Electronic Colloquium on Computational Complexity, Report No. 172 (2012)

Correctness and Corruption of Locally Decodable Codes

Mahdi Cheraghchi
CMU
cheraghchi@cmu.edu

Anna Gál[*]
UT Austin
panni@cs.utexas.edu

Andrew Mills
Digital Proctor
amillsx@gmail.com

December 5, 2012

Abstract

Locally decodable codes (LDCs) are error correcting codes with the extra property that it is sufficient to read just a small number of positions of a possibly corrupted codeword in order to recover any one position of the input. To achieve this, it is necessary to use randomness in the decoding procedures. We refer to the probability of returning the correct answer as the *correctness* of the decoding algorithm.

Thus far, the study of LDCs has focused on the question of the tradeoff between their length and the query complexity of the decoders. Another natural question is what is the largest possible correctness, as a function of the amount of codeword corruption and the number of queries, regardless of the length of the codewords. Goldreich et al. (Computational Complexity 15(3), 2006) observed that for a given number of queries and fraction of errors, the correctness probability cannot be arbitrarily close to 1. However, the quantitative dependence between the largest possible correctness and the amount of corruption δ has not been established before.

We present several bounds on the largest possible correctness for LDCs, as a function of the amount of corruption tolerated and the number of queries used, regardless of the length of the code. Our bounds are close to tight. We also investigate the relationship between the amount of corruption tolerated by an LDC and its minimum distance as an error correcting code. Even though intuitively the two notions are expected to be related, we demonstrate that in general this is not the case. However, we show a close relationship between minimum distance and amount of corruption tolerated for linear codes over arbitrary finite fields, and for binary nonlinear codes. We use these results to strengthen the known bounds on the largest possible amount of corruption that can be tolerated by LDCs (with any nontrivial correctness better than random guessing) regardless of the query complexity or the length of the code.

[*] Supported in part by NSF Grant CCF-1018060

ISSN 1433-8092

1 Introduction

Locally decodable codes (LDCs) are error correcting codes with the extra property that it is sufficient to read just a small number of positions of a possibly corrupted codeword in order to recover any one position of the input. The concept of LDCs dates back to several papers in the 1990s (for example [2, 1, 17]), but the formal definition is from Katz and Trevisan [11]:

Definition 1. (Katz and Trevisan [11]) For reals δ and ϵ, and a natural number q, we say that $\mathbf{C}\colon \Sigma^n \to \Gamma^m$ is a (q, δ, ϵ)-*Locally Decodable Code (LDC)* if there exists a probabilistic algorithm A such that: in every invocation, for every $x \in \Sigma^n$ and $y \in \Gamma^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$ (where $d(,)$ denotes Hamming distance) and for every $i \in [n]$, A reads at most q positions of y and we have $\Pr\left[A^y(i) = x_i\right] \geq \frac{1}{|\Sigma|} + \epsilon$, where the probability is taken over the internal coin tosses of A. A is called the *Decoding Algorithm* or *Decoder*.

We will refer to the value $\frac{1}{|\Sigma|} + \epsilon$ in Definition 1 as the *correctness* associated with the given decoding algorithm A while ϵ can be thought of as the *advantage* over random guessing. More formally, we use the following definition.

Definition 2. Let A be an algorithm operating on a code $\mathbf{C}\colon \Sigma^n \to \Gamma^m$. The *correctness* of the algorithm A for amount of corruption δ is defined as

$$\zeta_\delta(A) \triangleq \min_{i\in[n]} \min_{x\in\Sigma^n} \left(\min_{y\in\Gamma^m \,:\, d(y,\mathbf{C}(x))\leq\delta m} \Pr\left[A^y(i) = x_i\right] \right)$$

where the probability is taken over the internal coin tosses of A.

Unless stated otherwise, we consider codes for which the input and output alphabets Σ and Γ are the same. From Definition 1, there are several parameters related to an LDC, namely, the *length* m, *alphabet size* $|\Sigma|$, *number of queries* q, fraction of *tolerable errors* δ, and *correctness* ζ_δ that is the best correctness achievable by any decoding algorithm limited to q queries (when up to a δ fraction of the positions are adversarially corrupted). These parameters are competing, and ideally, one aims for small m (relative to n), small q, small $|\Sigma|$ (in particular, the binary case $\Sigma = \{0, 1\}$), large δ, and large ζ_δ. The central question on LDCs is to characterize the achievable range of parameters.

So far, research on LDCs has been mostly focused on the possible trade-offs between length and the number of queries (for a given alphabet size, possibly depending on n). Namely, for a given alphabet size and number of queries q (e.g., constant or a slowly growing function of m), the question is to find out the minimum possible codeword length m for which there are LDCs with any nontrivial (constant) δ and nontrivial advantage ϵ.

For $q = 2$, the Hadamard code over a finite field F is easily seen to achieve correctness $\zeta_\delta \geq 1 - 2\delta$ (which is nontrivial for $\delta < \frac{1}{2}(1 - 1/|F|)$) at exponential length $m = |F|^n$. Conversely, it is known that any two query LDC must have exponential length [12, 18, 9, 10], for linear codes over arbitrary finite fields, and for non-linear codes over not too large alphabets.

For $q > 2$, the gap between known upper and lower bounds on the length of LDCs has remained significant. In this case, classical Reed-Muller codes (cf. [15]) can achieve lengths $m = \exp(n^{1/(q-1)})$ which is super-polynomial for any constant number of queries [14, 2, 7, 8]. Following the breakthrough work of Yekhanin [20], subexponential-sized LDCs (i.e., $m = \exp(n^{o(1)})$) for constant q (as small as $q = 3$) were discovered [20, 16, 5, 3, 4]. For large number of queries, namely, $q = N^\alpha$, the *multiplicity codes* of [13] are locally decodable at rates arbitrarily close to 1. Known lower bounds for $q > 2$ show that any q-query LDC must have length $m = \Omega(n^{q/(q-1)})$, which is far from the best known upper bounds (see [11] and slight improvements in [18, 19]). For a comprehensive survey of these results and the literature on locally decodable codes refer to [21, 22].

By a union bound, any LDC equipped with a decoder that does not err on an uncorrupted codeword and for which each individual query position is uniformly distributed achieves correctness $\zeta_\delta \geq 1 - q\delta$.

This simple observation is what the error analysis of various families of LDCs, including the Hadamard code and 3-query "matching vector" LDCs of [20, 16, 5] are based on. Gál and Mills [6] have shown that the correctness bound for 3-query matching vector codes is essentially optimal, in that any code that noticeably improves their correctness bound has to have exponential length. At exponential length, the binary Hadamard code already achieves a better correctness $1 - 2\delta$ (using only 2 queries) than matching vector codes.

In this paper, we study the tradeoffs between correctness ζ_δ, tolerable errors δ, and the number of queries q. We estimate the maximum possible correctness achievable by any q-query LDC at error rate δ. Goldreich et al. [9] observed that for a given number of queries and fraction of errors, the correctness probability cannot be arbitrarily close to 1. However, the quantitative dependence between the largest possible correctness and the amount of corruption δ has not been established before. We believe that this is a fundamental question about the limitations of locally decodable codes.

First, we consider binary (possibly non-linear) codes, and obtain the following upper bounds on the correctness probability of any non-adaptive decoder for binary codes. We remark that our upper bound on correctness also holds considering uniformly random messages x in Definition 2 instead of the minimum over x, which makes the result even stronger.

Theorem 3. *Let* $\mathbf{C}\colon \{0,1\}^n \to \{0,1\}^m$ *be a code, and let* A *be a* q *query non-adaptive decoding algorithm for it. Then, for large enough* n, [1]

$$\zeta_\delta(A) \le 1 - \frac{\delta}{2\sqrt{q}}\Big(4\delta(1-\delta)\Big)^{q/4} + O_q(\frac{1}{n^{1/3}}).$$

In order to obtain a lower bound on the largest possible correctness of q-query LDCs, we look at the binary Hadamard code, which is a prototypical example of an LDC. By simply repeating the classical 2-query decoder for this code and taking the majority of the results (Lemma 16), we obtain a q-query decoder with correctness at least $1-\Big(4(2\delta)(1-2\delta)\Big)^{q/4}$ for the Hadamard code. This shows that the upper bound of Theorem 3 is close to tight, in the sense that it gives the correct exponent in the dependence on the number of queries. In fact, we prove the following more precise bound: $1 - 2^{q/2-1}(2\delta)^{\lceil\frac{q}{4}\rceil}(1-2\delta)^{\lfloor\frac{q}{4}\rfloor}$. For $q = 2$ this gives the $1 - 2\delta$ bound, which we show to be asymptotically tight.

Next, we derive specific upper bounds on the correctness of two-query binary LDCs. The reason to separately focus on this special case is its fundamental importance, which is exemplified by the Hadamard code that is used as an important building block in constructions of binary LDCs (typically as an inner code in a concatenation scheme, cf. [4]) or PCP constructions. It is natural to ask if the correctness $1-2\delta$ achieved by the Hadamard code may be improved by other codes. We prove that this is not the case and any 2-query binary LDC, no matter how long, is unable to substantially improve the correctness bound achieved by the Hadamard code.

