdot (3\log^d n) \cdot \mathop{\mathbf{Pr}}_{I \sim \mathcal{D}''}\left[w|_I \notin C''|_I\right] \leq (6\log^{2d} n) \cdot \mathop{\mathbf{Pr}}_{I \sim \mathcal{D}''}\left[w|_I \notin C''|_I\right],$$

where the last inequality holds for $n \geq 2$. Thus

$$\mathop{\mathbf{Pr}}_{I \sim \mathcal{D}''}\left[w|_I \notin C''|_I\right] \geq \max\left\{\frac{1}{3\log^d n''} \cdot \delta(w|_{A(C'')}, c''|_{A(C'')}), \frac{1}{6\log^{2d} n''} \cdot \delta(w|_{-A(C'')}, c''|_{-A(C'')})\right\}.$$

□

4 Gap Amplification Procedure for LTCs

In this section we describe the main result of Dinur [27] and its affect on the locally testable codes. We notice that two interesting alternatives were proposed. Radhakrishnan [51] suggested another option for the amplification lemma [27], where he used lazy random walks in the constraints graph. In particular, this suggestion improves some of the constants inside the Dinur's results. Goldreich and Meir [34] pointed out on a small gap in the proof of the amplification of assignment testers in [27]. Namely, while Dinur [27] argued that every an execution of the gap amplification costs a linear blowup for the underlying graph size, Goldreich and Meir [34] showed that sometimes this blowup can be larger and showed how one can easily correct this to have always only a linear blowup.

Nevertheless, in our paper we don't need the mentioned suggestions/corrections and we address the original work of Dinur [27]. The only modification we need is that the linearity of the underlying code can be preserved if the alphabet reduction stage in the gap amplification will be done by the concatenation with the Hadamard code [50, Section 6.4.3], and not by a general assignment testers composition as in [27]. Now we recall the gap amplification procedure [27] and describe how it is applied on the linear codes, and in particular to the 2-query linear relaxed LTCs.

A 2-query LTC can be associated with a constraints graph. In this section let $\mathbb{F} = \mathbb{F}_2^3$. Assume $C \subseteq \mathbb{F}^n$ has a 2-query tester $\mathcal{D}$. Let $G = (V, E)$ be an undirected graph, where $V = [n]$ and $\{i, j\} \in E$ if and only if $\mathcal{D}(\{i, j\}) > 0$. The degree of a symbol i of the code C is associated to the degree of the node i in the graph G, and equal to $|\{j \in [n] \mid \mathcal{D}(\{i, j\}) > 0\}|$.

The gap amplification for a relaxed LTC. Let us recall how the gap amplification would be applied on a relaxed LTC. This will be almost identical to the execution of the gap amplification on the assignment testers [27], where a single modification is that the alphabet reduction is done by the concatenation with the Hadamard code as was explained in [50, Section 6.4.3]. Assume that the input is the relaxed LTC $C \subseteq \mathbb{F}^n$ and its 2-query tester $\mathcal{D}$ such that $\mathcal{D}$ is uniform over $\mathrm{supp}(\mathcal{D})$. Assume that $A(C) \subseteq [n]$ is the core of the code C and without loss of generality assume that $A(C) = [|A(C)|]$, i.e., the core of the code is the first $|A(C)|$ coordinates. It is important to note that during the execution of the gap amplification the symbols of the core will be preserved in every stage of this procedure.

Let G be the graph corresponding to the code C and its 2-query tester $\mathcal{D}$. The gap amplification procedure contains the following three stages.

First stage - Preprocessing (described in [27, Section 4]) Assume that a coordinate i in C has degree d_i with respect to the tester $\mathcal{D}$. Let $d = \max_{i \in [n]} d_i$. The code $C \subseteq \mathbb{F}^n$ and its tester $\mathcal{D}$ are transformed to the new code $C' \subseteq \mathbb{F}^{n'}$ and its tester $\mathcal{D}'$ such that $n \leq n' \leq d \cdot n$ and $C'|_{[n]} = C$, i.e., all old code entries are preserved and some new code entries are added. It also holds that $\mathcal{D}'$ is uniform over a new collection of 2-query constraints, i.e., $\mathcal{D}'$ is uniform over $\mathrm{supp}(\mathcal{D}')$. The new entries are added by duplicating some original entries. The number of 2-query tests in $\mathcal{D}'$ is $|\mathrm{supp}(\mathcal{D}')| = O(d \cdot |\mathrm{supp}(\mathcal{D})|)$. The degree of every index i in the code C' with respect to the tester $\mathcal{D}'$ is a fixed constant (independent of any parameters). If the code C' is associated with a graph G', then G' is a constant degree expander graph (see [27, Section 4]).

We set the core of the code C' to be $A(C') = A(C)$ and note that $C|_{A(C)} = C'|_{A(C')}$, i.e., the core symbols are preserved. Similarly to the proof presented by Dinur [27], one could verify that if

for every $w \in \mathbb{F}^n$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}}[w|_I \notin C|_I] \geq \epsilon_1 \cdot \delta(w|_{A(C)}, C|_{A(C)})$, then for every $w \in \mathbb{F}^{n'}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}'}[w|_I \notin C'|_I] \geq (\epsilon_1/d) \cdot \delta(w|_{A(C')}, C'|_{A(C')})$. In particular, this means that if C is a $(2, \epsilon_1, \cdot)$-rLTC with respect to its tester $\mathcal{D}$ and its core $A(C)$, then C' is a $(2, \epsilon_1/d, \cdot)$-rLTC with respect to its tester $\mathcal{D}'$ and its core $A(C')$. I.e., the decrease in the first soundness parameter is bounded by a maximal degree of a code coordinate.

