Electronic Colloquium on Computational Complexity, Report No. 200 (2010)

Towards lower bounds on locally testable codes via density arguments

Eli Ben-Sasson*
Computer Science Department
Technion — Israel Institute of Technology
Haifa 32000, Israel
eli@cs.technion.ac.il

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

December 14, 2010

Abstract

The main open problem in the area of locally testable codes (LTCs) is whether there exists an asymptotically good family of LTCs and to resolve this question it suffices to consider the case of query complexity 3. We argue that to refute the existence of such an asymptotically good family one should prove that the number of dual codewords of weight at most 3 is super-linear in the blocklength of the code.

The main technical contribution of this paper is an improvement of the combinatorial lemma of Goldreich et al. [2006] which bounds the rate of 2-query locally decodable codes (LDCs) and is used in state-of-the-art rate-bounds for linear LDCs. The lemma of Goldreich et al. bounds the rate of 2-query LDCs of blocklength n in terms of the corruption parameter $\delta(n)$ — this is the maximal number of corrupted codeword bits for which a (2-query) decoder can recover correctly every message bit (with high probability). Our combinatorial lemma gives nontrivial rate bounds for any corruption parameter $\delta(n) = \omega(1)$, whereas the previous lemma works only for corruption parameter larger than $\log n$. The study of LDCs with sublinear corruption parameter is also motivated by Dvir's [2010] observation that sufficiently strong bounds on the rate of such LDCs imply explicit constructions of rigid matrices.

1 Introduction

This paper is motivated by one of the most important open problems regarding locally testable codes (LTCs), whether there exists an asymptotically good family of LTCs with constant query complexity. For an introduction to LTCs and explanation of their relation to property testing and probabilistically checkable proofs (PCPs) we refer the reader to the work of Goldreich and Sudan [20] which started the recent line of work on LTC-rate. For a recent survey of known results about rate-bounds of LTCs see [5]. To avoid repeating what is recounted in these works, suffice it to say

*The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 240258. Research of both authors supported by grant number 2006104 by the US-Israel Binational Science Foundation and by grant number 679/06 by the Israeli Science Foundation.

†Part of the work was done while the author was a summer intern at Microsoft Research New England.

ISSN 1433-8092

that for all the work that has gone into the study of LTCs, our understanding of their rate is very limited. The only negative results on LTCs rate concern special families of codes testable with just 2-queries [7, 21, 29, 28], random low density parity check (LDPC) codes [9], cyclic codes [4], solvable codes [26] and affine-invariant codes [11]. In fact, we cannot even rule out the existence of binary LTCs meeting the Gilbert-Varshamov bound (which is the best known rate for codes without any local testing restriction). So, for all we know, the strong testability requirement of LTCs may not "cost" anything extra over LDPC codes!

We suggest a strategy to disprove the existence of an asymptotically good family of linear LTCs. Without loss of generality we may deal with the case of query complexity 3 (cf. Theorem A.1). Our proof-strategy goes by way of contradiction and relies on proving the following pair of conjectures.

- If $\mathcal{C}$ is an *asymptotically good* 3-query LTC then C has a super-linear number of dual codewords of weight at most 3.
- If $\mathcal{C}$ is an asymptotically good 3-query LTC and has a super-linear number of dual codewords of weight at most 3 then $\text{rate}(\mathcal{C}) = o(1)$.

The result of Ben-Sasson et al. [8] seems to lead in the direction of proving the first item as it shows that all LTCs have more small-weight dual codewords than what is needed to characterize the code and the small-weight dual codewords display nontrivial dependencies among them. In this paper we make initial progress on the second item and show that a broad family of 3-query LTCs (including all "base constructions" of LTCs) cannot have both constant rate and a super-linear number of dual codewords of weight at most 3.

Roughly speaking, LTCs are invariably constructed by starting with a decent "base-construction" of an LTC (such as a Hadamard, Reed-Muller, or constant-blocklength code) and modifying it by various techniques like repetition [32], concatenation [2, 3], tensoring [10], gap-amplification [12], taking direct-products [13, 22] and PCPP-composition [6, 14]. These operations improve the LTC-related parameters of the code, they increase soundness and/or reduce query complexity but none of them increases rate. In fact, the improvement in LTC-related parameters of the afore-mentioned operations comes at the price of reduced rate. So if asymptotically good LTCs are to be constructed one should start with a "base-construction" that is asymptotically good, or come up with a new set of LTC-related techniques that do not decrease code-rate.

Looking into known "base-constructions" of q-query LTCs they all share a few properties that we formalize in this paper. First, they are *q-regular*, i.e., every codeword-bit sees the same number of dual codewords of weight $q' \leq q$ (see Definition 3.5). Second, they are all *q-dense*, by which we mean that the number of dual codewords of weight at most q is super-linear in the code blocklength. Indeed, a popular belief [34] (stated formally in Conjecture 3.3) says that all q-query LTCs are q-dense (see Definition 3.1).

Our main result is that families of 3-dense and 3-regular LTCs cannot be asymptotically good. We bound the rate of the code as a function of 3-density and show that even arbitrarily slowly growing 3-density implies vanishing rate (cf. Theorem 3.6 and Corollary 3.7). We then put forth a conjecture stating that all LTCs contain a punctured code that is roughly regular (Conjecture 3.11) and show that under this conjecture there are no asymptotically good families of LTCs whatsoever (cf. Theorem 3.9 and Corollary 3.12). We go on to say that regular codes strongly generalize symmetric[1] LTCs, i.e., LTCs which are invariant under a group of permutations that is 1-transitive.

[1] These codes are called symmetric since every coordinate of the code participates in a similar set of dual codewords.

A subclass of these codes — so-called "2-transitive" codes — was suggested by Alon et al. [1] as possibly being locally testable, and this family was first studied systematically for the special case of affine-invariant codes by Kaufman and Sudan [24]. As a corollary of our main results we show that 3-query symmetric LTCs with super-constant density are not asymptotically good.

Improved rate bounds for weak 2-query LDCs Our analysis of 3-query LTCs relies on a new upper bound on the rate of families of locally decodable codes (LDCs) with a rather weak requirement on their decoding capabilities described next. This LTC-to-LDC reduction is especially interesting in light of the fact that LTCs and LDCs seem to be two very different kinds of codes (cf. Kaufman and Viderman [25]).

Recall that q-query LDCs allow to recover each message entry with constant probability by reading only q entries of the codeword even if "large" number of codeword bits are adversely corrupted. Best known upper bounds on the rate of linear q-query LDCs [35, 36] (with the notable exception of the bounds for nonlinear 2-query LDCs of Kerenidis and de Wolf [27]) go by reducing the problem to that of showing rate bounds for 2-query LDCs. And the best rate bounds for 2-query LDCs follow from the so-called "combinatorial lemma" of Goldreich et al. [19, Lemma 3.3] (see also [16]). Our main technical contribution is an improvement of this combinatorial lemma as described next. The combinatorial lemma of [19] bounds the rate of a 2-query LDC in terms of the corruption parameter — the number of bits which can be adversarially corrupted. All things considered, as the corruption parameter decreases, it should get easier to construct LDCs (because the adversary is more restricted) and consequently is should get harder to prove upper bounds on the rate of such LDCs. Indeed, the rate bound given by the combinatorial lemma becomes trivial when the number of corrupted bits is roughly logarithmic in the blocklength of the code. Our improved combinatorial lemma (Lemma 3.15) gives nontrivial rate bounds for any super-constant corruption parameter (see Section 3.2).

Two additional remarks regarding our combinatorial lemma should be made. First, given that state-of-the-art bounds on rate of q-query LDCs for $q \geq 3$ rely on rate bounds 2-query LDCs with a sublinear number of errors [35, 36] shows that proving rate bounds for smaller values should result in improved bounds for q-query LDCs even for larger values of q. Second, the recent work of Dvir [15] shows that proving sufficiently strong lower bounds on locally decodable codes which can be corrected from a sublinear number of corruptions would result in explicit constructions of rigid matrices, giving further motivation for our lemma.

We end with a few words on our proof of the rate-bound on 2-query LDCs (Lemma 3.15) and how it differs from the proof method of Goldreich et. al in [19]. They provided two different proofs, the first uses an isoperimetric inequality statement regarding the hypercube and the second is an information-theoretic argument due to Alex Samorodnitsky. Our proof goes by removing carefully selected columns from the generating matrix of a locally decodable code. This removal, we argue, partitions the rows of the matrix into sets of identical rows. We study how the sets identical rows grow in size with the removal of additional columns and perform a careful amortized analysis of this process (see Section 5 for details).

Related work In the course of writing our result we have learned that Irit Dinur and Tali Kaufman have independently studied the effect of 3-density on rate of locally testable codes and have obtained related results through seemingly different methods.

Organization of the paper. In the following section we provide background regarding locally testable and locally decodable codes. In Section 3 we state our main results (Theorems 3.6, 3.9, 3.17). We prove our main results on LTCs in Section 4. We go on to prove the main technical Lemma 3.15 in Section 5. Finally, in Section 6 we prove our improved bound for 2-query LDCs over arbitrary fields (Theorem 3.17).

2 Preliminaries

We start with a few definitions. Let $\mathbb{F}$ be a finite field and $[n]$ be the set $\{1,\ldots,n\}$. Let $C \subseteq \mathbb{F}^n$ be a linear code over $\mathbb{F}$. The dimension of C is denoted by $\dim(C)$ and its rate is denoted by $\mathrm{rate}(C)$ and defined to be $\mathrm{rate}(C) = \dim(C)/n$. For $w \in \mathbb{F}^n$, let $\mathrm{supp}(w) = \{i \in [n] \mid w_i \neq 0\}$ and $|w| = |\mathrm{supp}(w)|$. We define the *relative distance* between two words $x, y \in \mathbb{F}^n$ to be $\delta(x,y) = \frac{|x-y|}{n}$. The relative distance of a code is denoted by $\delta(C)$ and defined to be $\delta(C) = \min_{x \neq y \in C} \delta(x,y)$. For $x \in \mathbb{F}^n$ and $C \subseteq \mathbb{F}^n$, let $\delta(x,C) = \min_{y \in C}\{\delta(x,y)\}$ denote the relative distance of x from the code C. The vector inner product between u_1 and u_2 is denoted by $\langle u_1, u_2\rangle$. The dual code $C^\perp$ is defined as $C^\perp = \{u \in \mathbb{F}^n \mid \forall c \in C : \langle u, c\rangle = 0\}$. In a similar way we define $C^\perp_{\leq t} = \left\{u \in C^\perp \mid |u| \leq t\right\}$ and $C^\perp_t = \left\{u \in C^\perp \mid |u| = t\right\}$. For $w \in F^n$ and $S = \{j_1, j_2, \ldots, j_m\} \subseteq [n]$, where $j_1 < j_2 < \ldots < j_m$, let $w|_S = (w_{j_1}, w_{j_2}, \ldots, w_{j_m})$ be the *restriction* of w to the subset S. For $V \subseteq \mathbb{F}^n$ let $V|_S = \{v|_S \mid v \in V\}$ denote the restriction of the subspace V to the subset S. For any integer $n \geq 2$ let $\binom{n}{2} = [n] \times [n] \setminus \{(i,i) \mid i \in [n]\}$.

2.1 LTCs and LDCs

In this section, we define LTCs and LDCs formally and recall a few concepts that will be used later in this paper. We define LTCs following [8].

Definition 2.1 (LTCs)**.** Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code. We say that $\mathcal{C}$ is a (q,ϵ,δ)-LTC if there exists a distribution $\mathcal{D}$ over $\mathcal{C}^\perp_{\leq q}$ such that the following condition holds. For all $x \in \mathbb{F}^n$ such that $\delta(x,\mathcal{C}) \geq \delta$ it holds that $\Pr_{u \sim \mathcal{D}}[\langle u, x\rangle \neq 0] \geq \epsilon$.

The parameter q is known as query complexity, ϵ is the rejection probability and δ is the distance threshold.