We will then move on to the connection between the minimum distance of LDCs and the fraction δ of errors tolerable by their local decoders. The minimum distance is a fundamental classical notion related to error-correcting codes which captures the fraction of adversarial errors that are *combinatorially* correctable by the code (regardless of any locality or efficiency concerns or the use of randomness). On the other hand, the parameter δ associated with LDCs in Definition 1 captures the fraction of adversarial errors that are tolerable for locally decoding any single message symbol, where "decoding" refers to obtaining a non-trivial guess for the correct symbol. While both notions intuitively capture error tolerance of the code and are therefore expected to be related, their exact relationship is not obvious from the standard definition of LDCs given by Definition 1. In this work, we study this relationship by showing bounds on the minimum distance of LDCs with a given error tolerance δ. For arbitrary binary LDCs of codeword length m, we verify the intuition that the minimum distance of the code is at least $2\delta m$. For linear LDCs of codeword length m

[1] We use the notation $O_q()$ for $O()$ with the hidden constant depending on q.

over a finite field F, we extend this bound to show that the minimum distance is at least $|F|\delta m/(|F|-1)$. For non-binary non-linear LDCs, we show that in general there is no relationship between the minimum distance and error tolerance δ. However, we prove that any LDC has a large sub-code having minimum distance at least δm.

The fact that the minimum distance of LDCs is not directly related to the error tolerance parameter is mainly because the standard definition of LDCs (Definition 1) is very weak for LDCs over large alphabets. It is noted by Goldreich et al. [9], that the correct answer may not be the value that is being output with the largest probability, and thus the definition does not allow amplification of the decoder's correctness probability by independent repetitions, unless the alphabet size is 2.

To circumvent this issue, we consider the following stronger definition of LDCs:

Definition 4. For reals δ and ϵ, and a natural number q, we say that $\mathbf{C}\colon \Sigma^n \to \Gamma^m$ is a *strong* (q, δ, ϵ)*-Locally Decodable Code (strong LDC)* if there exists a probabilistic algorithm A such that: in every invocation, for every $x \in \Sigma^n$ and $y \in \Gamma^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$ and for every $i \in [n]$, A reads at most q positions of y and for every $x' \in \Sigma \setminus x_i$ we have $\Pr[A^y(i) = x_i] \geq \Pr[A^y(i) = x'] + \epsilon$, where the probability is taken over the internal coin tosses of A.

It is easy to see from the definitions that a strong (q, δ, ϵ)-LDC is a $(q, \delta, \epsilon(1 - 1/|\Sigma|))$-LDC and thus, Definition 4 is indeed stronger than Definition 1. Note that the two definitions are equivalent for binary codes, up to constant factor difference in ϵ. Moreover, the strong definition is chosen to allow amplification of the correctness probability by independent repetitions. This property makes it possible to show that for strong LDCs over any alphabet (with $\epsilon > 0$), the minimum distance is at least $2\delta m$.

Finally, we combine the above-mentioned results with the classical Plotkin bound on codes (cf. [15]) to obtain an upper bound for the maximum δ tolerable by any (standard or strong) LDC in terms of the alphabet size. In particular, for standard LDCs (Definition 1), we show that any binary LDC must satisfy $\delta \leq 1/4 + o(1)$ and moreover conclude that any linear LDC over field F must satisfy $\delta \leq (1 - 1/|F|)^2 + o(1)$. For the special cases of binary codes and non-binary linear codes, this improves the upper bound $\delta \leq 1 - 1/|F|$ that is known to hold for any LDC [6]. For strong LDCs (Definition 4), however, we show $\delta \leq \frac{1}{2}(1 - 1/|F|)$ regardless of linearity, which gives a stronger bound when $|F| > 2$.

Techniques:

In order to prove Theorem 3, we define a measure on codes with respect to a given noise distribution that we call the *statistical influence* of the message variables (Definition 8). The statistical influence of a given variable measures the dependence of the distribution of local views of random (and possibly corrupted) codewords on the value of the variable. It is defined as the statistical distance between the distribution of the corrupted codewords restricted to a local view and conditioned on different values of the variable. Intuitively, if a variable has small statistical influence on a local view, then it is unlikely that any decoding algorithm can correctly recover its value from the given local view. Formally, we show that upper bounds on statistical influence, (averaged over all local views) translate into upper bounds on the correctness probability of the LDC.

We estimate statistical influence by relating it to an expression that only depends on the Hamming distance between the local views of pairs in a matching between codewords corresponding to messages with a 0 and those having a 1 at the given variable (Claim 11). We show the existence of a matching, using the probabilistic method, for which the Hamming distances are sufficiently small on average.

We remark that while it seems tempting to guess the bound proved in Theorem 3 by natural heuristics (such as estimating the number of times a typical decoder hits corrupted positions) which result in qualitatively sound estimates, it is not clear how to turn such simpler intuitions into correct proofs. This is partly because a local decoder may behave in arbitrary ways in choosing the query positions. In particular, it cannot be assumed that the query positions are chosen uniformly at random, since reductions involving such assumptions change the parameters of the code, including its correctness. The proof should also take into account the possibility of the decoder being correct even after reading corrupted positions.

1.1 Notation

Let F be an arbitrary finite field. We denote by F^* the multiplicative group of the non-zero elements of F. Arithmetic operations involving field elements are over F. This should be clear from the context, and will be omitted from the notation.

For two strings $x, y \in F^m$, we use $d(x,y)$ to denote the Hamming distance between x and y. Similarly, the Hamming distance between x and a code $\mathbf{C}$ is denoted by $d(x,\mathbf{C})$. For a code $\mathbf{C}$: $F^n \to F^m$ (that we identify by its encoding function throughout the work), we can represent any vector $y \in F^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$ as a sum of the form $y = \mathbf{C}(x) + B$, where $B \in F^m$, such that the number of nonzero entries in B is at most δm.

We use $(\mathbf{C}(x)+B)_Q \in F^{|Q|}$ to denote the codeword $\mathbf{C}(x)$ corrupted by B restricted to the positions indexed by the query set Q. Similarly, for any string $z \in F^m$ we denote by z_Q the restriction of z to the positions in Q.

We use the notation $\Pr_{x,B,A}$ to indicate probabilities over uniformly random input x from F^n, B chosen at random from a given distribution for corruption, and the random coin tosses of the given algorithm A.

We use $E; F$ to denote the intersection of the events E and F. $H()$ denotes the binary entropy function, i.e., $H(x) \triangleq -x \log_2 x - (1-x)\log_2(1-x)$.

The correlation between two Boolean functions f and g is defined as $Corr(f,g) \triangleq \Pr_x[f(x) = g(x)] - \Pr_x[f(x) \neq g(x)]$.

2 Preliminaries

We will use the following theorem of Katz and Trevisan [11].

Theorem 5. *(Theorem 2 in [11]) Let $g : \{0,1\}^n \to R$ be a function. Assume there is an algorithm A such that for every $i \in [n]$, we have $\Pr_x[A(g(x), i) = x_i] \geq \frac{1}{2} + \epsilon$, where the probability is taken over the internal coin tosses of A and uniform $x \in \{0,1\}^n$. Then $\log|R| \geq (1 - H(1/2+\epsilon))n$.*

The following property of decoders with respect to distributions of corruption that contain a truly random part was proved in [6]:

Lemma 6. *[6] Let $\mathbf{C}$: $F^n \to F^m$ be a code. Assume there exists a q query algorithm A such that $\Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(i) = x_i\right] \geq \frac{1}{|F|} + \epsilon$ where the probability is over the internal coin tosses of A, uniform $x \in F^n$, and $B = B_1 + B_2$ chosen by the product distribution of the distributions D_R and D_S, where R and S are disjoint subsets of $[m]$, D_R is arbitrary over vectors in F^m that are identically zero in coordinates outside of R, D_S is uniformly random when restricted to S, and identically zero in coordinates outside of S. Then there exists a q query algorithm $\tilde{A}$ such that $\Pr_{x,B,\tilde{A}}\left[\tilde{A}^{\mathbf{C}(x)+B}(i) = x_i\right] \geq \frac{1}{|F|} + \epsilon$ as well, and $\tilde{A}$ never queries any positions from S.*

The following lemma is implicit in [6], and it holds for any distribution B used by an adversary for corrupting the codewords. We note that the lemma would be straightforward if the event E was independent of all other events in the statement. However, while we require that E does not depend on the internal randomness of A, it may depend on the distribution B and on the input x. Therefore, the events we work with are in general not independent events. Nevertheless, the lemma holds.

Lemma 7. *(implicit in [6]) Let $\mathbf{C}$ be a code $\Sigma^n \to \Sigma^m$ and A be a non-adaptive q-query decoder for $\mathbf{C}$. Let E be an event that does not depend on the internal randomness of A. Then, for any $i \in [n]$, $Q \subset [m]$*

with $|Q| = q$, $v \in \Sigma$, and any bit string s (representing the answers to the queries A makes)

$$\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) = v \;|E; Q; (\mathbf{C}(x) + B)_Q = s] = \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) = v \;|Q; (\mathbf{C}(x) + B)_Q = s]$$

where Q denotes the event "A queries Q".

For completeness, we include a proof of the lemma in the Appendix.

3 The statistical influence of the message variables of codes

In this section we prove a general result about estimating the correctness of local decoding algorithms, for arbitrary codes. Our bounds are given in terms of a measure on codes that we call *statistical influence.*

Definition 8. Let $\mathbf{C}\colon \{0,1\}^n \to \{0,1\}^m$ be a code, and let B be randomly distributed on $\{0,1\}^m$. For $i \in [n]$, and $Q \subseteq [m]$, we define

$$\Delta_{i,Q} \triangleq \sum_{s \in \{0,1\}^q} \left| \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_i = 1] - \Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_i = 0] \right| ,$$

and call it the *statistical influence* of the i-th message variable of C on query set Q with respect to the distribution B.