Notice that if all degrees of the code coordinates are upper-bounded by a fixed constant, then the soundness will be decreased be only by a constant.

Second Stage - Amplification (described in [27, Sections 1 and 6]) The input of this stage is the code $C' \subseteq \mathbb{F}^{n'}$ and its 2-query tester $\mathcal{D}'$ which is uniform over its support. We recall that $A(C') \subseteq [n]$ is a core of the code C' such that $A(C') = [|A(C')|]$. We also know that the degree of every index of C' (after the first stage) with respect to $\mathcal{D}'$ is equal to a fixed constant $d \in \mathbb{N}^+$. In this stage we associate the code C' and the tester $\mathcal{D}'$ with a graph G'. Then the graph G' is transformed to the graph $(G')^t$ for sufficiently large constant $t \in \mathbb{N}^+$, where $(G')^t$ has the same vertexes as G' and the edge set $(E')^t$ contains k parallel edges between i_1 and i_2 if and only if the number of t-step walks from i_1 to i_2 is exactly k. The graph $(G')^t$ defines the code C'' and its tester $\mathcal{D}''$ which is uniform over all edges of the graph $(G')^t$, but the first $|A(C')|$ symbols are exactly the first $|A(C')|$ symbols of C', i.e., the core coordinates are preserved. We set $A(C'') = A(C')$. The underlying field of the code C'' is $\mathbb{F}^{d^{\lceil t/2 \rceil}}$, but the symbols indexed by $A(C'')$ belong to $\mathbb{F}$. Note that C'' is linear over $\mathbb{F}$. In particular, the blocklength of C'' is n' and $|\operatorname{supp}(\mathcal{D}'')| \leq |\operatorname{supp}(\mathcal{D}')| \cdot d^t$.

In [27] it is shown that the first soundness parameter is increased in $t' = \Omega(\sqrt{(t)})$, where the constant inside $\Omega(\cdot)$ is independent of t, and hence t is picked to be sufficiently large constant such that, e.g., $t' \geq 10$. As we mentioned, Radhakrishnan [51] improved the dependency on t, but we don't use his result in this paper.

That means if for every $w \in \mathbb{F}^{n'}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}'}[w|_I \notin C'|_I] \geq \epsilon_1 \cdot \delta(w|_{A(C')}, C|_{A(C')})$, then for every $w \in \mathbb{F}^{n''}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}''}[w|_I \notin C''|_I] \geq (\epsilon_1 \cdot t') \cdot \delta(w|_{A(C'')}, C'|_{A(C'')})$. Namely, if C' was a $(2, \epsilon_1, \cdot)$-rLTC with respect to $A(C')$ and $\mathcal{D}'$ then C'' is a $(2, t' \cdot \epsilon_1, \cdot)$-rLTC with respect to $A(C'')$ and $\mathcal{D}''$, where ϵ_1 is less than some fixed constant $\gamma > 0$.

Third Stage - Alphabet Reduction (described in [27, Sections 1 and 5]) and [50, Section 6.4.3] In this stage, we will use the suggestion of Meir [50, Section 6.4.3], where the alphabet reduction is done by the concatenation with the binary Hadamard code. In this stage, every code symbol, which is an element of $\mathbb{F}^{d^t}$ for some $d, t \in \mathbb{N}^+$, is encoded by the Hadamard code over the field $\mathbb{F}_2$. The required testability is preserved due to the fact that the Hadamard code is a 3-query strong LTC and a 2-query locally correctable code (LCC) (see Section A). The output of this stage is a code $C''' \subseteq \mathbb{F}^{n'''}$ and its 2-query tester $\mathcal{D}'''$ which is uniform over its support. The core of the code is preserved again, and we set $A(C''') = A(C'')$ and note that $C''|_{A(C'')} = C'''|_{A(C''')}$.

As was explained in [50, Section 6.4.3], this reduction decreases rejection probability by a fixed constant $g > 0$ (independent of the parameters of the code), i.e., if for every $w \in \mathbb{F}^{n''}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}''}[w|_I \notin C''|_I] \geq \epsilon_1 \cdot \delta(w|_{A(C'')}, C|_{A(C'')})$, then for every $w \in \mathbb{F}^{n'''}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}'''}[w|_I \notin C'''|_I] \geq (\epsilon_1/g) \cdot \delta(w|_{A(C''')}, C'|_{A(C''')})$. Namely, if C'' was a $(2, \epsilon_1, \cdot)$-rLTC with respect to $A(C'')$ and $\mathcal{D}''$ then C''' is a $(2, \epsilon_1/g, \cdot)$-rLTC with respect to $A(C''')$ and $\mathcal{D}'''$.