Note that if $\mathcal{C}$ is a (q,ϵ,δ)-LTC then $\mathcal{C}$ is also a (q,ϵ,δ')-LTC for all $\delta' \geq \delta$. We say that a family of codes $\{\mathcal{C}_n\}_{n\in\mathbb{Z}}$ over the field $\mathbb{F}$ is locally testable if there exist constants $q,\epsilon,\delta > 0$ such that for infinitely many n it holds that $\mathcal{C}_n \subseteq \mathbb{F}^n$ is a $(q,\epsilon,\delta/3)$-LTC, where $\delta(\mathcal{C}_n) \geq \delta$.

Remark 2.2. Note that every perfect code $\mathcal{C}$ is $(0,1,\delta(\mathcal{C})/2)$-LTC, i.e., the code is testable with 0 queries since there are no words which are $(\delta(\mathcal{C})/2)$-far from the code. Hence, to avoid trivial cases we must require the distance threshold parameter to be strictly less than $\delta(\mathcal{C})/2$. Moreover, in the area of LTCs we usually require $\delta(\mathcal{C})/3$. E.g., all known constructions of LTCs satisfy this requirement (see e.g., [10, 12, 20, 23, 24, 30]). On the other side, if for all constants $q,\epsilon > 0$ the code $\mathcal{C}$ is not $(q,\epsilon,\delta(\mathcal{C})/3)$-LTC we say that $\mathcal{C}$ is not locally testable (see e.g., [4, 8, 25]).

Now we define locally decodable codes (LDCs).

Definition 2.3 (LDCs)**.** Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code and let $k = \dim(\mathcal{C})$. Let $E_{\mathcal{C}}$ be the encoding function, i.e., $\mathcal{C} = \left\{E_{\mathcal{C}}(x) \mid x \in \mathbb{F}^k\right\}$. Then $\mathcal{C}$ is a (q, ϵ, δ)-LDC, where $q, \epsilon, \delta > 0$, if there exists a randomized decoder ($\mathbb{D}$) that reads at most q entries and the following condition holds:

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

The parameter q is known as query complexity, ϵ is the recovery probability and δ is the corruption parameter.

We say that a family of codes $\{C_n\}_{n\in\mathbb{Z}}$ over the field $\mathbb{F}$ is a (q, ϵ, δ)-locally decodable if for infinitely many n it holds that $C_n \subseteq \mathbb{F}^n$ is a (q, ϵ, δ)-LDC.

3 Main Results

Our main motivation is the study of rate limitations of families of LTCs and the results regarding this question are presented in Section 3.1. The main tool used in our proofs is a new bound on the rate of weak 2-query LDCs. We present this bound and discuss its implications to LDCs in Section 3.2. We start by stating the popular belief about density of locally testable codes and for this we need first to define the notion of "dense" codes.

The results presented in this section deal with linear codes over the binary field. These results can be extended to any finite field but for simplicity we prefer to state them for the binary case.

Definition 3.1 (q-density)**.** Let $C \subseteq \mathbb{F}_2^n$ be a linear code and $q > 0$. Let $\Delta_q(C) = \left|C^{\perp}_{\leq q}\right|$ be the number of dual codewords of weight at most q and $\Delta_{q,i}(C) = \left|\left\{u \in C^{\perp}_{\leq q} \mid i \in \mathrm{supp}(u)\right\}\right|$ be the number of small-weight dual codewords that "touch" the index i. The *q-density* of C is defined as $\sigma_q(C) = \frac{\Delta_q(C)}{n}$.

Remark 3.2. The repetition code $C = \{0^n, 1^n\}$ is a 3-query LTC but $|C_3^{\perp}| = 0$. This example shows that the above definition of density which counts all words of weight at most q should not be replaced with the finer definition which counts all words of weight exactly q.

Popular belief [34] says that q-query LTCs have a superlinear number of dual codewords of weight at most q (e.g. see [8, Abstract]). Recall that to rule out the existence of asymptotically good LTCs it is sufficient to rule out 3-query asymptotically good LTCs (cf. Theorem A.1). The main point of this paper is to show that if the following conjecture is proven to be true then there are no asymptotically good natural families of LTCs.

Conjecture 3.3 (LTCs are dense)**.** *Let $\epsilon, \delta > 0$ be constants. Then there exists a function $\sigma : \mathcal{N} \to \mathcal{N}$ s.t. $\sigma(n) = \omega_{\epsilon,\delta}(1)$ such that the following condition holds.*

$$\textit{If } C \subseteq \mathbb{F}_2^n \textit{ is a } (3, \epsilon, \delta/3)\textit{-LTC and } \delta(C) \geq \delta \textit{ then } \sigma_3(C) \geq \sigma(n). \tag{1}$$

Remark 3.4. To rule out the existence of asymptotically good families of LTCs it is sufficient to make the weaker assumption that the family of codes in the conjecture above is asymptotically good and then prove (1) for such families. Indeed, all our results regarding asymptotically good codes work under this weaker assumption. The recent work of Ben-Sasson et al. [8] may be useful in this context as they showed that LTCs have many linear dependencies in their small weight dual codewords and this number increases with the rate of the code.

3.1 Dense natural and regular LTCs cannot be asymptotically good

To state our main results about LTCs we formalize the notion of q-regular, and natural, codes. (Recall that we have argued in the introduction that all base LTCs are natural, and even regular.) We note that q-regular codes are similar to *regular LDPC* codes introduced by Gallager [17, 18]. The main difference is that regular LDPC codes are defined by the regular structure of the parity check matrix, while our q-regular codes assume a regular structure in the subspace of all dual codewords of weight at most q. Later on we shall argue that the class of regular codes strictly contains the class of symmetric codes, suggested as candidate LTCs in [1] and first studied systematically in [24].

The notion of a natural code should be viewed as a weaker definition of regularity. It does not require that all codeword coordinates participate in the exact same number of small-weight dual words. Rather, it suffices that an independent set of indices (a notion we define next) each participate in a large number of dual words of small weight. We say that $I \subseteq [n]$ is a set of *independent indices* of a code $C \subseteq \mathbb{F}^n$ if $C|_I = \mathbb{F}^I$, or equivalently, there is no $u \in C^\perp$ s.t. $\mathrm{supp}(u) \subseteq I$. It can be easily verified that C has at least one set of independent indices of size $\dim(C)$. So, in particular, all regular codes are natural according to the following definition but the converse is not true.

Definition 3.5 (Regular and natural codes)**.** We say that a code $C \subseteq \mathbb{F}_2^n$ is *q-regular* if for all $q' \leq q$ and $i, j \in [n]$ we have

$$\left|\left\{u \in C_{q'}^\perp \mid i \in \mathrm{supp}(u)\right\}\right| = \left|\left\{u \in C_{q'}^\perp \mid j \in \mathrm{supp}(u)\right\}\right|.$$

We say that C is (α, Δ)-*natural* if there exists a set of independent indices $I \subseteq [n]$ s.t. $|I| \geq \alpha \cdot \dim(C)$ and for every $i \in I$ it holds that $\Delta_{3,i}(C) \geq \Delta$.

Our first main result demonstrates a tight relation between the density and the rate of 3-regular codes.

Theorem 3.6 (3-density limits rate of regular codes)**.** *Let $C \subseteq F_2^n$ be a 3-regular code s.t. $\sigma_3(C) \geq 2$. Then*

$$\mathrm{rate}(C) \leq \frac{2\log(\sigma_3(C)) + 2}{\sqrt{\sigma_3(C)}}.$$

Spielman [33] suggested to use dense regular expander codes for constructing LTCs. The next corollary says that dense 3-regular codes cannot be asymptotically good even without any expansion assumptions. Furthermore, this corollary limits the rate of 3-regular LTCs under Conjecture 3.3.

Corollary 3.7 (No asymptotically good regular 3-query LTCs)**.** *Let $C = \{C_n\}_{n\in\mathbb{Z}}$ is a family of 3-regular codes, where $C_n \subseteq \mathbb{F}_2^n$.*

- *If $\sigma_3(C_n) = \omega(1)$ then*
$$\mathrm{rate}(C_n) \leq \frac{2\log(\sigma_3(C_n)) + 2}{\sqrt{\sigma_3(C_n)}} = o(1).$$
- *Let $\epsilon, \delta > 0$ be constants. Under Conjecture 3.3, if $C_n \subseteq \mathbb{F}_2^n$ is a $(3, \epsilon, \delta/3)$-LTC and $\delta(C_n) \geq \delta$ then $\mathrm{rate}(C_n) = o(1)$.*

Proof. The first bullet follows from Theorem 3.6. For the second bullet, assume the contra-positive, i.e., $\mathrm{rate}(C_n) \geq \rho$ for some constant $\rho > 0$. Conjecture 3.3 says that $\sigma_3(C_n) = \omega(1)$. Theorem 3.6 then implies that $\mathrm{rate}(C_n) = o(1)$. □

Natural codes Next we present limits on the rate of natural LTCs. We then present a believable conjecture that is stronger than Conjecture 3.3 and show that it implies there are no asymptotically good LTCs. We need the following definition which says that a code is "t-repetitive" for small t if not too many coordinates are identical in all codewords. All known basic constructions of LTCs, such as Hadamard, Reed-Muller and those appearing in [10, 20, 24, 30] have no dual codewords of weight 2, hence are non-repetitive, or 1-repetitive according to the following definition.

Definition 3.8 (Bounded repetition)**.** Let $C \subseteq \mathbb{F}_2^n$ be a linear code. For $i_1, i_2 \in [n]$ we say that i_1 is a repetition of i_2 if for all $c \in C$ we have $c_{i_1} = c_{i_2}$, which happens if and only if there exists $u \in C_2^\perp$ s.t. $\mathrm{supp}(u) = \{i_1, i_2\}$. We say that C is t-repetitive if for every $i \in [n]$ it holds that $|\{j \mid j \text{ is a repetition of } i\}| \leq t$. We say that C is non-repetitive if there exists a constant $t > 0$ s.t. C is t-repetitive.

We now show that natural non-repetitive LTCs have bounded rate.

Theorem 3.9 (Natural non-repetitive 3-query LTCs have bounded rate)**.** *Let $C \subseteq \mathbb{F}_2^n$ be (α, Δ)-natural and t-repetitive s.t. $\Delta \geq 2t$. Then*

$$\mathrm{rate}(C) \leq \frac{1}{\alpha} \cdot \frac{\log(\Delta/(4t)) + 1}{\Delta/(4t)}.$$

Corollary 3.10 (No asymptotically good natural dense codes)**.** *Let $\alpha, t > 0$ be constants and $\Delta : \mathcal{N} \to \mathcal{N}$ be a function s.t. $\Delta(n) = \omega(1)$. Let $C = \{C_n\}_{n \in \mathbb{Z}}$ be a family of codes, where $C_n \subseteq \mathbb{F}_2^n$ is an $(\alpha, \Delta(n))$-natural code that is t-repetitive. Then*

$$\mathrm{rate}(C_n) \leq \frac{1}{\alpha} \cdot \frac{\log(\Delta(n)/(4t)) + 1}{\Delta(n)/(4t)} = o(1).$$

Intuitively, the following conjecture says that if C is an asymptotically good 3-query LTCs then a large part of C looks like a natural code with super-linear density. Note that Conjecture 3.3 implies that 3-query LTCs have a superlinear number of dual codewords of weight at most 3.

Conjecture 3.11 (LTCs contain natural non-repetitive punctured code)**.** *Let $\epsilon, \delta, \rho > 0$ be constants. Then there exist a function $\Delta : \mathcal{N} \to \mathcal{N}$ s.t. $\Delta(n) = \omega_{\epsilon,\delta,\rho}(1)$ and constants $\alpha, \beta, t > 0$ which depend only on ϵ, δ, ρ such that the following condition holds. If $C \subseteq \mathbb{F}_2^n$ is a $(3, \epsilon, \delta/3)$-LTC, where $\delta(C) \geq \delta$ and $\mathrm{rate}(C) \geq \rho$ then there exists $J \subseteq [n]$ s.t. $|J| \geq \beta n$ and $C|_J$ is $(\alpha, \Delta(n))$-natural and t-repetitive.*

Under this conjecture we can rule out the existence of asymptotically good LTCs altogether.