Note that in other words, $\Delta_{i,Q}$ is the statistical distance between the distribution of the corrupted codewords restricted to Q and conditioned on $x_i = 1$ vs. $x_i = 0$. We assume uniform distribution over the input x, but the definition can be generalized to arbitrary probability distributions over x as well. We omit from the notation the dependence on the distribution B, in what follows, the distribution B will be clear from the context.

Intuitively, if the statistical influence of the i-th variable of a code is small on a query set Q with respect to a distribution B, then any decoder will have a large probability of error using the query set Q in trying to recover x_i, if the adversary uses the distribution B to corrupt the codewords. We prove this more formally in the following theorem.

Theorem 9. *Let $\mathbf{C}\colon \{0,1\}^n \to \{0,1\}^m$ be a code, and let A be a non-adaptive decoding algorithm for it. Let B be a distribution on $\{0,1\}^m$ that is independent of the input x. Let $i \in [n]$. Then*

$$\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) \neq x_i] \geq \frac{1}{2} \sum_{Q} (1 - \frac{1}{2}\Delta_{i,Q}) \Pr[A \textit{ queries } Q] .$$

Proof. Since the distribution B is independent of x, for all query sets Q of size q, $k \in \{0,1\}$, and $s \in \{0,1\}^q$, $\Pr_{x,B}[(\mathbf{C}(x) + B)_Q = s \mid x_1 = k] > 0$. Similarly, since A is non-adaptive, A is independent of x, and for Q that A queries with positive probability and $k \in \{0,1\}$, $\Pr_{x,A}[A \text{ queries } Q; x_1 = k] > 0$. Since all these probabilities are positive, we will be able to condition on these events properly.

Define $Err_{Q,k} \triangleq \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq k \mid A \text{ queries } Q; x_1 = k]$. Then

$$\begin{aligned}\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1] &= \sum_{Q} (Err_{Q,0} \Pr_x[x_1 = 0] + Err_{Q,1} \Pr_x[x_1 = 1]) \cdot \Pr_A[A \text{ queries } Q] \\ &= \frac{1}{2} \sum_{Q} (Err_{Q,0} + Err_{Q,1}) \Pr_A[A \text{ queries } Q] .\end{aligned}$$

Next we consider $Err_{Q,0} + Err_{Q,1}$

$$= \sum_{s\in\{0,1\}^q} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; x_1 = 0; (\mathbf{C}(x)+B)_Q = s]\cdot$$
$$\Pr_{x,B,A}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0]$$
$$+ \sum_{s\in\{0,1\}^q} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; x_1 = 1; (\mathbf{C}(x)+B)_Q = s]\cdot$$
$$\Pr_{x,B,A}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 1] \ .$$

The value of x_1 does not depend on the internal randomness of A. Therefore, by Lemma 7, we can remove the conditioning on the value of x_1 in the first terms of the above products. Using the notation

$$p_s^Q \triangleq \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) = 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = s] \ ,$$

this gives

$$Err_{Q,0} + Err_{Q,1} = \sum_{s\in\{0,1\}^q} (1-p_s^Q) \Pr_{x,B,A}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 0]$$
$$+ \sum_{s\in\{0,1\}^q} p_s^Q \Pr_{x,B,A}[(\mathbf{C}(x)+B)_Q = s \mid A \text{ queries } Q; x_1 = 1] \ .$$

Since A is non-adaptive, the internal randomness of A is independent of B, and the values of the positions labeled by Q are independent of whether the algorithm actually queries Q. So we have

$$\begin{aligned}
&Err_{Q,0} + Err_{Q,1}\\
&= \sum_{s\in\{0,1\}^q} (1-p_s^Q) \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] + \sum_{s\in\{0,1\}^q} p_s^Q \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\\
&= \sum_{s\in\{0,1\}^q} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\\
&\qquad + \sum_{s\in\{0,1\}^q} p_s^Q \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big)\\
&= 1 + \sum_{s\in\{0,1\}^q} p_s^Q \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\Big) \ .
\end{aligned}$$

This expression is smallest when p_s^Q is 0 for s such that

$$\Pr[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] > \Pr[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]$$

and when p_s^Q is 1 for s such that

$$\Pr[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] < \Pr[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] \ .$$

Thus

$$Err_{Q,0} + Err_{Q,1} \geq 1 - \sum_{\substack{s\in\{0,1\}^q\\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \Big(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\Big) \ .$$

Next, note that

$$\sum_{s\in\{0,1\}^q} \left(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\right) = 1 - 1 = 0$$

This implies

$$\sum_{\substack{s\in\{0,1\}^q \\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \left(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\right) =$$

$$\sum_{\substack{s\in\{0,1\}^q \\ \Pr[s|x_1=1]>\Pr[s|x_1=0]}} \left(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\right)$$

Therefore,

$$\begin{aligned}
&\sum_{\substack{s\in\{0,1\}^q \\ \Pr[s|x_1=1]<\Pr[s|x_1=0]}} \left(\Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1]\right) \\
&= \frac{1}{2} \sum_{s\in\{0,1\}^q} \left| \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\right| ,
\end{aligned}$$

and this concludes the proof of the theorem. □

By Theorem 9, we can estimate the largest possible correctness of any decoder of a code by estimating the statistical influence of the message variables of the code. We use the following equivalent expression of statistical influence, for obtaining our estimates. The next claims hold for any input position i. To simplify notation, we state them for $i = 1$, and use $1w$ ($0w$) to denote the input string with $x_1 = 1$ ($x_1 = 0$) followed by the string $w \in \{0,1\}^{n-1}$.

Claim 10.

$$\Delta_{1,Q} = \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \left| \sum_{w\in\{0,1\}^{n-1}} \left(\Pr_B[B_Q = s - \mathbf{C}(1w)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\right)\right| .$$

Proof.

$$\begin{aligned}
\Delta_{1,Q} &\triangleq \sum_{s\in\{0,1\}^q} \left| \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 1] - \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x_1 = 0]\right| \\
&= \sum_{s\in\{0,1\}^q} \Bigg| \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 1w] \Pr_x[x_2x_3...x_n = w \mid x_1 = 1] \\
&\quad - \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 0w] \Pr_x[x_2x_3...x_n = w \mid x_1 = 0]\Bigg| \\
&= \sum_{s\in\{0,1\}^q} \Bigg| \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 1w] \Pr_x[x_2x_3...x_n = w] - \\
&\quad \sum_{w\in\{0,1\}^{n-1}} \Pr_{x,B}[(\mathbf{C}(x)+B)_Q = s \mid x = 0w] \Pr_x[x_2x_3...x_n = w]\Bigg| \\
&= \sum_{s\in\{0,1\}^q} \left| \sum_{w\in\{0,1\}^{n-1}} \Pr_x[x_2x_3...x_n = w] \left(\Pr_B[B_Q = s - \mathbf{C}(1w)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\right)\right| \\
&= \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \left| \sum_{w\in\{0,1\}^{n-1}} \left(\Pr_B[B_Q = s - \mathbf{C}(1w)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w)_Q]\right)\right| .
\end{aligned}$$

This concludes the proof of the claim. □

Claim 11. *Let M be any matching*[2] *between the set of vectors in $\{0,1\}^n$ with first bit 0 and the set of vectors in $\{0,1\}^n$ with first bit 1. Then*

$$\Delta_{1,Q} \leq \frac{1}{2^{n-1}} \sum_{(w_1,w_2)\in M} \Delta_{w_1,w_2} \ ,$$

where

$$\Delta_{w_1,w_2} \triangleq \sum_{s\in\{0,1\}^q} \left| \Pr_B[B_Q = s] - \Pr_B[B_Q = s + \mathbf{C}(1w_1)_Q - \mathbf{C}(0w_2)_Q] \right| .$$

Proof.

$$\begin{aligned}\Delta_{1,Q} &= \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \left| \sum_{(w_1,w_2)\in M} \Big(\Pr_B[B_Q = s - \mathbf{C}(1w_1)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w_2)_Q] \Big) \right| \\ &\leq \frac{1}{2^{n-1}} \sum_{s\in\{0,1\}^q} \sum_{(w_1,w_2)\in M} \left| \Pr_B[B_Q = s - \mathbf{C}(1w_1)_Q] - \Pr_B[B_Q = s - \mathbf{C}(0w_2)_Q] \right|\end{aligned}$$

by reordering the terms inside the absolute value of the expression in Claim 10 and using the triangle inequality. The claim follows by switching the order of summations and renaming $s - \mathbf{C}(1w_1)_Q$ as s. □

To obtain our results, we choose an appropriate distribution B for the corruption, and use a carefully chosen matching to estimate Δ_{w_1,w_2} with respect to B.

4 Correctness versus corruption and query complexity

4.1 Proof of the main theorem

In this section we prove Theorem 3. The statement will follow from the following more precise estimate, by substituting $t = 1/n^{1/3}$ (which in turn makes $\nu = O(1/n^{1/3})$ by the Taylor expansion of the entropy function $H(x)$).