An interesting point is that there are two options:

1. to obtain the binary linear code C''', where $\mathcal{D}'''$ is a 3-query tester. As was said, this is done simply by the concatenation with the binary Hadamard code (see [50, Section 6.4.3]).

2. to obtain the code C''' over the field $\mathbb{F}$, where $\mathcal{D}'''$ is a 2-query tester. This can be done by applying the first bullet and then turn the 3-query rLTC over $\mathbb{F}_2$ to the 2-query rLTC over $\mathbb{F} = \mathbb{F}_2^3$ using the standard technique (as in Corollary 3.4).

This gap amplification procedure will be applied a number of times. Each iteration, besides the last one, we use the second bullet, i.e., we obtain a 2-query rLTC over $\mathbb{F}$ (which is linear over $\mathbb{F}_2$). This code can be passed to the new iteration of gap amplification. However, in the last iteration we choose the first bullet and obtain a binary linear 3-query rLTC.

Overall, the output of the gap amplification procedure is the code C''' and its tester $\mathcal{D}'''$.

5 Proof of Theorem 1.4

In this section we state and prove Theorem 1.4.

Recall that Dinur [27] suggested the gap amplification procedure to increase the rejection probability of verifiers for PCPs and explained that this procedure fits also for the assignment testers [29] (or alternatively, PCPs of proximity [10]). Then, Meir [50] explained that exploring the fact that the Hadamard code is 3-query strong LTC and 2-query locally correctable one can use it (instead of the PCP composition) to reduce the alphabet size in the alphabet reduction stage of the gap amplification procedure. Hence, as we recall in Section 4, the gap amplification procedure can be applied to the linear codes and preserve their linearity. Let us denote by Amplify$(\cdot)$ this version of the gap amplification procedure.

Our contribution here is that we observe that when Amplify$(\cdot)$ is invoked on a relaxed LTC, it improves the first soundness parameter, while does not destroy the second soundness parameter too much. First, we summarize the known affect of Amplify$(\cdot)$ on $\mathbb{F}_2$-linear codes with 2-query testers in Theorem 5.1. Before we state Theorem 5.1. We need to define some auxiliary concepts.

Let $C \subseteq \mathbb{F}^n$ be a code. Let $m \in [n]$ and $J = \{1, 2, \ldots, m\}$. Assume that given a message first of all we compute the values for all coordinates indexed by J, and then based on their values we compute the rest of the coordinates. That means for every $i \in [n] \setminus J$ there exists a function $g_i : \mathbb{F}^{d_i} \to \mathbb{F}$ and $j_1, j_2, \ldots, j_{d_i} \in J$ such that for every $c \in C$ it holds that $c|_i = g_i(c|_{j_1}, c|_{j_2}, \ldots, c|_{j_{d_i}})$. In this case we say that the index i depends only on $j_1, j_2, \ldots, j_{d_i}$. Also, for $k \in [d_i]$ we say that i depends on j_k. We say that i depends on at most d entries of J if $d_i \leq d$. In the given case, for any $k \in [d_i]$ we say that j_k affects i. We say that each entry in J of C affects at most h entries in $[n] \setminus J$ if for all $j \in J$ it holds that j affects at most h entries in $[n] \setminus J$.

Theorem 5.1 (Implicit in [27] and [50])**.** *Let $\mathbb{F} = \mathbb{F}_{2^3}$. There exist constants $h \in \mathbb{N}^+$ and $\gamma > 0$ such that the following holds. Let $C \subseteq \mathbb{F}^n$ be a code (linear over $\mathbb{F}_2$) with a 2-query tester $\mathcal{D}$ (uniform over its support) such that for every $i \in [n]$ we have $|N_{\mathcal{D}}(i)| \leq g$ for some $g > 0$ (g may depend on n). Then letting $C' = \text{Amplify(C)} \subseteq \mathbb{F}^{n'}$ and $\mathcal{D}'$ be its tester (uniform over its support), where $n \leq n' \leq (g \cdot h) \cdot n$ we have*

- *C' is a code (linear over $\mathbb{F}_2$) and $\text{supp}(\mathcal{D}') = O(g \cdot n')$,*
- *For all $i \in [n']$ we have $|N_{\mathcal{D}'}(i)| \leq h$,*
- *All core entries are preserved: $C'|_{[n]} = C$ and all new entries are computed from the original entries: $\dim(C') = \dim(C)$,*
- *Every entry indexed by $[n]$ affects at most $h \cdot g$ entries indexed by $[n'] \setminus [n]$ in the code C',*

- *All new entries are added as a sequence of blocks (small Hadamard codes) such that all these blocks are of the same constant size and each block is a $(3, \frac{1}{2})$-strong LTC,*
- *For every $w \in \mathbb{F}^{n'}$: if $\mathbf{Pr}_{I\sim\mathcal{D}}\left[(w|_{[n]})|_I \notin C|_I\right] \geq \epsilon$, then we have $\mathbf{Pr}_{I\sim\mathcal{D}'}[w|_I \notin C'|_I] \geq \min\{10\cdot\epsilon, \gamma\}$.*
  (Note: a single iteration of gap amplification procedure can improve the rejection probability by any multiplicative constant.)

Now we conclude the following theorem that summarizes the affect of Amplify$(\cdot)$ on the relaxed LTCs.