Corollary 3.12 (No Asymptotically good LTCs)**.** *Under Conjecture 3.11 there is no family of asymptotically good 3-query LTCs. Consequently (cf. Theorem A.1) there is no asymptotically good family of linear LTCs.*

Proof. Assume the contrary, i.e., there exists a family $C = \{C_n\}_{n \in \mathbb{Z}}$, where $C_n \subseteq \mathbb{F}_2^n$ is a $(3, \epsilon, \delta/3)$-LTC, $\delta(C_n) \geq \delta$ and $\mathrm{rate}(C_n) \geq \rho$ for some constants $\epsilon, \delta, \rho > 0$. Conjecture 3.11 implies that there exist a function $\Delta(n) : \mathcal{N} \to \mathcal{N}$ s.t. $\Delta(n) = \omega_{\epsilon,\delta,\rho}(1)$, constants $\alpha, \beta, t > 0$ which depend only on ϵ, δ, ρ and $J_n \subseteq [n]$ s.t. $|J_n| \geq \beta n$ and $(C_n)|_{J_n}$ is $(\alpha, \Delta(n))$-natural and t-repetitive.

Note that $\Delta(n) \geq 2t$ for sufficiently large n. Theorem 3.9 implies that $\mathrm{rate}(C_n) \leq \frac{\dim(C_n)}{\beta n} \leq \frac{1}{\beta\alpha} \cdot \frac{\log(\Delta(n)/(4t))+1}{\Delta(n)/(4t)} \leq o(1)$. Contradiction. □

Symmetric codes are regular We end this section by focusing on an interesting class of regular codes that has been investigated intensively in recent years (cf. [1, 24]) — the class of symmetric, or 1-transitive, LTCs.

Let G be a group of permutations over $[n]$. For $\pi \in G$ and $w = (w_1, w_2, ..., w_n) \in \mathbb{F}^n$ with some abuse of notation we let $\pi(w) = (w_{\pi^{-1}(1)}, ..., w_{\pi^{-1}(n)})$ be a π-permuted word. Note that since G is a group and $\pi \in G$ we have $\pi^{-1} \in G$. A linear code $\mathcal{C}$ is invariant under G if for every $\pi \in G$ and $c \in \mathcal{C}$ we have $\pi(c) \in \mathcal{C}$. Note that if $\mathcal{C}$ is invariant under G then also $\mathcal{C}^\perp$ is invariant under G. G is called 1-transitive if for all $i, j \in [n]$ we have $\pi \in G$ such that $\pi(i) = j$. A linear code $\mathcal{C}$ is 1-transitive if it is invariant under some 1-transitive permutation group G.

All relevant LTCs based on the "invariance" approach are regular. This is true since 1-transitivity is a minimal possible requirement for such LTCs and all 2-transitive codes, affine-invariant codes, linear invariant codes etc. are 1-transitive (for further information see [24]). It is not hard to show that 1-transitive codes are q-regular for every $q > 0$ (cf. Claim A.3) and this leads to the following corollary. Moreover, the next corollary shows that under Conjecture 3.3 there is no asymptotically good 1-transitive 3-query LTCs.

Corollary 3.13 (Dense 1-transitive LTCs are not asymptotically good)**.** *Let $C = \{C_n\}_{n\in\mathbb{Z}}$ be a family of codes, where $C_n \subseteq \mathbb{F}_2^n$ is 1-transitive.*

- *If $\sigma_3(C_n) = \omega(1)$ then*
$$\text{rate}(C_n) \leq \frac{2\log(\sigma_3(C_n)) + 2}{\sqrt{\sigma_3(C_n)}} = o(1).$$
- *Under Conjecture 3.3, if C_n is a $(3, \epsilon, \delta/3)$-LTC and $\delta(C_n) \geq \delta$ then $\text{rate}(C_n) = o(1)$.*

Proof. The first bullet follows from Claim A.3 (which implies that C_n is 3-regular) and Corollary 3.7. The second bullet follows from the first bullet. □

3.2 Limiting the rate of weak 2-query LDCs

The proof of our main theorems regarding LTCs, presented in the previous section, follow from an improved version of the rate-bound on 2-query LDCs due to [19]. In this section we present this improved version and discuss its corollaries for locally decodable codes.

The following lemma is due to Goldreich et al. [19], stated there as Lemma 3.3. This lemma had a crucial role in proving lower bounds for LDCs (see, e.g., the results of Goldreich et al. [19], Dvir and Shpilka [16], Obata [31], Woodruff [35, 36]). The lemma is used as a combinatorial core which analyzes the relation between the rate of a LDC and the number of tuples used in the decoding.

Let us first recall the definition of a singleton vector: let $e_i = 0^{i-1}10^{k-i}$ for $i \in [k]$. For a matrix G we let G_i denote the ith row of G. In this section we think of $G \in \mathbb{F}_2^{n\times k}$ as a generator matrix for some 2-query LDC C. We also relate k to $\dim(C)$ and n to the blocklength of C.

Lemma 3.14 (Lemma 3.3 in [19])**.** *Let $G \in \mathbb{F}_2^{n\times k}$ be a matrix and $\Delta \geq 1$. Suppose for every $i \in [k]$ there is a matching $M_i \subseteq \binom{n}{2}$, i.e., a set of disjoint pairs of indices (j_1, j_2), such that $G_{j_1} + G_{j_2} = e_i$. Moreover, suppose it holds that $\frac{\sum_{i=1}^k |M_i|}{k} \geq \Delta$. Then $k \leq \frac{n(\log n)}{2\Delta}$.*

Goldreich et al. [19] prove the lemma using the assumption that $\sum_i |M_i|$ is large. They go on to point out that in the context of LDCs one has a stronger assumption, namely, that every single matching M_i is large but this stronger assumption is not used. The following lemma, which

is the main technical contribution of this paper, improves upon Lemma 3.14 by using the stronger assumption on the size of individual matchings.

Lemma 3.15 (Main technical lemma)**.** *Let $G \in \mathbb{F}_2^{n \times k}$ be a matrix and $\Delta \geq 1$. Suppose for every $i \in [k]$ there is a matching $M_i \subseteq \binom{n}{2}$, i.e., a set of disjoint pairs of indices (j_1, j_2), such that $G_{j_1} + G_{j_2} = e_i$. Moreover, suppose for every $i \in [k]$ it holds that $|M_i| \geq \Delta$. Then $k \leq \frac{n(\log \Delta)+n}{\Delta}$.*

Notice that this lemma implies Lemma 3.14 and works for smaller densities. In particular, for any super-constant function $\Delta(n) \geq \omega(1)$ our lemma gives $\frac{k}{n} = o(1)$ but for $\Delta(n) \leq (\log n)/2$ Lemma 3.14 gives no nontrivial bounds.[2]

In Section 5 we prove Lemma 3.15 and in Section 6.1 we generalize it to arbitrary fields. The tightness of Lemmata 3.15, 3.14 is shown in Section 5.4.

Next we use Lemma 3.15 to limit the rate of weak 2-query LDCs, i.e., LDCs that allow correct decoding of message bits under the weak assumption that a super-constant (but sublinear) number of codeword bits are corrupted. We believe that Theorem 3.17 might be useful for improving the existing rate bounds of q-query locally decodable codes with $q \geq 3$ and subconstant corruption parameter $\delta = o(1)$. The point is that the best known lower bounds for q-query LDCs ($q \geq 3$) are obtained by way of reduction to 2-query LDC (with worse parameters) and applying the lower bound for 2-query LDC (see e.g. [35], [36]). However, the parameter δ of an LDC is strongly decreased in such a reduction and becomes $o(1)$ even if initially we have started the reduction from a q-query LDC with $\delta = \Omega(1)$.

The best known lower bound for 2-query LDCs is due to Goldreich et al. [19] who proved it for binary fields (see also [31]), it was generalized to general fields in [16]:

Theorem 3.16 ([16])**.** *Let $\mathbb{F}$ be any field. Let $C \subseteq \mathbb{F}^n$ be a linear $(2, \delta, \epsilon)$-LDC with $k = \dim(C)$. Then $n \geq 2^{\frac{\epsilon\delta k}{4}-1}$.*

Previous lower bounds on LDCs with $\delta = o(1)$ were not achieved because of lack of tight lower bounds on 2-query LDCs with very small but non-trivial δ, i.e., where $\omega(1) \leq \delta n \leq \log n$ (see Dvir [15] for motivation for such bounds). In Theorem 3.17 we give such a lower bound.

Theorem 3.17 (Main Theorem on LDCs)**.** *Let $\mathbb{F}$ be any field. If $C \subseteq \mathbb{F}^n$ is a $(2, \epsilon, \delta)$-LDC with $k = \dim(C)$ then*

$$n \geq 2^{\frac{\delta k}{32(1-\epsilon)}-1} \cdot \frac{1-\epsilon}{\delta}.$$

Corollary 3.18. *Let $\mathbb{F}$ be any field, $\epsilon > 0$ and $\delta : \mathcal{N} \to \mathcal{N}$ be a function s.t. $\delta(n) \geq \omega(1)$. Let $C = \{C_n\}_{n \in \mathbb{Z}}$ be a family of codes, where $C \subseteq \mathbb{F}^n$ is a $(2, \epsilon, \delta(n))$-LDC. Then $\mathrm{rate}(C_n) \leq O(\frac{\log \delta(n)}{\delta(n)}) = o(1)$.*

Proof. Let $k = \dim(C_n)$. Theorem 3.17 implies that $n \geq 2^{\frac{\delta(n)k}{32(1-\epsilon)}-1} \cdot \frac{1-\epsilon}{\delta(n)} \geq 2^{\frac{\delta(n)k}{32}-1} \cdot \frac{1}{\delta(n)}$. Hence $\delta(n)n \geq 2^{\frac{\delta(n)k}{32}-1}$. We conclude that $\mathrm{rate}(C_n) = k/n \leq O\left(\frac{\log \delta(n)}{\delta(n)}\right) = o(1)$. □

Remark 3.19. The above corollary says that there is no constant rate 2-query LDC s.t. $\delta(n) \cdot n = \omega(1)$. In contrast, the best known lower bound for 2-query LDCs (by Dvir and Shpilka [16]) does not give any non-trivial bound when $\delta(n) \cdot n \leq \log n$.

[2] Recall that we think of k as $\dim(C)$ and n is a blocklength of C, where C is a linear code. Hence $\dim(C) = k \leq n$ is a trivial bound in this case, in contrast to the bound $k/n = o(1)$.

4 Proof of Main Results for LTCs

In this section we prove our main results regarding LTCs — Theorems 3.6 and 3.9 — and show how they follow from the main technical Lemma 3.15.

We first prove an auxiliary Proposition 4.1 which is the main place where Lemma 3.15 is used. Then we show how Proposition 4.1 implies Theorem 3.9. Theorem 3.6 will follow from Theorem 3.9.

Proposition 4.1. *Let $C \subseteq \mathbb{F}_2^n$ be a t-repetitive code and let $I \subseteq [n]$ be a set of independent indices. Assume that for every $i \in I$ it holds that $|\{u \in C_3^\perp \mid i \in \mathrm{supp}(u)\}| \geq \Delta$. Then, $|I|/n \leq \frac{\log(\Delta/(2t))+1}{\Delta/(2t)}$.*

Proof. We start from showing the following claim.

Claim 4.2. *For every $i \in I$ there exists $M_i \subseteq \binom{n}{2}$ s.t. $|M_i| \geq \Delta/(2t)$ and the following condition holds. For every $(j_1, j_2) \in M_i$ we have $u \in C_3^\perp$, where $\mathrm{supp}(u) = \{i, j_1, j_2\}$ and for every $(j_1, j_2) \neq (j_1', j_2') \in M_i$ we have $\{j_1, j_2\} \cap \{j_1', j_2'\} = \emptyset$.*

Proof. Let $i \in I$. We construct the subset M_i iteratively. With some abuse of notation, for $S \subseteq [n]$ we say that $S \cap M_i = \emptyset$ if for all $x \in M_i$ we have $S \cap x = \emptyset$.