Theorem 12. *Let $\mathbf{C}\colon \{0,1\}^n \to \{0,1\}^m$ be a code, and let A be a q query non-adaptive decoder for it. Fix $t < 1$, and let $\nu \triangleq \frac{1}{.99n(1-H(\frac{1}{2}+\frac{t}{2}))}$. Then, for large enough n,*

$$\zeta_\delta(A) \leq 1 - \frac{\delta}{2\sqrt{q(1+t^2)}} \Big(4\delta(1-\delta)\Big)^{\frac{q}{4}(1+t^2)} + \frac{2^{q+1}q^2}{n} + (q+1)\nu$$

Proof. For $i \in [n]$, define:

$$R_i \triangleq \Big\{ j \in [m] \ \Big| \ |Corr(x_i, \mathbf{C}(x)_j)| > t \Big\}$$

Now consider:

$$S \triangleq \Big\{ i \in [n] \ \Big| \ |R_i| \geq \nu m \Big\}$$

Clearly $|S|\nu m \leq \sum_{i\in[n]} |R_i|$. So there exists a $j \in [m]$ belonging to at least $\nu|S|$ of the sets R_i. Theorem 5 then implies that $\nu|S| \leq \frac{1}{1-H(\frac{1}{2}+\frac{t}{2})}$. Therefore, $|S| \leq \frac{1}{\nu}\frac{1}{1-H(\frac{1}{2}+\frac{t}{2})} = .99n < n$. So $\bar{S}$ contains at least one i. Without loss of generality, $1 \in \bar{S}$. That is, $|R_1| < \nu m$.

[2] Throughout this work by *matching* we mean perfect matching.

Because the argument below works for arbitrary algorithms, without loss of generality, we can assume the algorithm always queries exactly q positions. If the algorithm ever queried fewer than q positions, have it query more and ignore the additional values obtained.

Define $\gamma \triangleq \frac{|[m]\setminus R_1|}{m}$ and $\beta \triangleq \frac{\delta-\nu}{\gamma}$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset [m] \setminus R_1$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1$, and the internal randomness of A. (We emphasize that the corruption B_1 always has the same size; but, for B_2, it is chosen whether to include each member of R_1 independently). Let $B \triangleq B_1 \cup B_2$, generated by the product distribution of B_1 and B_2. We also use B to denote the characteristic vector of the set B. By our choice of the parameters, $|B| \leq \delta m$ always holds.

By Lemma 6, we can, without loss of generality, assume that A never queries any positions from R_1. Thus, by Theorem 9, in order to prove Theorem 12 it is sufficient to prove an upper bound on $\Delta_{1,Q}$ for every query set Q of size q, such that $Q \cap R_1 = \emptyset$.

To obtain bounds on $\Delta_{1,Q}$, we will use the following claim.

Claim 13. *Let $\beta < 1/2$ and B be the distribution that chooses subsets of size βm uniformly at random from $[m]$. Then, for any $z \in \{0,1\}^q$ of Hamming weight a,*

$$\begin{aligned}\Delta_z &\triangleq \sum_{s\in\{0,1\}^q} \left|\Pr_B[B_Q = s] - \Pr_B[B_Q = s+z]\right| \\ &\leq 2 - 4\binom{a}{\lfloor (a-1)/2 \rfloor}\beta^{\lceil (a+1)/2\rceil}(1-\beta)^{\lfloor (a-1)/2\rfloor} + \frac{3 * 2^q q^2}{m} \\ &\leq 2 - 2\frac{\beta}{\sqrt{2a}}(4\beta(1-\beta))^{a/2} + \frac{3 * 2^q q^2}{m}\end{aligned}$$

Proof. For a given s, $\Pr_B[B_Q = s]$ is a function of only the Hamming weight of s, namely, it is given by the expression

$$\frac{\binom{m-q}{\beta m-\ell}}{\binom{m}{\beta m}},$$

where ℓ is the Hamming weight of s. Recall that a is the Hamming weight of z. Also, let b be the number of positions where z and s both equal 1; and let c be number of positions where z equals 0 but s equals 1. Then, by Claims 24 and 25 (for large enough m), we have

$$\begin{aligned}\Delta_z &= \sum_{b=0}^{a}\binom{a}{b}\sum_{c=0}^{q-a}\binom{q-a}{c}\left|\frac{\binom{m-q}{\beta m-(b+c)}}{\binom{m}{\beta m}} - \frac{\binom{m-q}{\beta m-(a+c-b)}}{\binom{m}{\beta m}}\right| \\ &\leq \sum_{b=0}^{a}\binom{a}{b}\sum_{c=0}^{q-a}\binom{q-a}{c}\left(\left|\beta^{b+c}(1-\beta)^{q-b-c} - \beta^{a-b+c}(1-\beta)^{q-a+b-c}\right| + \frac{3q^2}{m}\right)\end{aligned}$$

Simplifying, we have

$$\begin{aligned}&= \sum_{b=0}^{a}\binom{a}{b}\sum_{c=0}^{q-a}\binom{q-a}{c}\left|\beta^{b+c}(1-\beta)^{q-b-c} - \beta^{a-b+c}(1-\beta)^{q-a+b-c}\right| + \frac{3 * 2^q q^2}{m} \\ &= \sum_{b=0}^{a}\binom{a}{b}\left(\sum_{c=0}^{q-a}\binom{q-a}{c}\beta^{c}(1-\beta)^{q-a-c}\right)\left|\beta^{b}(1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^{b}\right| + \frac{3 * 2^q q^2}{m} \\ &= \sum_{b=0}^{a}\binom{a}{b}\left|\beta^{b}(1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^{b}\right| + \frac{3 * 2^q q^2}{m}\end{aligned}$$

Since $\beta < 1/2$, we have $\beta < 1-\beta$. Therefore, the expression inside of the absolute value is positive if and only if $b < \frac{a}{2}$. Also note that the value of the expression inside of the absolute value, for a given b, has the opposite sign of that same expression when b is replaced by $a-b$. Using these facts, we have

$$\Delta_z \leq 2 \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} \Big(\beta^b (1-\beta)^{a-b} - \beta^{a-b}(1-\beta)^b \Big) + \frac{3 * 2^q q^2}{m}$$

Because $\sum_{b=0}^{a} \binom{a}{b} \beta^b (1-\beta)^{a-b} = (1-\beta+\beta)^a = 1$, the last line equals

$$\begin{aligned}
&= 2\Big(1 - \sum_{b=\lfloor (a+1)/2 \rfloor}^{a} \binom{a}{b} \beta^b (1-\beta)^{a-b} - \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} \beta^{a-b} (1-\beta)^b \Big) + \frac{3 * 2^q q^2}{m} \\
&\leq 2\Big(1 - 2 \sum_{b=0}^{\lfloor (a-1)/2 \rfloor} \binom{a}{b} \beta^{a-b} (1-\beta)^b \Big) + \frac{3 * 2^q q^2}{m} \\
&< 2 - 4 \binom{a}{\lfloor (a-1)/2 \rfloor} \beta^{\lceil (a+1)/2 \rceil} (1-\beta)^{\lfloor (a-1)/2 \rfloor} + \frac{3 * 2^q q^2}{m}
\end{aligned}$$

Here we obtained the first inequality by interchanging b and $a-b$ in second term of the previous line. The last line follows by omitting all but the $b = \lfloor (a-1)/2 \rfloor$ term from the sum. This proves the first inequality of the claim.

For proving the second inequality of the claim, we first observe that

$$\beta^{\lceil (a+1)/2 \rceil} (1-\beta)^{\lfloor (a-1)/2 \rfloor} \geq \beta \big(\beta(1-\beta)\big)^{a/2}.$$

To see this, consider separately the cases when a is even and when a is odd. Next, note that for any integer $k \geq 1$, $\binom{2k}{k} \geq \frac{4^k}{2\sqrt{k}}$. This follows from Stirling's formula for $k \geq 2$ and can be directly verified for $k = 1$. Using this, and considering the cases when a is even and when a is odd, we see that

$$\binom{a}{\lfloor (a-1)/2 \rfloor} \geq \frac{2^a}{2\sqrt{2a}} = \frac{4^{a/2}}{2\sqrt{2a}} \ ,$$

and thus we have

$$4 \binom{a}{\lfloor (a-1)/2 \rfloor} \beta^{\lceil (a+1)/2 \rceil} (1-\beta)^{\lfloor (a-1)/2 \rfloor} \geq 4\beta \frac{1}{2\sqrt{2a}} \big(4\beta(1-\beta)\big)^{a/2}.$$

This concludes the proof of the claim. □

The next two lemmas allow us to find an appropriate matching M that gives a good bound on $\Delta_{1,Q}$ when applying Claim 11 with respect to the matching M.