Theorem 5.2 (Gap Amplification for relaxed LTCs)**.** *Let $\mathbb{F} = \mathbb{F}_{2^3}$. There exist fixed constants $\gamma, d > 0$ such that the following holds. Let $C \subseteq \mathbb{F}^n$ be a $(2, \epsilon_1, \epsilon_2)$-rLTC (linear over $\mathbb{F}_2$), where $\epsilon_2 \leq \epsilon_1$, with regards to its core $A(C)$ and its tester $\mathcal{D}$ (uniform over its support), such that for every $i \in [n]$ we have $|N_{\mathcal{D}}(i)| \leq g$ for some $g > 0$ (g may depend on n). Then letting $C' =$ Amplify(C) and $A(C') = A(C)$ we have*

- *$C' \subseteq \mathbb{F}^{n'}$ is a $(2, \min\{2\epsilon_1, \gamma\}, \frac{\epsilon_2}{3d\cdot g})$-rLTC with regards to its core $A(C')$ and its new tester $\mathcal{D}''$ (uniform over its support),*
- *for all $i \in [n']$ we have $|N_{\mathcal{D}''}(i)| \leq d$,*
- *$C'|_{[n]} = C$ and thus $\delta(C'|_{A(C')}) = \delta(C|_{A(C)})$, and*
- $\mathrm{rate}(C') = \Omega(g \cdot \mathrm{rate}(C))$.

Proof. Theorem 5.1 says that $C'|_{[n]} = C$. The fact that $A(C') = A(C)$ implies that $\delta(C'|_{A(C')}) = \delta(C|_{A(C)})$. Theorem 5.1 also claims that $\dim(C') = \dim(C)$ and $n \leq n' \leq (g\cdot h)\cdot n$ for some constant $h \in \mathbb{N}^+$. Hence $\mathrm{rate}(C') = \Omega(g \cdot \mathrm{rate}(C))$.

Let $w' \in \mathbb{F}^{n'}$ and $w = w'|_{[n]} \in \mathbb{F}^n$. We know that there exists $c \in C$ such that letting $\hat{\epsilon} = \max\left\{\epsilon_1 \cdot \delta(w|_{A(C)}, c|_{A(C)}), \epsilon_2 \cdot \delta(w|_{-A(C)}, c|_{-A(C)})\right\}$ we have

$$\mathbf{Pr}_{I\sim\mathcal{D}}[w|_I \notin C|_I] \geq \hat{\epsilon}.$$

Let $c' =$ Amplify(c), i.e., the codeword $c' \in C'$ is produced from c by Amplify$(\cdot)$. Note that $c'|_{[n]} = c$. Theorem 5.1 implies that

$$\mathbf{Pr}_{I\sim\mathcal{D}'}\left[w'|_I \notin C'|_I\right] \geq \min\{\gamma, 10\cdot\hat{\epsilon}\} \geq \min\left\{\gamma, 10\epsilon_1 \cdot \delta(w'|_{A(C')}, c|_{A(C')})\right\},$$

where $\mathcal{D}'$ is a tester for C' guaranteed by Theorem 5.1.

Note that by Theorem 5.1, all new coordinates that are added form a sequence of the Hadamard blocks of the equal constant size, and each original entry (indexed by $[n]$) affects at most $d\cdot g$ new entries (indexed by $[n'] \setminus [n]$). Thus we can define a new tester $\mathcal{D}''$ for C' that invokes the original tester $\mathcal{D}'$ with probability $\frac{1}{2}$, and otherwise pick a random Hadamard block and test it. One can verify that for all $i \in [n']$ we have $|N_{\mathcal{D}''}(i)| \leq d$, and that $\mathcal{D}''$ is uniform over its support (by considering its support as a multiset).

We claim that

$$\mathbf{Pr}_{I\sim\mathcal{D}''}\left[w'|_I \notin C'|_I\right] \geq \frac{1}{2}\cdot\mathbf{Pr}_{I\sim\mathcal{D}'}\left[w'|_I \notin C'|_I\right] \geq \min\left\{\frac{1}{2}\cdot\gamma, 2\epsilon_1 \cdot \delta(w'|_{A(C')}, c'|_{A(C')})\right\}$$

and

$$\Pr_{I\sim\mathcal{D}''}\left[w'|_I \notin C'|_I\right] \geq \frac{\epsilon_2}{3d\cdot g}\cdot\delta(w'|_{[n']\setminus A(C')}, c'|_{[n']\setminus A(C')}).$$

The last inequality holds due to the guarantee of the tester $\mathcal{D}'$, the fact that each original entry affects at most $d\cdot g$ new entries and due to the fact that the Hadamard code is a $(3,\frac{1}{2})$-strong LTC (see Section A). Thus C' is a $(2,\min\{2\epsilon_1,\gamma'\},\frac{\epsilon_2}{3d\cdot g})$-rLTC, where $\gamma'=\gamma/2$, with regards to its core $A(C')$ and its tester $\mathcal{D}''$. □

We are ready to prove Theorem 1.4.

Proof of Theorem 1.4. Let $d\in\mathbb{N}^+$ and $C_1\subseteq(\mathbb{F}_2^3)^n$ be a code from Corollary 3.4. Now we execute the following algorithm.