- $M_i := \emptyset$
- While there exists $u \in C_3^\perp$ s.t. $i \in \mathrm{supp}(u)$ and $\mathrm{supp}(u) \cap M_i = \emptyset$
  - $M_i := M_i \cup (\mathrm{supp}(u) \setminus \{i\})$

The construction of M_i implies that for every $(j_1, j_2), (j_1', j_2') \in M_i$ we have $u \in C_3^\perp$, where $\mathrm{supp}(u) = \{i, j_1, j_2\}$ and $\{j_1, j_2\} \cap \{j_1', j_2'\} = \emptyset$. If $|M_i| \geq \Delta/(2t)$ we are done.

Assume that $|M_i| < \Delta/(2t)$. With some abuse of notation let $\mathrm{supp}(M_i) = \{j \mid \exists j' \in [n] : \ (j, j') \in M_i\}$. We have $|\mathrm{supp}(M_i)| = 2|M_i| < \Delta/t$. By assumption, it holds that $|\{u \in C_3^\perp \mid i \in \mathrm{supp}(u)\}| \geq h$ and by construction for every $u \in C_3^\perp$ s.t. $i \in \mathrm{supp}(u)$ we have $(\mathrm{supp}(u) \setminus \{i\}) \cap \{j_1, j_2\} \neq \emptyset$ for some $(j_1, j_2) \in M_i$. Let $T_{i,j} = \{u \in C_3^\perp \mid i, j \in \mathrm{supp}(u)\}$. By pigeonhole principle we conclude that there exists $j \in [n]$ s.t. $j \in \mathrm{supp}(M_i)$ and $|T_{i,j}| > t$. Note that if $u_1, u_2 \in T_{i,j}$ and $u_1 \neq u_2$ then $\mathrm{supp}(u_1) \cap \mathrm{supp}(u_2) = \{i, j\}$ but $|\mathrm{supp}(u_1)| = |\mathrm{supp}(u_2)| = 3$. Clearly, $u_1 + u_2 \in C_2^\perp$. Letting $i_1, i_2 \in [n]$ be s.t. $\{i_1\} = \mathrm{supp}(u_1) \setminus \{i, j\}$ and $\{i_2\} = \mathrm{supp}(u_2) \setminus \{i, j\}$ we have that i_2 is a repetition of i_1. Hence for every $u \in T$ letting $i' \in [n]$ be s.t. $\{i'\} = \mathrm{supp}(u) \setminus \{i, j\}$ it holds that i' is a repetition of i_1, so there are $|T| > t$ repetitions of i_1. Contradiction. □

Claim 4.2 implies that for every $i \in I$ there exists a subset $M_i \subseteq \binom{n}{2}$ of disjoint pairs s.t. $|M_i| \geq \Delta/(2t)$ and for all $(j_1, j_2) \in M_i$ we have $u \in C_3^\perp$ s.t. $\mathrm{supp}(u) = \{i, j_1, j_2\}$.

Let $k = \dim(C)$. Let $G \in \mathbb{F}_2^{n \times k}$ be a generator matrix for C and assume without loss of generality (reordering of indices) that $I = \{1, 2, ..., |I|\}$. Assume without loss of generality that the first $|I|$ rows and the first $|I|$ columns of G form an identity matrix. [3]

Let $G' \in \mathbb{F}_2^{n \times |I|}$ be a submatrix of G obtained by removing all columns c of G which have $c|_I = 0^{|I|}$ (there are $k - |I|$ such columns). Note that the top $|I|$ rows of G' form an identity matrix $|I| \times |I|$. Moreover, for all $u \in C^\perp$ it holds that $u^T \cdot G' = 0$ since G' contains only the columns of G

[3] Do gaussian elimination on columns to get identity matrix in the first $|I|$ rows, since $\mathrm{rank}(G|_{|I| \times k}) = |I|$ the submatrix $G|_{|I| \times k}$ will contain the identity submatrix $|I| \times |I|$.

(i.e., the codewords of C). For the rest of the proof let e_i be a singleton vector in $\mathbb{F}_2^{|I|}$. Note that for all $i \in [|I|]$ it holds that $G'_i = e_i$.

We conclude that for all $i \in I$ we have a set $M_i \subseteq \binom{n}{2}$ of disjoint pairs s.t. $|M_i| \geq \Delta/(2t)$ and for all $(j_1, j_2) \in M_i$ we have $G'_{j_1} + G'_{j_2} = e_i$. Lemma 3.15 implies that $|I|/n \leq \frac{\log(\Delta/(2t))+1}{\Delta/(2t)}$. □

We are ready to prove Theorem 3.9.

Proof of Theorem 3.9. The fact that C is $(\alpha, \Delta(n))$-natural implies that there exists a set of independent indices $I \subseteq [n]$ s.t. $|I| \geq \alpha \cdot \dim(C)$ and for every $i \in I$ it holds that $\Delta_{3,i}(C) \geq \Delta(n)$. Since C is t-repetitive it follows that for every $i \in I$ we have $\left|\left\{u \in C^{\perp}_{\leq 2} \mid i \in \mathrm{supp}(u)\right\}\right| \leq t$.

Hence for every $i \in I$ we have $\left|\left\{u \in C^{\perp}_{3} \mid i \in \mathrm{supp}(u)\right\}\right| \geq \Delta - t \geq \Delta/2$. Proposition 4.1 says that $|I|/n \leq \frac{\log(\Delta/(4t))+1}{\Delta/(4t)}$ and so $\mathrm{rate}(C) = \frac{\dim(C)}{n} \leq \frac{1}{\alpha} \cdot \frac{\log(\Delta/(4t))+1}{\Delta/(4t)}$. □

Now we prove Theorem 3.6.

Proof of Theorem 3.6. Let $\sigma = \sigma_3(C)$ and note that $\sigma \geq 2$ is an integer since C is regular. Note that $\left|C^{\perp}_{1}\right| = 0$ since otherwise $C = \{0^n\}$ (C is 3-regular and hence in particular 1-regular). The fact that $\mathcal{C}$ is 3-regular implies that every index $i \in [n]$ has the same number of repetitions in C (see Definition 3.8). Let t be this number of repetitions per index. Let $k = \dim(C)$. Then there exists an independent set $I \subseteq [n]$ s.t. $|I| = k$, and in particular, all indices in I are not repetitions of each other. So, $|I| \cdot t = k \cdot t \leq n$. If $t \geq \sqrt{\sigma}/2$ then $\frac{k}{n} \leq \frac{1}{t} \leq \frac{2}{\sqrt{\sigma}}$ and we are done. Otherwise, $t < \sqrt{\sigma}/2$ and hence C is $(\sqrt{\sigma}/2)$-repetitive. We argue that C is $(1, \sigma)$-natural. This is true since for every $i \in I$ it holds that $\Delta_{3,i}(C) \geq \sigma$, because every index $i \in [n]$ it holds that $\Delta_{3,i}(C) \geq \sigma_3(C) = \sigma$.

Theorem 3.9 implies that $\mathrm{rate}(C) \leq \frac{\log(\sigma/4t)+1}{\sigma/4t} \leq \frac{2\log\sigma+2}{\sqrt{\sigma}}$. □

5 Proof of Main Technical Lemma 3.15

In this section we prove Lemma 3.15. We end the section by showing that Lemmas 3.15 and 3.14 are tight (Section 5.4).

Overview of proof We study the generating matrix $G \in \mathbb{F}_2^{n \times k}$ of a 2-query LDC of dimension k and blocklength n. We may assume without loss of generality that the first k rows contain the k singleton vectors $e_1, \ldots, e_k$, where e_i has 1 in position i and is 0 elsewhere. Notice that when the first column of G is removed, for each pair of indices $i \neq j$ used to decode the first message bit (i.e., $G_i + G_j = e_1$) we now have that the i and j rows of the smaller $n \times (k-1)$ matrix are identical. In other words, after removing column 1 we may partition the rows of the residual matrix, denoted $G|_{n \times ([k] \setminus \{1\})}$, into sets of equal rows. Typically such sets will have size either 2 or 1. The former correspond to rows participating in a query for decoding the first message bit and the latter correspond to all other rows. Now, if we go on to remove the second column from G we may expect to see in the residual matrix sets of equivalent rows of sizes between 1 and 4. The former sets correspond to rows not participating in any decoding of bits $1, 2$ and the latter include rows that participate both in decoding message-bit number 1 and number 2. Continuing in this manner we would expect the size of sets of equivalent rows to double with every removal of an additional column from G and this would show that after $\approx \log n$ column-removals all rows are equivalent, which means $k = O(\log n)$.

Of course, the description above is a gross oversimplification of what actually happens when columns are removed. The problem is that the size of different sets of equivalent rows can grow in arbitrary ways. To prove our lemma we rely on a simple fact — that whenever two equivalence classes "merge" into one larger class after removing a column of G, then at least one of them (the smaller) must double in size. This observation leads us to measure size of sets on a logarithmic scale and carry out an amortized analysis of the number of times sets (of equivalent rows) are merged upon removal of columns of the generating matrix. We shall explain how we remove columns from G after making a few preliminary definitions and claims used in our proof.

5.1 Equivalence Relation and Matchings

With some abuse of notation consider every set as a multiset if not stated otherwise. The size of the multiset is the number of elements in it including repetitions. We recall that for $w \in F^n$ and $S \subseteq [n]$ we let $w|_S$ to be the *restriction* of w to the subset S.

We define an equivalence relation over the set of rows of G.

Definition 5.1 (Equivalence relation and class)**.** Let $J \subseteq [k]$. For any $i, j \in [n]$ we say that $G_i \approx_J G_j$ if and only if $G_i|_{[k]\setminus J} = G_j|_{[k]\setminus J}$. Since $\approx_J$ is an equivalence relation over G it defines equivalence classes. Let $[G_i]_J$ be the equivalence class of G_i under J, i.e., $[G_i]_J = \{G_j \mid G_i \approx_J G_j\}$.

We let P_J denote the quotient set of the multiset G by $\approx_J$, i.e., $P_J = \{[G_{i_1}]_J, ..., [G_{i_m}]_J\}$. It holds that P_J is a partition of the multiset G hence we will also say that P_J is a *J-partition* of G.

Now we define the important concept, called *valid matchings*. The concepts "equivalence classes" and "valid matchings" are central in the proof of Lemma 3.15.

Definition 5.2 (i-Matching)**.** Let $J \subseteq [k]$ and $i \in [k]$. Let $M \subseteq \binom{n}{2}$. We say that M is an *i-matching* if for all pairs $(i_1, i_2) \in M$ it holds that $G_{i_1} + G_{i_2} = e_i$. We say that the matching M is *valid* for J if for all pairs $(i_1, i_2) \in M$ it holds that $G_{i_1}|_{[k]\setminus J} + G_{i_2}|_{[k]\setminus J} = (e_i)|_{[k]\setminus J}$.

For $a \in [n]$ we say that an element $G_a \in G$ *appears* in the pair (i_1, i_2) if either $a = i_1$ or $a = i_2$. We say that G_a *appears* in the matching M if it appears in at least one pair of M.

Recall that for every $i \in [k]$ we have an i-matching M_i s.t. $|M_i| \geq \Delta$. Note that for every $i \in [k]$ it holds that every element of G appears at most once in the matching M_i. The following two simple claims summarize the effect of projection on the equivalence classes and matchings.

Claim 5.3 (Projection does not affect non-projected matchings)**.** *Let $J \subseteq [k]$ and $i \in [k] \setminus J$. If M is an i-matching then M is valid for J.*

Proof. This is true since for all pairs $(i_1, i_2) \in M$ we have $G_{i_1} + G_{i_2} = e_i$ hence $G_{i_1}|_{[k]\setminus J} + G_{i_2}|_{[k]\setminus J} = (e_i)|_{[k]\setminus J}$. □

Claim 5.4 (Projection implies Collapse of Equivalence Classes)**.** *Let $J \subseteq [k]$ and $e_i = G_j + G_{j'}$. If $i \in J$ then $G_j \approx_J G_{j'}$, or equivalently, $[G_j]_J = [G_{j'}]_J$.*

Proof. If $G_j + G_{j'} = e_i$ then $G_j|_{[k]\setminus J} + G_{j'}|_{[k]\setminus J} = 0$. So, $G_j|_{[k]\setminus J} = G_{j'}|_{[k]\setminus J}$ hence $G_j \approx_J G_{j'}$. □

5.2 Selection of columns to be removed from the generating matrix

In this section we describe the process by which columns of G are removed. We start with an explanation of the intuition behind this selection process. Recall that our goal is to upper-bound the number k. We start from the definition of small multisets and good matchings.