Lemma 14. *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a code, and let* $Q \subseteq [m]$*, such that* $|Q| = q$*. Assume that for every position* $j \in Q$*,* $|Corr(x_1, \mathbf{C}(x)_j)| \leq t$*. Then*

$$\mathop{\mathbb{E}}_{w_1, w_2 \in \{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \leq \frac{q}{2}(1+t^2)$$

Proof. Let $f(x) = \mathbf{C}(x)_j$, and let $g : \{1,-1\}^n \to \{1,-1\}$ be obtained from f by replacing 0s by 1s and 1s by -1s, that is for $y \in \{1,-1\}^n$, $g(y) = (-1)^{f((1-y)/2)}$. Then, the correlation between x_1 and $\mathbf{C}(x)_j$,

$$\begin{aligned} Corr(x_1, \mathbf{C}(x)_j) &= \Pr_{x\in\{0,1\}^n}[x_1 = \mathbf{C}(x)_j] - \Pr_{x\in\{0,1\}^n}[x_1 \neq \mathbf{C}(x)_j] \\ &= \Pr_{y\in\{1,-1\}^n}[y_1 = g(y)] - \Pr_{y\in\{1,-1\}^n}[y_1 \neq g(y)] \\ &= \frac{1}{2^n}\sum_{y\in\{1,-1\}^n} y_1 g(y) \\ &= \frac{1}{2}(S_1 - S_{-1}) \end{aligned}$$

where $S_1 = \frac{1}{2^{n-1}}\sum_{y\in\{1,-1\}^{n-1}} g(1y)$ and $S_{-1} = \frac{1}{2^{n-1}}\sum_{y\in\{1,-1\}^{n-1}} g(-1y)$.

We will estimate

$$\begin{aligned} &\mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \\ &\quad= \mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} \sum_{j\in Q} d\Big(\mathbf{C}(1w_1)_j, \mathbf{C}(0w_2)_j\Big) \\ &\quad= \sum_{j\in Q} \mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_j, \mathbf{C}(0w_2)_j\Big) \end{aligned}$$

Let $E_j \triangleq \mathbb{E}_{w_1,w_2\in\{0,1\}^{n-1}}\, d\Big(\mathbf{C}(1w_1)_j, \mathbf{C}(0w_2)_j\Big)$. Then,

$$E_j = \mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} \frac{1}{2}\big(1 - g(1w_1)g(-1w_2)\big)$$

and $\mathbb{E}_{w_1,w_2\in\{0,1\}^{n-1}}\, g(1w_1)g(-1w_2) = S_1 S_{-1}$. Thus, we have $E_j = \frac{1}{2}(1 - S_1 S_{-1})$.

Recall that by assumption, $|Corr(x_1, \mathbf{C}(x)_j)| \leq t$, thus $|(S_1 - S_{-1})| \leq 2t$. If S_1 and S_{-1} have the same sign, then $S_1 S_{-1} \geq 0$. Otherwise, if they have opposite signs, $|S_1 - S_{-1}| = |S_1| + |S_{-1}|$, and thus $|S_1| + |S_{-1}| \leq 2t$. Since $4ab \leq (a+b)^2$, substituting $a = |S_1|$ and $b = |S_{-1}|$, this implies that $S_1 S_{-1} \geq -t^2$, and in turn, $E_j \leq \frac{1}{2}(1 + t^2)$.

This concludes the proof of the lemma. □

Lemma 15. *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a code, and let* $Q \subseteq [m]$*, such that* $|Q| = q$*. Assume that for every position* $j \in Q$*,* $|Corr(x_1, \mathbf{C}(x)_j)| \leq t$*. Then there is a matching* M *between the set of vectors in* $\{0,1\}^n$ *with first bit* 0 *and the set of vectors in* $\{0,1\}^n$ *with first bit* 1 *such that*

$$\mathop{\mathbb{E}}_{(w_1,w_2)\in M} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \leq \frac{q}{2}(1 + t^2)$$

Proof. Consider the following family of matchings, parameterized by $u \in \{0,1\}^{n-1}$

$$M_u \triangleq \Big\{(v, v+u) \mid v \in \{0,1\}^{n-1}\Big\}$$

The key property we use about this family is that it is a partition of the set $\{(w_1, w_2) \mid w_1, w_2 \in \{0,1\}^{n-1}\}$. If we consider a probability distribution in which u is drawn uniformly at random from $\{0,1\}^{n-1}$ and then (w_1, w_2) is drawn uniformly at random from M_u, then

$$\mathop{\mathbb{E}}_{u\in\{0,1\}^{n-1}} \mathop{\mathbb{E}}_{(w_1,w_2)\in M_u} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) = \mathop{\mathbb{E}}_{w_1,w_2\in\{0,1\}^{n-1}} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \leq \frac{q}{2}(1 + t^2) \ ,$$

where the last inequality holds by Lemma 14. Therefore, for at least one u,

$$\mathop{\mathbb{E}}_{(w_1,w_2)\in M_u} d\Big(\mathbf{C}(1w_1)_Q, \mathbf{C}(0w_2)_Q\Big) \leq \frac{q}{2}(1+t^2) \ .$$

□

Now we are ready to finish the proof of Theorem 12. Let M be a matching with the properties in the above lemma, whose existence we proved. For $(w_1, w_2) \in M$, let $a(w_1, w_2)$ denote the Hamming weight of $\mathbf{C}(1w_1)_Q - \mathbf{C}(0w_2)_Q$. By Lemma 15

$$\frac{1}{2^{n-1}} \sum_{(w_1,w_2)\in M} a(w_1, w_2) \leq \frac{q}{2}(1+t^2) \ . \tag{1}$$

In Claim 26 in the Appendix, we show that for $\beta < \frac{1}{2}$, the function $\phi(a) = \frac{1}{\sqrt{2a}}(4\beta(1-\beta))^{a/2}$ is convex for positive a. Thus, applying Jensen's inequality with respect to (1) and Claim 13, by Claim 11 we get

$$\Delta_{1,Q} \leq 2 - 2\frac{\beta}{\sqrt{q(1+t^2)}}(4\beta(1-\beta))^{\frac{q}{4}(1+t^2)} + \frac{3 * 2^q q^2}{m} \ .$$

Note that $g(\beta) = 2\frac{\beta}{\sqrt{q(1+t^2)}}(4\beta(1-\beta))^{\frac{q}{4}(1+t^2)}$ is strictly increasing in β. Thus, $g(\beta)$ evaluated at $\beta = \frac{\delta-\nu}{\gamma}$ is lower bounded by $g(\delta-\nu) \geq g(\delta) - 2(q+1)\nu$. The statement of Theorem 12 follows by Theorem 9. □

4.2 Example: the Hadamard code

The binary Hadamard code of dimension n can be defined by the encoder $\mathbf{C}\colon \{0,1\}^n \to \{0,1\}^m$, where $m = 2^n$, as follows. Denote $\mathbf{C}(x) := (c_a)_{a\in\{0,1\}^n}$. Then $c_a := \langle x, a\rangle$, where the inner product is defined over the binary field. The following classical 2-query decoder for the Hadamard code is easily shown to achieve correctness at least $1 - 2\delta$:

Given a corrupted codeword $(w_a)_{a\in\{0,1\}^n}$, choose $a \in \{0,1\}^n$ uniformly at random and output $w_a + w_{a+e_i}$ (where e_i is the ith standard basis vector and addition is over the binary field) as the decoding of the ith message bit x_i.

A q-query decoder, for even q, can be obtained by repeating the above 2-query decoder independently $q/2$ times and taking majority vote[3]. The following observation analyzes the correctness of this simple decoder.

Lemma 16. *For $\delta < \frac{1}{4}$ and even number of queries q, the Hadamard code achieves*

$$\zeta_\delta \geq 1 - 2^{q/2-1}(2\delta)^{\lceil \frac{q}{4}\rceil}(1-2\delta)^{\lfloor \frac{q}{4}\rfloor} \geq 1 - \Big(4(2\delta)(1-2\delta)\Big)^{\lfloor \frac{q}{4}\rfloor}.$$

For $q = 2$ this gives the $1 - 2\delta$ bound, which we show to be tight below (Theorem 17). In light of the above lemma, we conclude that the exponent $q/4$ in the correctness estimate of Theorem 3 is tight.

Proof. Let z be defined so that $q = 2z$. For clarity, call the classical 2-query decoder as $\hat{A}$. The q-query decoder algorithm will perform $\hat{A}$ as a sub-procedure z times. Each sub-procedure will use its own random coin flips, and hence each answer produced by an $\hat{A}$ will be independent from the other answers, conditioned on the input to the code and the error. The algorithm will take the majority vote of the z answers it receives. In the case that z is even and the vote is tied, assume the algorithm guesses 0 or 1 with equal probability ($\frac{1}{2}$). For any fixed input to the code and error, let the probability, over the randomness

[3] Ties can be broken by outputting a random guess.

of $\hat{A}$, that the adversary makes one sub-procedure wrong be α. The majority vote operation produces the wrong answer when half or more of the sub-procedures return the wrong answer. So the probability of error is:

$$\begin{cases} \sum_{i=0}^{\frac{z-1}{2}} \binom{z}{i} (1-\alpha)^i \alpha^{z-i} & z \text{ odd} \\ \sum_{i=0}^{\frac{z}{2}-1} \binom{z}{i} (1-\alpha)^i \alpha^{z-i} + \frac{1}{2}\binom{z}{z/2}(1-\alpha)^{\frac{z}{2}} \alpha^{\frac{z}{2}} & z \text{ even} \end{cases}$$

We will upper bound these quantities. First note that

$$\begin{cases} \sum_{i=0}^{\frac{z-1}{2}} \binom{z}{i} & z \text{ odd} \\ \sum_{i=0}^{\frac{z}{2}-1} \binom{z}{i} + \frac{1}{2}\binom{z}{z/2} & z \text{ even} \end{cases}$$

both equal 2^{z-1}. Next, note that (by a union bound) $\alpha \leq 2\delta$ and, by assumption, $2\delta < \frac{1}{2}$. Therefore, $\frac{\alpha}{1-\alpha} < 1$, and $(1-\alpha)^i \alpha^{z-i}$ is increasing in i. Therefore, the error probability is upper bounded by

$$2^{z-1}(1-\alpha)^{\lfloor \frac{z}{2} \rfloor} \alpha^{z - \lfloor \frac{z}{2} \rfloor} = 2^{z-1}(1-\alpha)^{\lfloor \frac{z}{2} \rfloor} \alpha^{\lceil \frac{z}{2} \rceil}$$

This expression is increasing in α, and $\alpha \leq 2\delta$, so the result follows.