- For each $i=1,\ldots,d\cdot\log\log n$ do
  - $C_{i+1}:=\text{Amplify}(\text{C}_\text{i})$
  - $A(C_{i+1}):=A(C_i):=[n]$

Theorem 5.2 implies that if C_i is a $(2,\frac{2^i}{\log^d n},\alpha)$-rLTC then C_{i+1} is a $(3,\frac{2^{i+1}}{\log^d n},\frac{\alpha}{b})$-rLTC for a fixed constant $b>0$ (besides the first iteration, where b might be $\text{polylog}(n)$). Moreover, $\delta(C_{i+1}|_{A(C_{i+1})})=\delta(C_i|_{A(C_i)})=\delta(C)$, $\dim(C_{i+1})=\dim(C_i)$ and $\text{rate}(C_{i+1})\geq\beta\cdot\text{rate}(C_i)$ for some constant β (besides the first iteration, where $\text{rate}(C_2)\geq\frac{1}{\text{polylog}(n)}\cdot\text{rate}(C_1)$).

Let $C'=C_{d\cdot\log\log n}$ and $A(C')=A(C_{d\cdot\log\log n})$. Then C' is a $(2,\gamma,\frac{1}{\text{polylog}(n)})$-rLTC, $\text{rate}(C')=\frac{1}{\text{polylog}(n)}$ and $\delta(C'|_{A(C')})\geq\Omega(1)$.

We claim that the resulting code C' can be a binary linear $(3,\gamma)$-rLTC. Observe that the alphabet reduction stage in the gap amplification procedure is done by the encoding every node's assignment of the underlying graph by the binary Hadamard code which is a binary linear 3-query strong LTC. Thus, as was explained in Section 4, if we don't convert this C' to the 2-query LTC over $\mathbb{F}_2^3$, then it will stay binary linear $(3,\gamma)$-rLTC such that $\text{rate}(C')=\frac{1}{\text{polylog}(n)}$ and $\delta(C'|_{A(C')})\geq\Omega(1)$.

Corollary 2.5 implies the required construction of binary linear $(3,\gamma/6)$-strong LTCs from C'. □

6 Proof of Theorem 1.8

First, we state Lemma 6.1. Then we prove Theorem 1.8.

Lemma 6.1. *Let $C\subseteq\mathbb{F}_2^n$ be a linear (q,ϵ)-strong LTC such that $\delta(C)=\Omega(1)$. Assume that $T\subseteq[n]$ is (n',h)-junta with respect to C such that $\dim(C|_T)=\dim(C)$ and $\delta(C|_T)=\Omega(1)$.*

Then $C|_T\subseteq\mathbb{F}_2^{n'}$ is a linear $\left(q\cdot h\cdot\left\lceil\frac{n}{\epsilon\cdot|T|}\right\rceil,\frac{1}{2}\right)$-strong LTC, $\delta(C|_T)=\Omega(1)$ and $\text{rate}(C|_T)=\frac{\dim(C)}{n'}$.

The proof of Lemma 6.1 appears in Section 6.1. We are ready to prove Theorem 1.8.

Proof of Theorem 1.8. Conjecture 1.6 yields a linear $(\text{polylog}(n),\frac{1}{2})$-strong LTC $C\subseteq\mathbb{F}_2^n$ (for arbitrary large $n\in\mathbb{N}^+$) such that $\delta(C)=\Omega(1)$, $\text{rate}(C)\geq\frac{1}{\text{polylog}(n)}$ and a $(\Theta(\dim(C)),\text{polylog}(n))$-junta T with respect to C.

If $\text{rate}(C) > \frac{\delta(C)}{100} = \Omega(1)$, then $C' = C$ is the required code whose blocklength is $n' = n$, and we are done. Otherwise, assume that $\text{rate}(C) \leq \frac{\delta(C)}{100}$.

Proposition B.1 shows the existence of a subset $S \subseteq [n]$, $|S| \leq \frac{30\dim(C)}{\delta(C)}$ such that $\delta(C|_S) \geq \delta(C)/2$ and $\dim(C|_S) = \dim(C)$. It holds that $T' = T \cup S$ is a $(\Theta(\dim(C)), \text{polylog}(n))$-junta such that $\dim(C|_{T'}) = \dim(C)$ and $\delta(C|_{T'}) = \Omega(1)$. Let $n' = |T'|$ and $C' = C|_{T'}$ and note that $\frac{n}{n'} \leq \text{polylog}(n')$. By Lemma 6.1, we conclude that $C' \subseteq \mathbb{F}_2^{n'}$ is a linear $(\text{polylog}(n'), \frac{1}{2})$-strong LTC such that $\delta(C') = \Omega(1)$ and $\text{rate}(C') = \Omega(1)$. Finally, note that n' can be arbitrary large since $n' = |T'| \geq |S| \geq \dim(C)$ and $\dim(C) \geq \frac{n}{\text{polylog}(n)}$, where n can be arbitrary large. □

6.1 Proof of Lemma 6.1

We first state and prove the following folklore claim.