Definition 5.5 (Small Multisets and Good Matching)**.** A multiset S is called *small* if $|S| < \Delta$ and otherwise it is called *large*. We say that the i-matching M_i is J-*good* if $i \in [k] \setminus J$ and for all edges $(j, j') \in M_i$ it holds that at least one of $[G_j]_J$, $[G_{j'}]_J$ is a small multiset.

Let $J = \{i_1, i_2, ..., i_h\} \subseteq [k]$ and for $t \leq h$ let $J(t) = \{i_1, i_2, ..., i_t\}$ and $J(0) = \emptyset$. Assume that for all $t \leq h$ it holds that the i_t-matching M_{i_t} is $J(t-1)$-good. By Definition 5.5 all pairs of M_{i_t} "touch" many small subsets in $P_{J(t-1)}$ and note that Claim 5.4 implies that a large number of pairs of multisets $[G_{j_1}]_{J(t-1)}$ and $[G_{j_2}]_{J(t-1)}$ collapse into the single multiset $[G_{j_1}]_{J(t)}$. In this way, we can expect that for all $t \leq h$ the size of $P_{J(t)}$ will be much smaller than the size of $P_{J(t-1)}$. Note also that $\left|P_{J(h)}\right| \geq 1$ and $\left|P_{J(0)}\right| \leq n$. Hence the subset J cannot be too large. Later on in the proof we will upper-bound $|J|$ and on the other side we will argue that $|[k] \setminus J|$ is small, obtaining the upper bound on k.

The following algorithm constructs the set $J \subseteq [k]$. Roughly we maintain an iteration number t and set $J(t)$ which grows slowly. For analysis it is better to denote sets separately.

Construction of J

- $t := 0$
- $J(t) := \emptyset$
- While there exists $i \in [k] \setminus J(t)$ s.t. the matching M_i is $J(t)$-good
  - $J(t+1) := J(t) \cup \{i\}$
  - $t := t + 1$
- $J := J(t)$
- return J

For the rest of the proof, we assume that the algorithm returns the subset $J = \{i_1, i_2, ..., i_h\}$, where i_t is the element added in the t'th iteration of the algorithm. Notice $J(t) = \{i_1, i_2, ..., i_t\}$ and $J(0) = \emptyset$. We have two immediate but crucial properties, stated formally in Claims 5.6 and 5.7.

Claim 5.6. *For every $t \in [h]$ it holds that the i_t-matching M_{i_t} is $J(t-1)$-good.*

Claim 5.7. *For every $i \in [k] \setminus J$ it holds that the i-matching M_i is not J-good, i.e., there exists a pair $(j, j') \in M_i$ such that both multisets $[G_j]_J$ and $[G_{j'}]_J$ are large.*

Proof. The claim follows from the construction of J. If for some $i \in [k] \setminus J$ the i-matching is J-good then the construction of J would not stop. □

5.3 Completing the proof of Main Technical Lemma 3.15

In this section we present Lemmas 5.8 and 5.9. The proof of the Combinatorial Lemma 3.15 will follow immediately from these two lemmas. The rest of this section is devoted to the proofs of the two sub-lemmas stated next.

Lemma 5.8 (Bound on $k - |J|$)**.** *It holds that* $k - |J| \leq \frac{n}{\Delta}$.

Lemma 5.9 (Bound on $|J|$)**.** *It holds that* $|J| \leq \frac{n \log \Delta}{\Delta}$.

The proof of Lemma 3.15 follows by a combination of Lemmas 5.8 and 5.9.

Proof of Lemma 3.15. We have $k = |J| + (k - |J|) \leq \frac{n \log \Delta + n}{\Delta}$. □

In Sections 5.3.1 and 5.3.2 we prove Lemmas 5.8 and 5.9, correspondingly.

5.3.1 Proof of Lemma 5.8

Let $m = k - |J|$ and assume without loss of generality that $J = \{m+1, m+2, ..., k\}$. Let r be the number of large multisets in P_J and assume without loss of generality that the large multisets of P_J are $[G_1]_J, ..., [G_r]_J$. We have that $r \leq n/\Delta$ since the number of rows is $|G| = n$ and every large subset has size at least Δ.

Claim 5.7 says that for every $i \in [m] = [k] \setminus J$ the matching M_i is not J-good, i.e., there exists at least one edge $(j, j') \in M_i$ such that both $[G_j]_J$ and $[G_{j'}]_J$ are large, meaning $|[G_j]_J| \geq \Delta$ and $|[G_{j'}]_J| \geq \Delta$. Note that in this case $G_j|_{[m]} + G_{j'}|_{[m]} = e_i|_{[m]}$, i.e., $e_i|_{[m]} \in \text{span}\left\{G_j|_{[m]}, G_{j'}|_{[m]}\right\}$.

We conclude that for every $i \in [m]$ it holds that $e_i|_{[m]} \in \text{span}\left\{G_j|_{[m]} \mid j \in [r]\right\}$. We argue that $m \leq r$. To see this recall that $G_1|_{[m]}, ..., G_t|_{[m]} \in \mathbb{F}_2^m$ and note that for every $i \in [m]$ we have $e_i|_{[m]} \in \text{span}\left\{G_j|_{[m]} \mid j \in [r]\right\}$. Thus $m = \dim(\text{span}\left\{e_i|_{[m]} \mid i \in [m]\right\}) \leq \dim(\text{span}\left\{G_1|_{[m]}, ..., G_r|_{[m]}\right\}) \leq r$.

We conclude that $k - |J| = m \leq r \leq n/\Delta$ and this completes the proof of Lemma 5.8.

5.3.2 Proof of Lemma 5.9

In this section we prove that $|J| \leq \frac{n \log \Delta}{n}$. We first define the valence of a multiset.

Definition 5.10 (Valence of the Multiset)**.** Given a multiset $S \neq \emptyset$ its *valence* $v(S)$ is defined as $\lfloor \log |S| \rfloor$.

Remark 5.11. Recall from Definition 5.5 that a multiset S is large if $v(S) \geq \log \Delta$.

Definition 5.12 (Consumptions - Edges vs. Rows)**.** Let $t \leq h$. For $i_t \in J(t) \setminus J(t-1)$ let M_{i_t} be the i_t-matching and $e = (m, m') \in M_{i_t}$. In this case, we say that the edge e was *consumed* in iteration t.

If $|[G_m]_{J(t-1)}| \leq |[G_{m'}]_{J(t-1)}|$ we say that G_m *consumed* the edge e in iteration t. Note that if $|[G_m]_{J(t-1)}| = |[G_{m'}]_{J(t-1)}|$ then both G_m and $G_{m'}$ consumed e in iteration t.

Since every row of G appears in any given matching at most once, we know that every row consumes at most one edge in iteration t, hence we can define the following indicator variables. Let $E_{(m,m'),t}$ be the indicator for the event that the edge (m, m') was consumed in iteration t. Let $E_t = \sum_{e \in M_{i_t}} E_{e,t}$ be the number of edges that were consumed in time t and let $E_{\leq t} = \sum_{i=1}^{t} E_i$ be the number of edges that were consumed up to time t.

Similarly, for $l \in [n]$ we let $R_{l,t}$ be the indicator for the event that the row G_l consumes some edge in time t. Let $R_t = \sum_{l\in[n]} R_{l,t}$ be the number of rows that consume an edge in iteration t, and let $R_{\leq t} = \sum_{i=1}^{t} R_i$ be the number of consumptions which happen up to time t.

The intuition behind this definition is as follows. The consumption of edges is tightly related to the consumption by rows. The numbers are roughly equal since when an edge is consumed, it is consumed by at least one (and at most two) rows. So, on the one side there are many edges that were consumed and on the other side, as shown in Proposition 5.14, every row can not consume too many edges, since the valence of an equivalence class containing the row is increased at least by one after consumption.

We go on to present Claim 5.13 and Propositions 5.14 and 5.15. Then we prove Lemma 5.9. We end this section by proving Claim 5.13 and Propositions 5.14 and 5.15.

Claim 5.13 (Consumption implies increase of valence)**.** *Let $t < h$, $M_{i_{t+1}}$ be the i_{t+1}-matching and $(j, j') \in M_{i_{t+1}}$. It holds that*

- *at least one of $G_j, G_{j'}$ consumes the edge (j, j') in iteration $t + 1$, and*
- *if G_j consumes the edge (j, j') in iteration t then $v([G_j]_{J(t+1)}) \geq 1 + v([G_j]_{J(t)})$.*

Proposition 5.14 (Row Consumption)**.** *For every $t \leq h$ it holds that $R_{\leq t} \leq n \log \Delta$.*

Proposition 5.15 (Edge Consumption)**.** *For every $t \leq h$ we have $E_{\leq t} \geq \Delta \cdot |J(t)| = \Delta \cdot t$.*

We are ready now to prove the Lemma 5.9, which says $|J| \leq \frac{n \log(\Delta)}{\Delta}$.

Proof of Lemma 5.9. Recall that $h = |J|$. Proposition 5.14 implies that $R_{\leq h} \leq n \log \Delta$. Proposition 5.15 implies that the total number of edge consumptions $E_{\leq h}$ is at least $h \cdot \Delta$. Claim 5.13 implies that $E_{\leq h} \leq R_{\leq h}$. We conclude that $h \cdot \Delta \leq E_{\leq h} \leq R_{\leq h} \leq n \cdot \log \Delta$, and thus $|J| = h \leq \frac{n \log(\Delta)}{\Delta}$. □

Now we prove Claim 5.13 and Propositions 5.14 and 5.15.

Proof of Claim 5.13. Claim 5.3 implies that $M_{i_{t+1}}$ is valid for $J(t)$ and, in particular,

$$G_j|_{J(t)} + G_{j'}|_{J(t)} = e_{i_{t+1}}.$$

Hence $[G_j]_{J(t)} \neq [G_{j'}]_{J(t)}$ since otherwise, if $[G_j]_{J(t)} = [G_{j'}]_{J(t)}$ then it holds that $G_j|_{J(t)} + G_{j'}|_{J(t)} = 0 \neq e_{i_{t+1}}$. Clearly, either $|[G_j]_{J(t)}| \leq |[G_{j'}]_{J(t)}|$ or $|[G_{j'}]_{J(t)}| \leq |[G_j]_{J(t)}|$ hence, by definition, at least one of $G_j, G_{j'}$ consumes the edge in iteration $t + 1$. This completes the proof of the first bullet.

For the second bullet, by assumption, we have $|[G_j]_{J(t)}| \leq |[G_{j'}]_{J(t)}|$. Claim 5.4 implies that

$$[G_j]_{J(t+1)} = [G_{j'}]_{J(t+1)} \text{ but } [G_j]_{J(t)} \neq [G_{j'}]_{J(t)}.$$

This means $[G_j]_{J(t)} \cup [G_{j'}]_{J(t)} \subseteq [G_j]_{J(t+1)}$ and so, $|[G_j]_{J(t)}| + |[G_{j'}]_{J(t)}| \leq |[G_j]_{J(t+1)}|$. The fact that $|[G_j]_{J(t)}| \leq |[G_{j'}]_{J(t)}|$ implies that

$$2|[G_j]_{J(t)}| \leq |[G_j]_{J(t)}| + |[G_{j'}]_{J(t)}| \leq |[G_j]_{J(t+1)}|.$$

It follows that

$$1 + \lfloor \log |[G_j]_{J(t)}| \rfloor = \lfloor 1 + \log |[G_j]_{J(t)}| \rfloor = \lfloor \log(2|[G_j]_{J(t)}|) \rfloor \leq \lfloor \log |[G_j]_{J(t+1)}| \rfloor.$$

We conclude that

$$1 + v([G_j]_{J(t)}) = 1 + \lfloor \log |[G_j]_{J(t)}| \rfloor \leq \lfloor \log |[G_j]_{J(t+1)}| \rfloor = v([G_j]_{J(t+1)}).$$

□

Proof of Proposition 5.14. We first claim that for every row $G_l \in G$ it holds that $\sum_{t=1}^{h} R_{l,t} \leq \log \Delta$. Note that for all $v([G_i]_{J(\cdot)})$ is monotonic non-decreasing, i.e., $v([G_i]_{J(0)}) \leq v([G_i]_{J(1)}) \leq ... \leq v([G_i]_{J(h)})$. This is true because $[G_i]_{J(0)} \subseteq [G_i]_{J(1)} \subseteq ... \subseteq [G_i]_{J(h)}$.