□

4.3 Correctness of 2-query codes

It is possible to demonstrate more precise bounds than Theorem 3 on the largest possible correctness of 2-query binary codes. For binary linear codes, we obtain the following bound, which is tight, as it matches the correctness achieved by the Hadamard code (up to a sub-constant difference).

Theorem 17. *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a linear code. For any non-adaptive two query decoding algorithm* A*,* $\zeta_\delta(A) \leq \max(\frac{1}{2}, 1 - 2\delta + \frac{2}{n})$*.*

Proof. For $i \in [n]$, define:

$$R_i \triangleq \left\{ j \in [m] \mid \mathbf{C}(x)_j = x_i \right\}$$

So there exists at least one i such that $|R_i| \leq \frac{m}{n}$. Without loss of generality, assume $|R_1| \leq \frac{m}{n}$.

Let $S \subseteq [m]$ be the set of codeword positions that do not depend on x_1. Define T as whichever of S and $\bar{S}$ (the complement of S) has smaller size. If they have the same size, T can be either set. Clearly $|T| \leq \frac{m}{2}$.

Because the argument below works for arbitrary algorithms, without loss of generality, we can assume the algorithm always queries exactly 2 positions. If the algorithm ever queried fewer than 2 positions, have it query more and ignore the additional values obtained.

Define $\gamma \triangleq \frac{|T|}{m}$ and $\beta \triangleq \min(\frac{\delta - \frac{|R_1|}{m}}{\gamma}, \frac{1}{2})$. Let us consider the probability of error of the decoder over uniformly random $x \in \{0,1\}^n$, uniformly random $B_1 \subset T$ such that $|B_1| = \beta\gamma m$, uniformly random $B_2 \subseteq R_1$, and the internal randomness of A. (We emphasize that the corruption B_1 always has the same size; but, for B_2, it is chosen whether to include each member of R_1 independently). Let $B \triangleq B_1 \cup B_2$, generated by the product distribution of B_1 and B_2. We also use B to denote the characteristic vector of the set B. By our choice of the parameters, $|B| \leq \delta m$ always holds.

By Lemma 6, we can, without loss of generality, assume that A never queries any positions from R_1.

Now consider the decomposition:

$$\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1]$$
$$= \sum_{Q\subset[m],|Q|=2} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q] \Pr[A \text{ queries } Q]$$

Define $Err_Q \triangleq \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q]$. We will bound Err_Q depending on all the different possibilities for Q for which $\Pr[A \text{ queries } Q] > 0$. First we give some notation.

For Q and $a, b \in \{0, 1\}$ such that $\Pr_{x,B,A}[A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = ab] > 0$ (where ab denotes concatenation), define

$$p^Q_{ab} \triangleq \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) = 0 \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = ab]$$

For simplicity, let us define the following notation. For a given $S \subseteq Q$,

$$q^{Q,k}_{ab}(S) \triangleq \Pr_{x,B,A}[(\mathbf{C}(x) + B)_Q = ab \mid A \text{ queries } Q; |B \cap S| = k]$$

Let e_1 denote the binary vector of length n with 1 in its first coordinate and 0 everywhere else.

By Lemma 3.2 of Goldreich et al. [9], if e_1 is not spanned by the vectors a_{j_1} and a_{j_2} corresponding to the columns of the generator matrix of the code for $Q = \{j_1, j_2\}$, $Err_Q \geq \frac{1}{2}$. Because $\beta \leq \frac{1}{2}$, $Err_Q \geq \beta$ as well.

If $e_1 \in \text{span}\{a_{j_1}, a_{j_2}\}$, then exactly one bit of Q must be in T – assume it is j_1. We can decompose Err_Q into

$$\begin{aligned} Err_Q &= \sum_{k=0}^{1} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k] \cdot \Pr_B[|B \cap \{j_1\}| = k \mid A \text{ queries } Q] \\ &= \sum_{k=0}^{1} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k] \cdot \Pr_B[|B \cap \{j_1\}| = k] \end{aligned}$$

Note that for any Q, $j_1 \in Q$, and $0 \leq k \leq 1$, the events A queries Q and $|B \cap \{j_1\}| = k$ are independent. So for any Q, $j_1 \in Q$, and $0 \leq k \leq 1$, $\Pr[A \text{ queries } Q; |B \cap \{j_1\}| = k] > 0$. Thus, above we are conditioning on events with nonzero probability. The second equality above also holds because of the independence of A queries Q and $|B \cap \{j_1\}| = k$. For simplicity, define, $Err_{Q,k} \triangleq \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k]$. We can further decompose $Err_{Q,k}$ into

$$Err_{Q,k} = \sum_{a,b} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k; (\mathbf{C}(x) + B)_Q = ab] \cdot$$
$$\Pr_{x,B,A}[(\mathbf{C}(x) + B)_Q = ab \mid A \text{ queries } Q; |B \cap \{j_1\}| = k]$$

Since neither bit is in R_1 but e_1 is in the span of both bits, the sum of the two bits (when uncorrupted) is x_1. So $a + b = x_1 + |B \cap \{j_1\}|$, and the above becomes:

$$\begin{aligned} &Err_{Q,k} = \\ &\sum_{\substack{a,b \\ a+b=k}} q^{Q,k}_{ab}(\{j_1\}) \cdot \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k; (\mathbf{C}(x) + B)_Q = ab] \\ &\quad + \sum_{\substack{a,b \\ a+b=1+k}} q^{Q,k}_{ab}(\{j_1\}) \cdot \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; |B \cap \{j_1\}| = k; (\mathbf{C}(x) + B)_Q = ab] \end{aligned}$$

The event $|B \cap \{j_1\}| = k$ does not depend on the internal randomness of A. Therefore, by Lemma 7,

$$\begin{aligned} &Err_{Q,k} = \\ &\sum_{\substack{a,b\\ a+b=k}} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq 0 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = ab] q_{ab}^{Q,k}(\{j_1\}) + \\ &\sum_{\substack{a,b\\ a+b=1+k}} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq 1 \mid A \text{ queries } Q; (\mathbf{C}(x)+B)_Q = ab] q_{ab}^{Q,k}(\{j_1\}) \end{aligned}$$

This means,

$$Err_{Q,k} = \sum_{\substack{a,b\\ a+b=k}} (1-p_{ab}^Q) q_{ab}^{Q,k}(\{j_1\}) + \sum_{\substack{a,b\\ a+b=1+k}} p_{ab}^Q q_{ab}^{Q,k}(\{j_1\})$$

The two query bits cannot be equal because one is in T and one is not. Since neither is 0, they are linearly independent. Since, also, x is uniformly random, $(\mathbf{C}(x)+B)_{j_1}$ and $(\mathbf{C}(x)+B)_{j_2}$ are two independent, uniformly random bits. Thus, $\forall k, a, b$: $q_{ab}^{Q,k}(\{j_1\}) = \frac{1}{4}$. So, in the $k = 0$ case,

$$Err_{Q,0} = \Big(p_{01}^Q + p_{10}^Q + (1-p_{00}^Q) + (1-p_{11}^Q)\Big)/4$$

For simplicity, define $P_Q \triangleq \Big(p_{01}^Q + p_{10}^Q + (1-p_{00}^Q) + (1-p_{11}^Q)\Big)/4$. On the other hand, in the $k = 1$ case,

$$Err_{Q,1} = \Big((1-p_{01}^Q) + (1-p_{10}^Q) + p_{00}^Q + p_{11}^Q\Big)/4 = 1 - P_Q$$

The probability that j_1 was corrupted is β. Combining everything, we find

$$Err_Q = (1-\beta)P_Q + \beta(1-P_Q) = \beta + (1-2\beta)P_Q$$

Because $\beta \leq \frac{1}{2}$, $Err_Q \geq \beta$.

Since for all Q, $Err_Q \geq \beta$, $\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq \beta$. Thus, there exists an x and B such that $\Pr[A^{\mathbf{C}(x)+B}(1) \neq x_1] \geq \beta$ (where the probability is only over the internal coin flips of A). When $\beta = \frac{1}{2}$, we are done. Otherwise $\beta = \frac{\delta - \frac{|R_1|}{m}}{\gamma} \geq \frac{\delta - \frac{1}{n}}{\gamma}$, and we know $\gamma \leq \frac{1}{2}$. In this case $\beta \geq 2\delta - \frac{2}{n}$. Combining these two possibilities gives the result. □

For arbitrary binary (possibly nonlinear) codes, we prove the following.

Theorem 18. *Let $\mathbf{C}\colon \{0,1\}^n \to \{0,1\}^m$ be a code. For any non-adaptive two query decoding algorithm A, and large enough n, $\zeta_\delta(A) \leq 1 - 2\delta(1-\delta) + O(\frac{1}{n^{1/3}})$.*

Proof. Let $t = 1/n^{1/3}$, and let $\nu \triangleq \frac{1}{.99n(1-H(\frac{1}{2}+\frac{t}{2}))}$. We will show that for any non-adaptive decoding algorithm A, $\zeta(A) \leq 1 - 2\delta(1-\delta) + 2\nu + t \ + \frac{8}{n}$.