Claim 6.2 (Folklore)**.** *If $C \subseteq \mathbb{F}^n$ is a (q, ϵ)-strong LTC, then it is also a $\left(q \cdot \left\lceil \frac{1}{\epsilon} \right\rceil, \frac{1}{2}\right)$-strong LTC.*

Proof. Let $\mathcal{D}$ be a (q, ϵ)-strong tester for C. Let $\mathcal{D}'$ be a tester for C defined as follows: $\mathcal{D}'$ samples $\mathcal{D}$ on the input word $\left\lceil \frac{1}{\epsilon} \right\rceil$ times and rejects if and only if at least one invocation of $\mathcal{D}$ rejected. Clearly, $\mathcal{D}'$ always accepts the codewords and the query complexity of $\mathcal{D}'$ is upper-bounded by $q \cdot \left\lceil \frac{1}{\epsilon} \right\rceil$.

Let $w \in \mathbb{F}^n$. We know that $\underset{I \sim \mathcal{D}}{\mathbf{Pr}}[w|_I \sim C|_I] \geq \epsilon \cdot \delta(w, C)$. The tester D' accepts w with probability at most $(1 - \epsilon \cdot \delta(w, C))^{1/\epsilon}$ and rejects w with probability at least $1 - (1 - \epsilon \cdot \delta(w, C))^{1/\epsilon}$. We argue that $1 - (1 - \epsilon \cdot \delta(w, C))^{1/\epsilon} > \frac{1}{2} \cdot \delta(w, C)$ and this yields the Claim.

It holds that $\left((1 - \epsilon \cdot \delta(w, C))^{1/\epsilon}\right)^{\frac{1}{\delta(w,C)}} = (1 - \epsilon \cdot \delta(w, C))^{\frac{1/\epsilon}{\delta(w,C)}} \leq e^{-1} < \frac{1}{2}$ and $1 - (1 - \epsilon \cdot \delta(w, C))^{\frac{1}{\epsilon \cdot \delta(w,C)}} > \frac{1}{2}$. On the other hand,

$$1 - (1 - \epsilon \cdot \delta(w, C))^{\frac{1/\epsilon}{\delta(w,C)}} \leq \frac{1}{\delta(w, C)} \cdot \left(1 - (1 - \epsilon \cdot \delta(w, C))^{(1/\epsilon)}\right),$$

where we used the fact that $1 - p^l \leq l \cdot (1 - p)$ for $p \leq 1$ and $l \in \mathbb{N}^+$.[8] The required inequality is obtained by replacing p with $(1 - \epsilon\delta(w, C))^{1/\epsilon}$ and l with $\frac{1}{\delta(w,C)}$ (we assume without loss of generality that $\frac{1}{\delta(w,C)}$ is an integer because otherwise we could use $\left\lceil \frac{1}{\delta(w,C)} \right\rceil$).

We conclude that $\frac{1}{2} < \frac{1}{\delta(w, C)} \cdot \left(1 - (1 - \epsilon \cdot \delta(w, C))^{1/\epsilon}\right)$ and $\frac{\delta(w, C)}{2} < 1 - (1 - \epsilon \cdot \delta(w, C))^{1/\epsilon}$. □

Now we prove Lemma 6.1.

Proof of Lemma 6.1. By definition, $C|_T \subseteq \mathbb{F}_2^{|T|}$ is a linear code such that $\delta(C|_T) = \Omega(1)$ and $\text{rate}(C|_T) = \frac{\dim(C|_T)}{|T|} = \frac{\dim(C)}{n'}$. We prove that $C|_T$ is a $(q \cdot h, \epsilon \cdot \frac{|T|}{n})$-strong LTC. By definition, for every $j \in [n] \setminus T$ there exists $u_j \in C^{\perp}_{\leq h}$ such that $j \in \text{supp}(u_j)$ and $\text{supp}(u_j) \setminus \{j\} \subseteq T$. Fix these vectors u_j.

Let $\mathcal{D}$ be a tester for C and let us define a tester $\mathcal{D}_T$ for the code $C|_T$.

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

- Input: $w \in \mathbb{F}_2^{n'}$
- Sample $\mathcal{D}$ and let $\left\{j_q, j_2, \ldots, j_{q'}\right\}$ be the output of $\mathcal{D}$ ($q' \leq q$)
- For every j_m, if $j_m \in T$ then let $J_m = \{j_m\}$, and otherwise let $J_m = (\mathrm{supp}(u_{j_m}) \setminus \{j_m\})$.
- Output $J_1 \cup J_2 \cup \ldots \cup J_{q'}$

Clearly, the query complexity of $\mathcal{D}_T$ is at most $q \cdot h$. We need to prove that $\mathbf{Pr}_{I \sim \mathcal{D}_T}[w|_I \notin (C|_T)|_I] \geq \epsilon \cdot \frac{|T|}{n} \cdot \delta(w, C|_T)$. Let $w' \in \mathbb{F}_2^n$ be reconstructed from w, i.e., $w'|_T = w$ and for every $j \in [n] \setminus T$ we have that $w'|_j = \sum_{i \in \mathrm{supp}(u_j) \setminus \{j\}} w'|_i$. We know that

$$\delta(w, C|_T) = \frac{\Delta(w, C|_T)}{|T|} \leq \frac{\Delta(w', C)}{|T|} = \frac{\Delta(w', C)}{n} \cdot \frac{n}{|T|} = \delta(w', C) \cdot \frac{n}{|T|}.$$

We have $\mathbf{Pr}_{I \sim \mathcal{D}_T}[w|_I \notin (C|_T)|_I] \geq \mathbf{Pr}_{I \sim \mathcal{D}}[w'|_I \notin C|_I] \geq \epsilon \cdot \delta(w', C) \geq \epsilon \cdot \frac{|T|}{n} \cdot \delta(w, C)$.