We argue that if for some time $t \leq h$ we have $v([G_l]_{J(t)}) \geq \log \Delta$ then for every t' such that $t < t' \leq h$ we have $R_{l,t'} = 0$ and $v([G_l]_{J(t')}) \geq \log \Delta$. Assume the contrary. Clearly, for every $t' > t$ we have $v([G_l]_{J(t')}) \geq v([G_l]_{J(t)}) \geq \log \Delta$ since $[G_l]_{J(t)} \subseteq [G_l]_{J(t')}$. So, there exists $t' > t$ s.t. $v([G_l]_{J(t'-1)}) \geq \log \Delta$ but $R_{l,t'} = 1$. Note that $i_{t'} \in J$. From the definition of "consumption" (Definition 5.12) it follows that there exists an edge $(l, l') \in M_{i_{t'}}$ such that $|[G_l]_{J(t'-1)}| \leq |[G_{l'}]_{J(t'-1)}|$. But then $|[G_{l'}]_{J(t'-1)}| \geq |[G_l]_{J(t'-1)}| \geq \Delta$. In this case, the matching $M_{i_{t'}}$ is not $J(t'-1)$-good, contradicting Claim 5.6. We conclude that if for some time $t \leq h$ we have $v([G_l]_{J(t)}) \geq \log \Delta$ then for every t' such that $t < t' \leq h$ we have $R_{l,t'} = 0$ and $v([G_l]_{J(t')}) \geq v([G_l]_{J(t)}) \geq \log \Delta$.

Now, in iteration 0 the valence of $[G_l]_\emptyset$ is at least 0. Claim 5.13 implies that if G_l consumes an edge in iteration $t' \leq h$ then $v([G_l]_{J(t')}) \geq v([G_l]_{J(t'-1)}) + 1$. This means that if $R_{l,t'} = 1$ then $v([G_l]_{J(t')}) \geq v([G_l]_{J(t'-1)}) + R_{l,t'}$. Note that if $R_{l,t} = 0$ it is also true that $v([G_l]_{J(t')}) \geq v([G_l]_{J(t'-1)}) + R_{l,t'}$. Hence for every $t' \leq h$ we have $v([G_l]_{J(t')}) \geq v([G_l]_{J(t'-1)}) + R_{l,t'}$. Recalling $v([G_l]_{J(0)}) \geq 0$ it follows that for every $t' \leq h$ we have $\sum_{t=1}^{t'} R_{l,t'} \leq v([G_l]_{J(t')})$.

We conclude that for every row $G_l \in G$ it holds that $\sum_{t=1}^{h} R_{l,t} \leq \log \Delta$. Recalling that $|G| = n$, we have

$$R_{\leq t} = \sum_{i=1}^{t} \sum_{l \in [n]} R_{l,t} = \sum_{l \in [n]} (\sum_{i=1}^{t} R_{l,t}) \leq \sum_{l \in [n]} \log \Delta = n \log \Delta.$$

□

Proof of Proposition 5.15. Recall that $J(t) = \{i_1, i_2, ..., i_t\}$ is an ordered set. By construction of J, for every $t \leq h$ it holds that $i_t \in [k] \setminus J(t-1)$. Claim 5.13 implies that all edges of the i_t-matching M_{i_t} are consumed in iteration t. Thus for every $t \leq h$ we have $|E_t| \geq |M_{i_t}| \geq \Delta$. Recalling that $|M_{i_l}| \geq \Delta$ we conclude

$$E_{\leq t} = \sum_{l=1}^{t} E_l = \sum_{l=1}^{t} |M_{i_l}| \geq t \cdot \Delta.$$

□

5.4 Tightness of Lemmas 3.15 and 3.14

We end our discussion of Lemmas 3.15 and 3.14 by showing that each of them is tight.

Lemma 5.16 (Tightness of Lemma 3.15)**.** *Let $\Delta : \mathcal{N} \to \mathcal{N}$ be a function s.t. $\Delta(n) \leq n/2$. Then there exists a matrix $G \in \mathbb{F}_2^{n \times k}$ and for every $i \in [k]$ there exists a set of disjoint pairs of indices $M_i \subseteq \binom{n}{2}$ such that*

- *For every $(i_1, i_2) \in M_i$ we have $G_{i_1} + G_{i_2} = e_i$,*

- *For every $i \in [k]$ we have $|M_i| \geq \Delta(n)$,*
- *Furthermore, it holds that $k = \frac{n \log \Delta(n) + n}{2\Delta(n)}$.*

Remark 5.17. We assume that $\Delta(n)$, $\frac{n}{2\Delta(n)}$ and $\log \Delta(n)$ are integers. Otherwise, we would work in terms of $\lfloor \Delta(n) \rfloor$, $\lfloor n/2\Delta(n) \rfloor$ and $\lfloor \log \Delta(n) \rfloor$.

Proof of Lemma 5.16. Let $k = \frac{n(\log(\Delta(n))+1)}{2\Delta(n)}$. Let $k_1 = \log(\Delta(n)) + 1$ and $n_1 = 2^{k_1} = 2\Delta(n)$. Let $H \in \mathbb{F}_2^{n_1 \times k_1}$ be the generator matrix of the Hadamard code (with blocklength n_1 and dimension k_1).

We show how to construct the required matrix $G \in \mathbb{F}_2^{n \times k}$. Informally, G will be constructed from $\frac{n}{2\Delta(n)}$ copies of matrix H and they will be located along the diagonal of the matrix G.

1. Initialization $G := 0^{n \times k}$.
2. For $\texttt{row} = 1$ to n and for $\texttt{column} = 1$ to k
   (a) Copy the matrix H to the submatrix of G with coordinates
   $$[\texttt{row}, ..., \texttt{row} + n_1 - 1] \times [\texttt{column}, ..., \texttt{column} + k_1 - 1]$$
   (b) $\texttt{row} := \texttt{row} + n_1$
   (c) $\texttt{column} := \texttt{column} + k_1$

We argue that for every $i \in [k]$ there are at least $\Delta(n)$ disjoint pairs $(i_1, i_2) \in [n] \times [n]$ such that $G_{i_1} + G_{i_2} = e_i$. Let $i \in [k]$. Assume without loss of generality that $i \in [k_1]$.[4] It is sufficient to show that there are $n_1/2 = \Delta(n)$ disjoint pairs $(i_1, i_2) \in [n_1] \times [n_1]$ such that $G_{i_1} + G_{i_2} = e_i$. Recall that $G|_{[n_1] \times [k_1]} = H \in \mathbb{F}_2^{n_1 \times k_1}$ is the generating matrix for the Hadamard code hence contains $n_1/2 = \Delta(n)$ disjoint pairs $(i_1, i_2) \in [n_1] \times [n_1]$ such that $H_{i_1} + H_{i_2} = e_i$. This true for $G_{[n_1] \times [k]}$ as well since $G|_{[n_1] \times [k]}$ is zero outside the submatrix $[n_1] \times [k_1]$. □

We now show that it is crucial to take into account the fact that *every* matching, not just an average on, is large. In particular, we show that if this fact is not taken into account then the lower bound of Goldreich et al. [19] is tight.

Lemma 5.18 (Tightness of Lemma 3.14)**.** *Let $\Delta : \mathcal{N} \to \mathcal{N}$ be a function s.t. $\Delta(n) \leq n/2$. Then there exists matrix $G \in \mathbb{F}_2^{n \times k}$ and for every $i \in [k]$ there exists a set of disjoint pairs of indices $M_i \subseteq \binom{n}{2}$ such that*

- *For every $(i_1, i_2) \in M_i$ we have $G_{i_1} + G_{i_2} = e_i$,*
- *$\sum_{i=1}^{k} |M_i| = k \cdot \Delta(n)$, i.e., in the average $|M_i| = \Delta(n)$,*
- *Furthermore, it holds that $k = \frac{n \log n}{2\Delta(n)}$.*

Remark 5.19. Once again, we assume that $\Delta(n)$, $\frac{n}{2\Delta(n)}$ and $\log \Delta(n)$ are integers. Otherwise, we would work in terms of $\lfloor \Delta(n) \rfloor$, $\lfloor n/2\Delta(n) \rfloor$ and $\lfloor \log \Delta(n) \rfloor$.

[4] It can be assumed without loss of generality since the matrix G was constructed in a completely symmetric way.

Proof of Lemma 5.18. Note that $\Delta(n) \leq n/2$ since a single matching can not be larger. Let $k = \frac{n \log n}{2\Delta(n)}$. Let $k_1 = \log n$ and $k_2 = k - k_1$. Let $H \in \mathbb{F}_2^{n \times k_1}$ be the Hadamard generator matrix. Let $L = 0^{n \times k_2}$ be a zero matrix. Let $G = H \circ L$ (we took H and appended L). We argue that for every $i \in [k_1]$ there are $n/2$ distinct pairs $G_{i_1}, G_{i_2} \in G$ such that $G_{i_1} + G_{i_2} = e_i$. This is true since for every $i \in [k_1]$ there are $n/2$ distinct pairs $H_{i_1}, H_{i_2} \in H$ such that $H_{i_1} + H_{i_2} = e_i$ and L is a zero matrix and hence does not affect this property when it is appended to H. Note also that $\sum_{i=k_1+1}^{k} |M_i| = 0$ because of zero matrix L. Hence $\sum_{i=1}^{k} |M_i| = \sum_{i=1}^{k_1} |M_i| = (n/2) \cdot k_1 = n \log(n)/2 = k \cdot \Delta(n)$. □

6 Limiting the rate of weak 2-query LDCs — Proof of Theorem 3.17

In this section we prove Theorem 3.17. We first present Lemmas 6.1 and 6.2. The proof of Theorem 3.17 will follow by a combination of these lemmas.

Lemma 6.1 (Combinatorial Lemma for General Field)**.** *Let $\mathbb{F}$ be any field and let $G \in \mathbb{F}^{n \times k}$. For every $i \in [k]$ let $M_i \subseteq [n] \times [n]$ be a set of disjoint pairs of indices such that $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$ for every $(j_1, j_2) \in M_i$. Assume that for all $i \in [k]$ we have $|M_i| \geq \Delta$, where $\Delta \geq 1$. Then,*

$$k \leq \frac{16n \log \Delta + 16n}{\Delta}$$

The proof of Lemma 6.1 is postponed to Section 6.1. The following Lemma 6.2 is due to Obata [31]. The main result of [31] (Lemma 6.2) provides a tight analysis of the number of matchings which has a 2-query LDC. Although Obata [31] proved this result over the binary field but generalization to arbitrary fields is straightforward. To make the paper self-contained we give a proof-sketch of Lemma 6.2, stated for all fields, in Section 6.2.[5]

Lemma 6.2. *Let $C \subseteq \mathbb{F}^n$ be a $(2, \epsilon, \delta)$-LDC and $k = \dim(C)$. Let $G \in \mathbb{F}^{n \times k}$ be a generator matrix for C. Then for every $i \in [k]$ there exists $M_i \subseteq [n] \times [n]$ of disjoint pairs s.t. $|M_i| \geq \frac{1}{2} \cdot \frac{\delta n}{1 - \frac{|\mathbb{F}|}{|\mathbb{F}|-1} \cdot \epsilon}$ and $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$ for every $(j_1, j_2) \in M_i$.*

We are ready to prove Theorem 3.17.

Proof of Theorem 3.17. Let $G \in \mathbb{F}^{n \times k}$ be a generator matrix for C. Let $\Delta = \frac{1}{2} \cdot \frac{\delta n}{1 - \epsilon \cdot |\mathbb{F}|/(\mathbb{F}-1)} \geq \frac{1}{2} \cdot \frac{\delta n}{1-\epsilon}$. Lemma 6.2 implies that for every $i \in [k]$ there is a set $M_i \subseteq [n] \times [n]$ of disjoint pairs such that $|M_i| \geq \Delta$ and for every $(j_1, j_2) \in M_i$ we have $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$. Then Lemma 6.1 implies that $k \leq \frac{16n \log \Delta + 16n}{\Delta}$.