We define R_i, B and β as in the proof of Theorem 12. By Lemma 6, we can, without loss of generality, assume that A never queries any positions from R_1.

Without loss of generality, we assume that A flips all of its random coins first, and then, based on those random values, chooses a query set $Q \subset [m]$ and a deterministic function ϕ to apply on the two values it receives from querying Q. Without loss of generality, $Q = \{1, 2\}$. We use the shorthand "Q, ϕ" to mean the event A has chosen to query Q and applies the function ϕ on the query results. Now consider the decomposition:

$$\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1] = \sum_{Q \subset [m]: |Q|=2, \phi} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid Q, \phi] \Pr[Q, \phi]$$

Define $Err_{Q,\phi} \triangleq \Pr_{x,B}[A^{\mathbf{C}(x)+B}(1) \neq x_1 \mid Q, \phi]$. Recall that the correlation between two Boolean functions f and g is defined as

$$Corr(f,g) \triangleq \Pr_x[f(x) = g(x)] - \Pr_x[f(x) \neq g(x)]$$

Let $\chi_S(Y_1, Y_2) \triangleq \sum_{s \in S} Y_s$ for $S \subseteq \{1,2\}$, then (as shown in [6])

$$\left|Corr(x_i, \phi(Y_1,Y_2))\right| \leq \left|Corr(x_i, 0)\right| + \sum_{S \subseteq \{1,2\} \,:\, |S|=1} \left|Corr(x_i, \chi_S(Y_1,Y_2))\right| + \left|Corr(x_i, Y_1 + Y_2)\right| .$$

The first term of this expression is 0 because $\Pr_x[x_i = 0] = \frac{1}{2}$. The two absolute values in the second term are each at most t. This is because for any $j \in [m]$, if $|Corr(x_i, \mathbf{C}(x)_j)| > t$, then j_1 is corrupted by B into a uniformly random value in $\{0,1\}$. Therefore, the correlation of the corrupted value with x_i is 0. This gives:

$$\left|Corr(x_i, \phi(Y_1,Y_2))\right| \leq 2t + \left|Corr(x_i, Y_1 + Y_2)\right|$$

Because of the independence of x and B, we have:

$$\left|Corr(x_i, Y_1 + Y_2)\right| \leq \left|Corr(0, B_1 + B_2)\right|$$

Because $\beta \leq \frac{1}{2}$, we have

$$\begin{aligned} \left|Corr(x_i, \phi(Y_1,Y_2))\right| &\leq 2t + \Big(1 - \Pr_B[|B \cap Q| = 1]\Big) - \Big(\Pr_B[|B \cap Q| = 1]\Big) \\ &= 2t + 1 - 2\Big(\Pr_B[|B \cap Q| = 1]\Big) \\ &\leq 2t + 1 - 2\Big(2\beta(1-\beta) - \frac{8}{m}\Big) \end{aligned}$$

Noting that $Err_{Q,\phi} \leq \frac{1}{2}$ or else the algorithm would just guess randomly, we have:

$$(1 - Err_{Q,\phi}) - Err_{Q,\phi} = \left|(1 - Err_{Q,\phi}) - Err_{Q,\phi}\right| = \left|Corr(x_i, \phi(Y_1,Y_2))\right|$$

Thus we have $Err_{Q,\phi} \geq 2\beta(1-\beta) - t - \frac{8}{m}$.

Recall that $\beta = \min(\frac{\delta-\nu}{\gamma}, \frac{1}{2})$. When $\beta = \frac{\delta-\nu}{\gamma}$, first note that the expression $2\beta(1-\beta)$ is strictly increasing in β. Therefore, we can lower bound $2\beta(1-\beta) - t - \frac{8}{m}$ evaluated at $\beta = \frac{\delta-\nu}{\gamma}$ with $2\hat{\beta}(1-\hat{\beta}) - t - \frac{8}{m}$ evaluated at $\hat{\beta} = \delta - \nu$:

$$2\Big(\delta - \nu\Big)\Big(1 - \delta + \nu\Big) - t - \frac{8}{m} \geq 2\delta(1-\delta) - 2\nu - t - \frac{8}{m}$$

The lower bound of Katz and Trevisan [11] implies that $m > n$, for large enough n. Therefore, this expression is more than $2\delta(1-\delta) - 2\nu - t - \frac{8}{n}$.

When $\beta = \frac{1}{2}$, $2\beta(1-\beta) - t - \frac{8}{m} = \frac{1}{2} - t - \frac{8}{n}$ for large enough n (again, note that $m > n$). □

5 Minimum distance and largest tolerable corruption

In this section, we study the relationship between the amount of corruption tolerable by LDCs and their minimum distance as an error-correcting code. As we noted in the introduction, while intuitively it is expected that the two notions are related, in general, for non-binary codes this may not be the case. Then, we study the largest possible corruption parameter δ that any LDC may have.

5.1 Corruption versus minimum distance

It is easy to see that for non-binary codes, local decodability does not imply large minimum distance. As an example, consider the ternary code $\mathbf{C}\colon \{-1,0,+1\}^n \to \{-1,0,+1\}^m$ with $m \triangleq n + 2^n$ defined as $\mathbf{C}(x_1,\dots,x_n) \triangleq (x_1,\dots,x_n,H_1,\dots,H_{2^n})$ where $(H_1,\dots,H_{2^n})$ is the binary Hadamard encoding of the binary vector $(|x_1|,\dots,|x_n|)$. The absolute minimum distance of this code is 1 which can be seen, for example, by looking at the encodings of the two vectors $(1,0,\dots,0)$ and $(-1,0,\dots,0)$. However, this code is a $(2,\delta,\varepsilon)$-LDC according to Definition 1 for every constant $\delta \in [0,1/12)$ and $\varepsilon \triangleq 1/6 - 2\delta - o(1)$. Namely, in order to locally decode a message bit x_i, it suffices to run the standard 2-query local decoder of the Hadamard code on $(H_1,\dots,H_{2^n})$ to obtain $\tilde{x}_i \in \{0,1\}$. If $\tilde{x}_i = 0$, the decoder outputs 0 and otherwise randomly outputs -1 or $+1$ with equal probabilities. If $x_i = 0$, this procedure errs with probability at most $2\delta + o(1)$ (as each of the two queries coincide with an error position with probability at most $\delta + o(1)$), and otherwise the error probability would be at most $1/2 + 2\delta + o(1)$ (since the coin flip to decide between -1 and $+1$ errs with probability $1/2$). Altogether this decoder attains correctness $1/2 - 2\delta$ which is greater than $1/3$ (as required by Definition 1) by at least ε. Thus, for non-binary codes, the minimum distance may be very small, even for codes that tolerate a large fraction of errors as LDCs.

However, we show a direct relationship between minimum distance and the fraction of errors tolerated by LDCs in the case of binary codes. Moreover, we are able to extend this result to linear codes over arbitrary finite fields.

Lemma 19. *Let $\mathbf{C}\colon \{0,1\}^n \to \{0,1\}^m$ be a (q,δ,ϵ)-LDC with $\epsilon > 0$. Then $\mathbf{C}$ has minimum distance at least $2\delta m + 1$.*

Proof. Assume there are two codewords $\mathbf{C}(a)$ and $\mathbf{C}(b)$ with $a \neq b \in \{0,1\}^n$ such that the Hamming distance between them is less than $2\delta m + 1$. Because $a \neq b$, a and b differ in at least one bit – without loss of generality, let $i \in [n]$ be one such bit in the support of $a - b$. Because $d(\mathbf{C}(a),\mathbf{C}(b)) \leq 2\delta m$, there exists a string, call it Y, such that $d(\mathbf{C}(a),Y) \leq \delta m$ and $d(Y,\mathbf{C}(b)) \leq \delta m$. Whenever the input to the code is a or b, the adversary will change the codeword into Y. Either $\Pr[A^Y(i)$ outputs $1] \leq \frac{1}{2}$ or $\Pr[A^Y(i)$ outputs $1] \geq \frac{1}{2}$, where the probabilities are over the internal coin tosses of A. In the first case, the algorithm fails with probability at least $\frac{1}{2}$ on whichever input a or b has i'th position 1. In the second case, the algorithm fails with probability at least $\frac{1}{2}$ on whichever input a or b has i'th position 0. Thus, in either case, we have shown there exists an input and an adversary error pattern of size at most δm so that the probability of error is at least $\frac{1}{2}$, which contradicts the assumption that $\epsilon > 0$. □

We remark that Lemma 19 can be alternatively proved by independently running the local decoder sufficiently many times and taking majority votes for each message position, thus recovering the entire message from corrupted encodings. However, as we noted in the introduction, this argument cannot be used for non-binary codes. The proof presented here can be generalized to arbitrary fields for the case of linear codes, as follows.

Lemma 20. *Let $\mathbf{C}\colon F^n \to F^m$ be a linear (q,δ,ϵ)-LDC with $\epsilon > 0$. Then $\mathbf{C}$ has minimum distance at least $\frac{|F|}{|F|-1}\delta m + 1$.*

Proof. Assume there are two codewords $\mathbf{C}(g_0)$ and $\mathbf{C}(g_1)$ with $g_0 \neq g_1 \in F^n$ such that the Hamming distance between them is less than $\frac{|F|}{|F|-1}\delta m + 1$. Then, for $f \in F, f \neq 0,1$ define $g_f \triangleq g_0 + f(g_1 - g_0)$.