Thus $C|_T$ is a $(q \cdot h, \epsilon \cdot \frac{|T|}{n})$-strong LTC with respect to the tester $\mathcal{D}_T$. Claim 6.2 proves that $C|_T$ is a $(q', \frac{1}{2})$-strong LTC, where $q' = q \cdot h \cdot \left\lceil \frac{1}{\epsilon \cdot \frac{|T|}{n}} \right\rceil$. □

7 Open Questions and Discussions

This work leaves two open questions. The first one is obtaining asymptotically good strong LTCs with poly-log query complexity and constant soundness. In Theorem 1.8 we argued their existence under Conjecture 1.6. One can try to prove this conjecture. Our feel is that it might be possible to implement the random projection operation [50, 55] to preserve the "junta" property (see Remark 1.7). E.g., it might be possible to argue that there exists some invariant feature that is preserved each iteration in the construction of [55] and use this feature to re-implement the random projection operation. Resolving this task would yield an unconditional proof for Theorem 1.8.

The second open question, mentioned in [36, 50], is the *explicit* construction of strong LTCs with inverse polylogarithmic rate, constant relative distance, constant query complexity and constant soundness. Recall that our construction of strong LTCs in Theorem 1.4 is based on the construction of [55] (which is almost identical to [50]), where the construction of [55] was probabilistic. One of possible approach to provide an explicit construction of such strong LTCs is by applying the arguments of [55] and the arguments used in this paper to the construction of Ben-Sasson and Sudan [19]. While the work [19] yields weak LTCs, the underlying construction has some similarities to the constructions of [50, 55] discussed [50, Sectioin 7.2]. On the other hand, the ideas presented in [55] seem fairly general and it might that these ideas can be applied to [19] to conclude the explicit construction of strong LTCs with the required range of parameters.

Acknowledgements

The author thanks Eli Ben-Sasson for helpful discussions. We would like to thank Or Meir for raising the suggestion (discussed in Section 7) to get an explicit construction of strong LTCs by applying the arguments of [55] to the codes of [19].

References

[1] 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.

[2] 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.

[3] Sanjeev Arora and Shmuel Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.

[4] L. Babai, A. Shpilka, and D. Stefankovic. Locally testable cyclic codes. In *Proceedings: 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2003, 11–14 October 2003, Cambridge, Massachusetts*, pages 116–125. IEEE Computer Society Press, 2003.

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

[6] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 294–304, New York, 1993. ACM SIGACT, ACM Press.

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

[8] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free Bits, PCPs, and Nonapproximability—Towards Tight Results. *SIAM Journal on Computing*, 27(3):804–915, June 1998.

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

[10] 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.

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

[12] Eli Ben-Sasson, Elena Grigorescu, Ghid Maatouk, Amir Shpilka, and Madhu Sudan. On sums of locally testable affine invariant properties. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 6845 of *Lecture Notes in Computer Science*, pages 400–411. Springer, 2011.

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

[14] Eli Ben-Sasson, Prahladh Harsha, Oded Lachish, and Arie Matsliah. Sound 3-Query PCPPs Are Long. *TOCT*, 1(2), 2009.

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

[16] Eli Ben-Sasson, Ghid Maatouk, Amir Shpilka, and Madhu Sudan. Symmetric LDPC Codes are not Necessarily Locally Testable. In *IEEE Conference on Computational Complexity*, pages 55–65. IEEE Computer Society, 2011.

[17] Eli Ben-Sasson, Noga Ron-Zewi, and Madhu Sudan. Sparse affine-invariant linear codes are locally testable. In *FOCS*, pages 561–570. IEEE Computer Society, 2012.

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

[19] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM J. Comput*, 38(2):551–607, 2008.

[20] Eli Ben-Sasson and Madhu Sudan. Limits on the Rate of Locally Testable Affine-Invariant Codes. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 6845 of *Lecture Notes in Computer Science*, pages 412–423. Springer, 2011.

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

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

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

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

[25] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *FOCS*, pages 488–497. IEEE Computer Society, 2010.

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

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

[28] Irit Dinur and Tali Kaufman. Dense locally testable codes cannot have constant rate and distance. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 6845, pages 507–518, 2011.

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

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

[31] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[32] Oded Goldreich. Home page.

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

[34] Oded Goldreich and Or Meir. A Small Gap in the Gap Amplification of Assignment Testers. *Electronic Colloquium on Computational Complexity (ECCC) - TR05-046. Comment 3*, 2007.

[35] Oded Goldreich and Dana Ron. On proximity-oblivious testing. *SIAM J. Comput*, 40(2):534–566, 2011.

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

[37] Elena Grigorescu, Tali Kaufman, and Madhu Sudan. 2-Transitivity Is Insufficient for Local Testability. In *IEEE Conference on Computational Complexity*, pages 259–267. IEEE Computer Society, 2008.