We conclude that $k \leq \frac{16n \log(\frac{1}{2} \cdot \frac{\delta n}{1-\epsilon}) + 16n}{(\frac{1}{2} \cdot \frac{\delta n}{1-\epsilon})} \leq \frac{32 \log(\frac{\delta n}{1-\epsilon}) + 32}{\frac{\delta}{1-\epsilon}}$. Hence $\frac{\delta k}{32(1-\epsilon)} - 1 \leq \log(\frac{\delta n}{1-\epsilon})$ and $n \geq 2^{\frac{\delta k}{32(1-\epsilon)} - 1} \cdot \frac{1-\epsilon}{\delta}$. □

[5] It might be the case that this lemma was already stated for all fields as we wrote. We did not verify it.

6.1 Proof of Lemma 6.1 – General field $\mathbb{F}$

In this section we prove Lemma 6.1. We need the following Lemma 6.3 due to Dvir and Shpilka [16, Lemma 2.5].

Lemma 6.3 ([16]). *Let $\mathbb{F}$ be any field and let $G \in \mathbb{F}^{n\times k}$. For every $i \in [k]$ let $M_i \subseteq [n] \times [n]$ be a set of disjoint pairs of indices, such that $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$ for every $(j_1, j_2) \in M_i$. Then, there exist $G'' \in \mathbb{F}_2^{n\times k}$ and k sets $M_1'', ..., M_k'' \subseteq \binom{n}{2}$ of disjoint pairs, such that:*

- *For every $(j_1, j_2) \in M_i''$ it holds that $G''_{j_1} \oplus G''_{j_2} = e_i$,*
- *$\sum_{i=1}^k |M_i| \leq 2\sum_{i=1}^k |M_i''| + n$,*
- *For every $i \in [k]$ it holds that $M_i'' \subseteq M_i$*

Remark 6.4. The only difference between [16, Lemma 2.5] and Lemma 6.3 is that the third bullet was not explicitly stated in [16, Lemma 2.5]. However, it can be readily verified that for all $i \in [k]$ it holds that $M_i'' \subseteq M_i$. We briefly explain this and refer a reader to [16, Lemma 2.5] for notation and definitions.

This is true since Dvir and Shpilka [16, Lemma 2.5] showed the reduction from a general field $\mathbb{F}$ to binary field in two steps. In the first step some pairs were removed from the matchings $M_1, ..., M_k$ resulting in the matchings $M_1', ..., M_k'$ s.t. $M_i' \subseteq M_i$. In the second step they suggested a transformation from $\mathbb{F}$ to $\mathbb{F}_2$ s.t. for all $i \in [k]$ some pairs from M_i' were removed resulting in M_i''. So, they obtained matchings $M_1'', ..., M_k''$ s.t. $M_i'' \subseteq M_i'$.

Proof of Lemma 6.1. Let $M_1, ..., M_k \subseteq [n]\times[n]$ be matchings s.t. for every $i \in [k]$ we have $|M_i| \geq \Delta$. We can assume w.l.o.g. that for every $i \in [k]$ we have $|M_i| = \Delta$ (otherwise remove some pairs from M_i).

Lemma 6.3 implies the existence of $G'' \in \mathbb{F}_2^{n\times k}$ and matchings M_i'' s.t. $\Delta \cdot k = \sum_{i=1}^k |M_i| \leq 2\sum_{i=1}^k |M_i''| + n$ and for every $i \in [k]$ it holds that $M_i'' \subseteq M_i$, which means $|M_i''| \leq |M_i| \leq \Delta$. Moreover, for every $(j_1, j_2) \in M_i''$ it holds that $G''_{j_1} \oplus G''_{j_2} = e_i$. We say that the matching M_i'' is bad if $|M_i''| < \Delta/4$. If the number of bad matchings is more than $3k/4$ then $fk \leq 2\sum_{i=1}^k |M_i''| + n \leq 2((3k/4)(\Delta/4) + (k/4)\Delta) + n \leq (14/16)k\Delta + n = (7/8)k\Delta + n$. In this case we get $k \leq 8n/\Delta$ and we are done since $8n/\Delta \leq \frac{16n\log\Delta+16n}{\Delta}$. Otherwise, the number of bad matchings is less than $3k/4$, hence there are at least $k/4$ good matchings (those with $|M_i''| \geq \Delta/4$).

Assume w.l.o.g. that for all $i \in [k/4]$ the matching M_i'' is good, i.e., $|M_i''| \geq \Delta/4$. Consider $A'' = G''|_{n\times(k/4)} \in \mathbb{F}_2^{n\times(k/4)}$ and note that for every $i \in [k/4]$ and $(j_1, j_2) \in M_i''$ we have $A''_{j_1} \oplus A''_{j_2} = e_i$. Lemma 3.15 implies that $k/4 \leq \frac{n\log(\Delta/4)+n}{\Delta/4} \leq \frac{4n\log\Delta+4n}{\Delta}$. We conclude that $k \leq \frac{16n\log\Delta+16n}{\Delta}$. □

6.2 Proof of Lemma 6.2

In this section we give a sketch of the proof of Lemma 6.2. We start from the (non-standard) definition of non-redundant matchings.

Definition 6.5 (Non-redundant Edges and Matching)**.** Let $G \in \mathbb{F}^{n\times k}$ and let $i \in [k]$. We say that $(j_1, j_2) \in [n] \times [n]$ is a non-redundant i-edge if we have $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$, and moreover, if $e_i \in \text{span}\{G_{j_1}\}$ or $e_i \in \text{span}\{G_{j_2}\}$ then $j_1 = j_2$. We say that $E_i \subseteq [n] \times [n]$ is an i-set of non-redundant edges if for every $(j_1, j_2) \in E_i$ we have that (j_1, j_2) is a non-redundant i-edge. We say that $M_i \subseteq E_i$ is a non-redundant i-matching if every $i \in [n]$ appears in at most one edge of M_i.

Note that Definition 6.5 allows self-loops in the non-redundant matchings, and we demonstrate this in the next example.

Example 6.6. Let $G \in \mathbb{F}_2^{3\times 3}$ such that $G = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$, i.e., $G_1 = (100)$ is a first row, $G_2 = (111)$ is a second row and $G_3 = (011)$ is a third row. Then, $E_1 = \{(1,1),(2,3)\}$ is a 1-set of (non-redundant) edges and a non-redundant 1-matching $M_1 = E_1$. Note that M_1 is a (legal) non-redundant 1-matching, e.g., 1 appears only in the single edge $(1,1)$ of M_1. Moreover, a 2-set and a 3-set of non-redundant edges are empty, i.e., $E_2 = E_3 = \emptyset$.

The intuition behind the definition of "non-redundant" edges (Definition 6.5) is as follows. Let C be a 2-query linear LDC and G be its generator matrix. Without loss of generality [19], a 2-query decoder for C recovers the message bit i by querying (at most) two bits indexed by j_1,j_2 s.t. $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$. However, if it holds that $e_i \in \text{span}(G_{j_1})$ or $e_i \in \text{span}(G_{j_2})$ we can assume w.l.o.g. [19] that the decoder queries (at the same invocation) at most one from j_1,j_2. So, if $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$ but $e_i \notin \text{span}\{G_{j_1}\}$ and $e_i \notin \text{span}\{G_{j_2}\}$ then (j_1, j_2) is a non-redundant i-edge; and if $e_i \in \text{span}\{G_{j_1}\}$ (or $e_i \in \text{span}\{G_{j_2}\}$) then (j_1, j_1) (or (j_2, j_2)) is a non-redundant i-edge.

We continue by recalling an implicit argument from [19] (see also [16]).

Claim 6.7 (Implicit in [19])**.** *Let $C \subseteq \mathbb{F}^n$ be a $(2, \epsilon, \delta)$-LDC and $k = \dim(C)$. Let $G \in \mathbb{F}^{n\times k}$ be a generator matrix for C. The decoder $\mathcal{D}$ for C is associated with a list of distributions $\{\mathcal{D}_i\}_{i\in[k]}$, where $\mathcal{D}_i$ is a distribution over the i-set of non-redundant edges E_i. On a word w and input $i \in [k]$ the decoder $\mathcal{D}$ picks a pair $(j_1, j_2) \in E_i$ according to the distribution $\mathcal{D}_i$ and recovers the ith message entry in the following way. If $j_1 \neq j_2$ then $c' \cdot G_{j_1} + c'' \cdot G_{j_2} = e_i$ for some $c', c'' \in \mathbb{F} \setminus \{0\}$, and the message bit is recovered by $c' \cdot w_{j_1} + c'' \cdot w_{j_2}$. Otherwise, $j_1 = j_2$ and then $c' \cdot G_{j_1} = e_i$ for some $c' \in \mathbb{F} \setminus \{0\}$, and the message bit is recovered by $c' \cdot w_{j_1}$.*

We are ready to prove Lemma 6.2.

Proof of Lemma 6.2. Let $i \in [k]$. Let E_i be an i-set of non-redundant edges as in Claim 6.7. Let $T_i = ([n], E_i)$ be an undirected graph, where $[n]$ is a set of nodes and E_i is a set of edges. Let $\mathcal{D}_i$ be a distribution over E_i as in Claim 6.7, i.e., the probability that the edge (j_1, j_2) is chosen is $\mathcal{D}_i(j_1, j_2)$.

Let $L \subseteq [n]$ be a maximal independent set in the graph T_i and let $\alpha_i > 0$ be s.t. $|L| = \alpha_i n$. Let $R = [n] \setminus L$ and note that $|R| = (1-\alpha_i)n$. Notice that (L, R) is a partition of $[n]$ and by definition there are no edges going from L to L. [6]

We argue that $1 - \alpha_i \geq \frac{\delta}{1-\frac{|\mathbb{F}|}{|\mathbb{F}|-1}\cdot\epsilon}$. We consider the following sampling. A set $R_0 \subseteq R$ is selected uniformly (independently) at random s.t. $|R_0| \leq \delta n$, and independently, the edge $(j_1, j_2) \in T_i$ is sampled according to $\mathcal{D}_i$. Let Ind be an indicator variable for the event $R_0 \cap (j_1, j_2) \neq \emptyset$. Then,

$$\mathbf{E}[Ind] = \mathbf{Pr}[Ind = 1] = \sum_{(j_1,j_2)\in E_i} \mathcal{D}_i(j_1, j_2) \cdot \mathbf{Pr}[j_1 \in R_0 \vee j_2 \in R_0] = \sum_{(j_1,j_2)\in E_i} \mathcal{D}_i(j_1, j_2) \cdot \frac{\delta n}{(1-\alpha_i)n} =$$

$$\frac{\delta}{1-\alpha_i} \cdot \sum_{(j_1,j_2)\in E_i} \mathcal{D}(j_1, j_2) = \frac{\delta}{1-\alpha_i}.$$

[6] Note that the graph might be not bipartite.

Let R_0 s.t. $|R_0| \leq \delta n$ be a subset which achieves (at least) this expectation, i.e.,

$$\sum_{(j_1,j_2)\in T_i} \mathcal{D}_i(j_1,j_2) \cdot \mathbf{Pr}[j_1 \in R_0 \vee j_2 \in R_0] \geq \frac{\delta}{1-\alpha_i}.$$

Change every symbol in R_0 independently to uniformly chosen random element of $\mathbb{F}$, i.e., every symbol from R_0 is independently and uniformly distributed in $\mathbb{F}$. Then, the probability that the decoder will not recover correctly the ith message symbol is at least $\frac{|\mathbb{F}|-1}{|\mathbb{F}|} \cdot \frac{\delta}{1-\alpha_i}$.[7] But the mistake of the decoder must be at most $1-(\frac{1}{|\mathbb{F}|}+\epsilon) = \frac{|\mathbb{F}|-1}{|\mathbb{F}|} - \epsilon$. Hence $\frac{|\mathbb{F}|-1}{|\mathbb{F}|} \cdot \frac{\delta}{1-\alpha_i} \leq \frac{|\mathbb{F}|-1}{|\mathbb{F}|} - \epsilon$ and $\frac{\delta}{1-\alpha_i} \leq 1 - \frac{|\mathbb{F}|}{|\mathbb{F}|-1} \cdot \epsilon$. We conclude that $1-\alpha_i \geq \frac{\delta}{1-\frac{|\mathbb{F}|}{|\mathbb{F}|-1}\cdot\epsilon}$.