[38] Elena Grigorescu, Tali Kaufman, and Madhu Sudan. Succinct representation of codes with applications to testing. *SIAM J. Discrete Math*, 26(4):1618–1634, 2012.

[39] Venkatesan Guruswami. On 2-Query Codeword Testing with Near-Perfect Completeness. In *Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC)*, volume 4288 of *Lecture Notes in Computer Science*, pages 267–276. Springer, 2006.

[40] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

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

[42] Tali Kaufman and Shachar Lovett. New Extension of the Weil Bound for Character Sums with Applications to Coding. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, (FOCS)*, pages 788–796. IEEE, 2011.

[43] Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM J. Comput*, 36(3):779–802, 2006.

[44] Tali Kaufman and Madhu Sudan. Sparse Random Linear Codes are Locally Decodable and Testable. In *FOCS*, pages 590–600. IEEE Computer Society, 2007.

[45] Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC), Victoria, British Columbia, Canada, May 17-20, 2008*, pages 403–412. ACM, 2008.

[46] Tali Kaufman and Avi Wigderson. Symmetric LDPC codes and local testing. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 406–421. Tsinghua University Press, 2010.

[47] Gillat Kol and Ran Raz. Bounds on 2-query locally testable codes with affine tests. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:138, 2009.

[48] Gillat Kol and Ran Raz. Locally testable codes analogues to the unique games conjecture do not exist. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:128, 2009.

[49] Swastik Kopparty and Shubhangi Saraf. Local list-decoding and testing of random linear codes from high error. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 417–426. ACM, 2010.

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

[51] Jaikumar Radhakrishnan. Gap amplification in PCPs using lazy random walks. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I*, volume 4051 of *Lecture Notes in Computer Science*, pages 96–107. Springer, 2006.

[52] Noga Ron-Zewi and Madhu Sudan. A new upper bound on the query complexity for testing generalized reed-muller codes. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 7408 of *Lecture Notes in Computer Science*, pages 639–650. Springer, 2012.

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

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

[55] Michael Viderman. Strong LTCs with inverse polylogarithmic rate and soundness. *To appear in CCC 2013. Electronic Colloquium on Computational Complexity (ECCC)*, 19:168, 2012.

A The Hadamard code

Let us first define locally correctable codes (LCCs).

Definition A.1 (LCCs)**.** Let $C \subseteq \mathbb{F}_2^n$ be a linear code of dimension $k = \dim(C)$. Then C is a (q, ϵ, α)-LDC (where $\epsilon < \frac{1}{|\mathbb{F}_2|} = \frac{1}{2}$) if there exists a local corrector **Cor** that reads at most q symbols from the input word and the following condition holds.

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

The following fact regarding the testability of the Hadamard code is known due to the works [26, 9, 7, 41], while the fact saying that the Hadamard code is a 2-query LCC is a folklore.

Theorem A.2. *Let $H \subseteq \mathbb{F}_2^{2^k-1}$ be the Hadamard code and note that $k = \dim(H)$. Then, H is a $(3, \frac{1}{2})$-strong LTC and a $(2, 2\alpha, \alpha)$-LCC for every $0 \leq \alpha \leq 1/4$.*

B Auxiliary probabilistic claims

The following proposition appeared in [50, Theorem 4.7]. We reproduce it for the sake of completeness.

Proposition B.1 ([50])**.** *Let $C \subseteq \mathbb{F}_2^n$ be a linear code such that $\mathrm{rate}(C) \leq \frac{\delta(C)}{100}$. Let $h = \frac{30 \cdot \dim(C)}{\delta(C)}$. Then,*

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

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

Proof. To prove that $C|_S$ has relative distance at least $\frac{\delta(C)}{2}$ and that $\dim(C|_S) = \dim(C)$, we use a standard probabilistic argument. Fix a non-zero codeword $c \in C$, and let $S \subseteq [n]$ be a uniformly chosen set such that $|S| \leq h$. The relative weight of c is at least $\delta(C)$, and therefore the expected relative weight of $c|_S$ is at least $\delta(C)$. Applying the Chernoff Bound (Claim B.2), it follows that the probability that the relative weight of $c|_S$ is less than $\frac{\delta(C)}{2}$ is at most $\exp\left(-\frac{\frac{1}{4} \cdot \delta(C) \cdot h}{3}\right)$.

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

$$2^{\dim(C)} \cdot \exp(-\frac{1}{12} \cdot \delta(C) \cdot h) \leq \exp(-\dim(C)).$$

□

For the sake of completeness we state the particular version of Chernoff's inequality that we use.

Claim B.2 (Chernoff Bound)**.** *If $X = \sum_{i=1}^m X_i$ is a sum of independent $\{0,1\}$-valued random variables, where $\mathbf{Pr}[X_i = 1] = \gamma$, then*

$$\mathbf{Pr}\left[\frac{X}{m} < (1-\sigma)\gamma\right] \leq \exp(-\frac{\sigma^2 \gamma m}{3}) \quad \text{and} \quad \mathbf{Pr}\left[\frac{X}{m} > (1+\sigma)\gamma\right] \leq \exp(-\frac{\sigma^2 \gamma m}{3}).$$

ECCC ISSN 1433-8092
http://eccc.hpi-web.de