Let $M_i \subseteq E_i$ be a maximal matching (self loops are allowed). We argue that $|M_i| \geq (1-\alpha_i)n/2$. The vertices left uncovered by M_i must be an independent set, since for an edge between any of these vertices would allow us to increase the size of the matching at least by one. Since the size of the maximal independent set is $\alpha_i n$ it follows that the number of vertices covered by M_i is at least $(1-\alpha_i)n$. Since every edge of M_i covers at most two vertices (self-loop covers only one vertex) we have $|M_i| \geq (1-\alpha_i)n/2$.

Thus $|M_i| \geq \frac{(1-\alpha_i)n}{2} \geq \frac{1}{2} \cdot \frac{\delta n}{1-\frac{|\mathbb{F}|}{|\mathbb{F}|-1}\cdot\epsilon}$ and recall that for every $(j_1,j_2) \in M_i$ we have $e_i \in \mathrm{span}\{G_{j_1}, G_{j_2}\}$. □

Acknowledgements

We would like to thank Madhu Sudan for valuable discussions about LTCs and LDCs.

References

[1] N. Alon, T. Kaufman, M. Krivelevich, S. Litsyn, and D. Ron, "Testing Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 4032–4039, 2005. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TIT.2005.856958

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof Verification and the Hardness of Approximation Problems," *Journal of the ACM*, vol. 45, no. 3, pp. 501–555, May 1998.

[3] S. Arora and S. Safra, "Probabilistic Checking of Proofs: A New Characterization of NP," *Journal of the ACM*, vol. 45, no. 1, pp. 70–122, Jan. 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*, IEEE, Ed. IEEE Computer Society Press, 2003, pp. 116–125.

[5] E. Ben-Sasson, "Limitation on the rate of families of locally testable codes," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 17, p. 123, 2010.

[7] Here we used an assumption that E_i is an i-set of non-redundant edges, since a decoder uses both endpoints of an edge to recover a message bit and a change in any of this endpoint affects its recovery output.

[6] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. P. Vadhan, "Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding," *SIAM Journal on Computing*, vol. 36, no. 4, pp. 889–974, 2006.

[7] E. Ben-Sasson, O. Goldreich, and M. Sudan, "Bounds on 2-Query Codeword Testing," in *RANDOM-APPROX*, ser. Lecture Notes in Computer Science, vol. 2764. Springer, 2003, pp. 216–227. [Online]. Available: http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2764&am

[8] E. Ben-Sasson, V. Guruswami, T. Kaufman, M. Sudan, and M. Viderman, "Locally Testable Codes Require Redundant Testers," in *IEEE Conference on Computational Complexity*. IEEE Computer Society, 2009, pp. 52–61. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CCC.2009.6

[9] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova, "Some 3CNF Properties Are Hard to Test," *SIAM Journal on Computing*, vol. 35, no. 1, pp. 1–21, 2005. [Online]. Available: http://epubs.siam.org/SICOMP/volume-35/art_44544.html

[10] E. Ben-Sasson and M. Sudan, "Simple PCPs with poly-log rate and query complexity," in *STOC*. ACM, 2005, pp. 266–275. [Online]. Available: http://doi.acm.org/10.1145/1060590.1060631

[11] ——, "Limits on the rate of locally testable affine-invariant codes," vol. 17, 2010, p. 108. [Online]. Available: http://www.eccc.uni-trier.de/report/2010/108/

[12] I. Dinur, "The PCP theorem by gap amplification," *Journal of the ACM*, vol. 54, no. 3, pp. 12:1–12:44, Jun. 2007.

[13] I. Dinur and E. Goldenberg, "Locally testing direct product in the low error range," in *FOCS*. IEEE Computer Society, 2008, pp. 613–622. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2008.26

[14] I. Dinur and O. Reingold, "Assignment Testers: Towards a Combinatorial Proof of the PCP Theorem," *SIAM Journal on Computing*, vol. 36, no. 4, pp. 975–1024, 2006. [Online]. Available: http://dx.doi.org/10.1137/S0097539705446962

[15] Z. Dvir, "On Matrix Rigidity and Locally Self-Correctable Codes," in *IEEE Conference on Computational Complexity*. IEEE Computer Society, 2010, pp. 291–298. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CCC.2010.35

[16] Z. Dvir and A. Shpilka, "Locally Decodable Codes with Two Queries and Polynomial Identity Testing for Depth 3 Circuits," *SIAM J. Comput*, vol. 36, no. 5, pp. 1404–1434, 2007. [Online]. Available: http://dx.doi.org/10.1137/05063605X

[17] R. G. Gallager, *Low-density Parity Check Codes*. MIT Press, 1963.

[18] ——, *Information Theory and Reliable Communication*. Wiley, New York, 1968.

[19] O. Goldreich, H. J. Karloff, L. J. Schulman, and L. Trevisan, "Lower bounds for linear locally decodable codes and private information retrieval," *Computational Complexity*, vol. 15, no. 3, pp. 263–296, 2006. [Online]. Available: http://dx.doi.org/10.1007/s00037-006-0216-3

[20] O. Goldreich and M. Sudan, "Locally testable codes and PCPs of almost-linear length," *Journal of the ACM*, vol. 53, no. 4, pp. 558–655, Jul. 2006.

[21] V. Guruswami, "On 2-Query Codeword Testing with Near-Perfect Completeness," in *ISAAC*, ser. Lecture Notes in Computer Science, vol. 4288. Springer, 2006, pp. 267–276. [Online]. Available: http://dx.doi.org/10.1007/11940128_28

[22] R. Impagliazzo, V. Kabanets, and A. Wigderson, "New direct-product testers and 2-query PCPs," in *STOC*, M. Mitzenmacher, Ed. ACM, 2009, pp. 131–140. [Online]. Available: http://doi.acm.org/10.1145/1536414.1536435

[23] T. Kaufman and M. Sudan, "Sparse Random Linear Codes are Locally Decodable and Testable," in *FOCS*. IEEE Computer Society, 2007, pp. 590–600. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/FOCS.2007.65

[24] ——, "Algebraic property testing: the role of invariance," in *STOC*. ACM, 2008, pp. 403–412. [Online]. Available: http://doi.acm.org/10.1145/1374376.1374434

[25] T. Kaufman and M. Viderman, "Locally Testable vs. Locally Decodable Codes," in *APPROX-RANDOM*, ser. Lecture Notes in Computer Science, M. J. Serna, R. Shaltiel, K. Jansen, and J. D. P. Rolim, Eds., vol. 6302. Springer, 2010, pp. 670–682. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15369-3

[26] T. Kaufman and A. Wigderson, "Symmetric LDPC Codes and Local Testing," in *ICS*, A. C.-C. Yao, Ed. Tsinghua University Press, 2010, pp. 406–421. [Online]. Available: http://conference.itcs.tsinghua.edu.cn/ICS2010/content/papers/32.html

[27] I. Kerenidis and R. de Wolf, "Exponential lower bound for 2-query locally decodable codes via a quantum argument," *J. Comput. Syst. Sci*, vol. 69, no. 3, pp. 395–420, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.jcss.2004.04.007

[28] G. Kol and R. Raz, "Bounds on 2-query locally testable codes with affine tests," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 16, p. 138, 2009.

[29] ——, "Locally testable codes analogues to the unique games conjecture do not exist," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 16, p. 128, 2009.

[30] O. Meir, "Combinatorial Construction of Locally Testable Codes," *SIAM J. Comput*, vol. 39, no. 2, pp. 491–544, 2009. [Online]. Available: http://dx.doi.org/10.1137/080729967

[31] K. Obata, "Optimal Lower Bounds for 2-Query Locally Decodable Linear Codes," in *RANDOM*, ser. Lecture Notes in Computer Science, J. D. P. Rolim and S. P. Vadhan, Eds., vol. 2483. Springer, 2002, pp. 39–50. [Online]. Available: http://link.springer.de/link/service/series/0558/bibs/2483/24830039.htm

[32] R. Raz, "A parallel repetition theorem," *SIAM J. Comput.*, vol. 27, no. 3, pp. 763–803, 1998. [Online]. Available: http://dx.doi.org/10.1137/S0097539795280895

[33] D. A. Spielman, "Computationally Efficient Error-Correcting Codes and Holographic Proofs," Massachusetts Institute of Technology, PhD thesis, 1995.

[34] M. Sudan, *Personal Comunication*, 2010.

[35] D. P. Woodruff, "New Lower Bounds for General Locally Decodable Codes," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 14, no. 006, 2007. [Online]. Available: http://eccc.hpi-web.de/eccc-reports/2007/TR07-006/index.html

[36] ——, "A Quadratic Lower Bound for Three-Query Linear Locally Decodable Codes over Any Field," in *APPROX-RANDOM*, ser. Lecture Notes in Computer Science, M. J. Serna, R. Shaltiel, K. Jansen, and J. D. P. Rolim, Eds., vol. 6302. Springer, 2010, pp. 766–779. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15369-3

A Proofs of folklore statements

This section contains two statements used earlier in the paper, the proofs of which we view as folklore. We present these results and their proofs for the sake of completeness.

A.1 Query reduction

The following theorem (its proof is folklore) stresses the importance of obtaining lower bounds on 3-query LTCs.

Theorem A.1 (Folklore)**.** *If there exists an asymptotically good family of LTCs then there exists an asymptotically good family of binary 3-query LTCs. Equivalently, if there is no asymptotically good family of 3-query LTCs then there is no asymptotically good family of LTCs.*

The proof of Theorem A.1 follows from the following folklore proposition, which appeared e.g. in [30, Theorem 6.11].

Proposition A.2 (Query Reduction)**.** *If $C \subseteq \mathbb{F}^n$ is a (q, ϵ, δ)-LTC and $k = \dim(C)$ then there exist constants $\alpha, m > 0$ (which depend only on q) and $C' \subseteq \mathbb{F}^{nm}$ s.t. C' is a $(3, \alpha\epsilon, \delta)$-LTC, $\mathrm{rate}(C') = \alpha \cdot \mathrm{rate}(C)$ and $\delta(C') \geq 0.99 \cdot \delta(C)$. Moreover, the code C' is obtained from C by appending additional symbols.*

Proposition A.2 implies that every LTC over the field of constant size can be converted to 3-query LTC over the same field (with only a constant factor loss in parameters). Hence we conclude Theorem A.1.

A.2 Transitive codes are regular

Claim A.3. *Let $C \subseteq \mathbb{F}_2^n$ be a code. If C is 1-transitive then C is q-regular for every $q > 0$.*

Proof. For $l \in [n]$ and $q > 0$ let $T_l^q = \left\{u \in C_q^\perp \mid l \in \mathrm{supp}(u)\right\}$. It is sufficient to argue that for every $i, j \in [n]$ and $q > 0$ we have $|T_i^q| = |T_j^q|$.

Assume the contrary, i.e., there exist $i, j \in [n]$ and $q > 0$ s.t. $|T_i^q| > |T_j^q|$. Let G be a 1-transitive group s.t. C is invariant under G. For $\pi \in G$ let $\pi(T_i^q) = \{\pi(u) \mid u \in T_i^q\}$. Note that for all $\pi \in G$ we have $|T_i^q| = |\pi(T_i^q)|$. Let $\pi \in G$ be s.t. $\pi(i) = j$ (such π exists since G is 1-transitive). It holds that $\pi(T_i^q) \subseteq T_j^q$ since for all $u \in \pi(T_i^q)$ we know that $j \in \mathrm{supp}(u)$ and $u \in C_q^\perp$. This implies that $|T_i^q| = |\pi(T_i^q)| \leq |T_j^q|$. Contradiction. □

ECCC ISSN 1433-8092

http://eccc.hpi-web.de