Because $g_0 \neq g_1$, g_0 and g_1 differ in at least one position – without loss of generality, let $i \in [n]$ be one such position in the support of $g_0 - g_1$. For $f \in F$, define h_f as the unique $g_{f'}$ $(f' \in F)$ such $(g_{f'})_i = f$.

Construct a string Y in the following way. In the positions outside the support of $\mathbf{C}(g_0) - \mathbf{C}(g_1)$, let Y equal $\mathbf{C}(g_0)$. (Notice for later that because $\mathbf{C}$ is linear, the $\mathbf{C}(g_f)$ are identical outside of the support of $\mathbf{C}(g_0) - \mathbf{C}(g_1)$.) Divide the positions in the support of $\mathbf{C}(g_0) - \mathbf{C}(g_1)$ into $|F|$ equal pieces and label each piece by a member of F. For the positions in the $f \in F$ piece, let Y be the same as h_f. This implies

$\forall f \in F, d(\mathbf{C}(h_f), Y) \leq \frac{|F|-1}{|F|}\frac{|F|}{|F|-1}\delta m = \delta m$. Whenever the input to the code is h_f, for some $f \in F$, the adversary will change the codeword into Y. Now $\sum_{f\in F}\Pr[A^Y(i)$ outputs $f] = 1$ where the probability is over the internal coin tosses of A. So there exists at least one $f \in F$ such that, if the adversary corrupts $\mathbf{C}(h_f)$ into Y, the probability of the algorithm correctly answering f is at most $\frac{1}{|F|}$. Therefore, we have shown there exists an input x and an adversary error pattern of size at most δm so that the probability of error is at least $1 - \frac{1}{|F|}$, which contradicts the assumption that $\epsilon > 0$. □

Even though the code in the example at the beginning of this section has very small minimum distance, the code still contains a large subcode with large minimum distance. Namely, the set of codewords corresponding to messages that lie in $\{0, 1\}$ is a subcode of size 2^n, as opposed to size 3^n of the code $\mathbf{C}$, and relative distance at least 1/2. This brings up the following question:

Question: Does every (q, δ, ε)-LDC with constant $\varepsilon > 0$ and message length n contain a subcode of size $\exp(n)$ and relative minimum distance at least 2δ?

Lemma 19 shows that the answer to this question is positive for binary codes (the code itself must have relative minimum distance at least 2δ). For non-binary codes we are able to show the following.

Proposition 21. *Let $\mathbf{C}\colon \Sigma^n \to \Gamma^m$ be any (q, δ, ε)-LDC with $\varepsilon > 0$. Then, $\mathbf{C}$ has a sub-code of size at least $(|\Sigma|/(|\Sigma| - 1))^n$ that has relative minimum distance at least δ.*

Proof. Consider any $y \in \Gamma^m$, and for every $i \in [n]$, denote by X_i the probability distribution on Σ induced by applying the local decoding algorithm on the received word y and index i (this distribution depends on the choice of y and the internal coin flips of the decoder). Let $w_i \in \Sigma$ be the symbol for which $p_i \triangleq \Pr[X_i = w_i]$ is the least. By an averaging argument, $p_i \leq 1/|\Sigma|$. Thus, for any $x = (x_1, \dots, x_n) \in \Sigma^n$ with $x_i = w_i$ we must have $d(\mathbf{C}(x), y) > \delta m$ since otherwise, y can be interpreted as a corruption of $\mathbf{C}(x)$ and the definition of locally decodable codes would then imply that $p_i \geq 1/|\Sigma| + \varepsilon$, which we know is not the case. We conclude that for any y, the Hamming ball of radius δm around y can contain at most $(|\Sigma| - 1)^n$ codewords.

Now consider the following greedy procedure: Start with any codeword y in $\mathbf{C}$, keep y in the code, remove all codewords at distance up to δm of y (we know there are at most $(|\Sigma| - 1)^n$), and continue the procedure with unseen codewords until the code is exhausted. The resulting sub-code of $\mathbf{C}$ has at least $(|\Sigma|/(|\Sigma| - 1))^n$ codewords. Moreover, the codewords belonging to the sub-code have pairwise distances of δm or more by construction. □

As we saw above, for non-binary codes, local decodability in general does not imply large distance. We show that under the stronger Definition 4, Proposition 21 can be strengthened as follows.

Proposition 22. *Let $\mathbf{C}\colon \Sigma^n \to \Gamma^m$ be a strong (q, δ, ε)-LDC with $\varepsilon > 0$. Then, $\mathbf{C}$ has relative minimum distance greater than 2δ.*

Proof. Consider any $y \in \Gamma^m$ and for every $i \in [n]$, denote by X_i the probability distribution on Σ induced by providing the local decoding algorithm with the received word y and index i. Let $x_i \in \Sigma$ be the symbol for which $p_i \triangleq \Pr[X_i = x_i]$ is the largest. Definition 4 implies that the Hamming ball of radius δm around y can only contain up to one codeword; namely, $\mathbf{C}(x_1, \dots, x_n)$. As a result, the Hamming balls of radius δm around codewords must not collide. □

5.2 Largest tolerable amount of corruption

We now combine the results of the previous subsection with Plotkin's bound to obtain upper bounds on the corruption parameter δ that any LDC may allow. Recall that the Plotkin bound on codes (cf. [15]) asserts that any code $\mathbf{C}\colon \Sigma^n \to \Sigma^m$ with minimum distance $d > (1 - 1/|\Sigma|)m$ has size $|\mathbf{C}| \leq d/(d - (1 - 1/|\Sigma|)m)$. In particular, when $d = (1-1/|\Sigma|+\gamma)m$, we have $|\mathbf{C}| \leq 1/\gamma$. By choosing, say, $\gamma \triangleq 1/n$, one can ensure that there is no code[4] over Σ with message length n and relative distance at least $1-1/|\Sigma|+1/n = 1-1/|\Sigma|+o(1)$. Combining this with Proposition 21 gives $\delta < 1-1/|\Sigma|+\frac{1}{n}$, which almost recovers the bound $\delta < 1-1/|\Sigma|$ proved in [6] (Observation 2.1). Using Lemma 19, Lemma 20, and Proposition 22, respectively, we get the following stronger bounds:

Corollary 23. *Let* $\mathbf{C}\colon \Sigma^n \to \Sigma^m$ *be any* (q, δ, ε)*-LDC where* $\varepsilon > 0$*. Then,*

1. *If* $|\Sigma| = 2$*, then* $\delta < 1/4 + \frac{1}{n}$*.*
2. *If* $\mathbf{C}$ *is linear, then* $\delta < (1-1/|\Sigma|)^2 + \frac{1}{n}$*.*
3. *If* $\mathbf{C}$ *satisfies the stronger Definition 4, then* $\delta < \frac{1}{2}(1-1/|\Sigma|) + \frac{1}{n}$*.*

References

[1] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 21–32, New York, NY, USA, 1991.

[2] Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *7th International Symposium on Theoretical Aspects of Computer Science (STACS 1990)*, pages 37–48, 1990.

[3] Avraham Ben-Aroya, Klim Efremenko, and Amnon Ta-Shma. Local list decoding with a constant number of queries. In *Proceedings of the 51st IEEE Symposium on Foundations of Computer Science (FOCS'10)*, pages 715–722, Washington, DC, USA, 2010.

[4] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM J. on Computing*, 40(4):1154–1178, 2011.

[5] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 39–44, New York, NY, USA, 2009.

[6] Anna Gál and Andrew Mills. Three query linear locally decodable codes with higher correctness require exponential length. *ACM Transactions on Computation Theory, Vol. 3, No. 2, Article 5 (see also ECCC preprint TR11-030 and STACS 2011)*, 2012.

[7] Peter Gemmell, Richard Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self testing/correcting for polynomials and for approximate functions. In *Proceedings of the 23rd ACM Symposium on Theory of Computing (STOC)*, pages 32–42, 1991.

[8] Peter Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43:169–174, 1992.

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

[10] Rahul Jain. Towards a classical proof of exponential lower bound for 2-probe smooth codes. Manuscript (arXiv:cs/0607042), 2006.

[11] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the 32nd annual ACM Symposium on Theory of Computing (STOC'00)*, pages 80–86, New York, NY, USA, 2000.

[4] In fact, any $\gamma > 1/|\Sigma|^n$ would work.

[12] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC'03)*, pages 106–115, New York, NY, USA, 2003.

[13] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. In *Proceedings of the 43rd annual ACM symposium on Theory of Computing (STOC'11)*, STOC '11, pages 167–176, New York, NY, USA, 2011.

[14] Richard Lipton. Efficient checking of computations. In *Proceedings of the 7th International Symposium on Theoretical Aspects of Computer Science (STACS'90)*, pages 207–215, 1990.

[15] F.J. MacWilliams and N.J. Sloane. *The Theory of Error-Correcting Codes.* North Holand, 1977.

[16] Prasad Raghavendra. A note on Yekhanin's locally decodable codes. *ECCC TR07-016*, 2007.

[17] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and Systems Sciences*, 62(2):236–266, 2001.

[18] S. Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *Proceedings of the 32nd ICALP*, volume 3580 of *LNCS*, pages 1424–1436, 2005.

[19] David Woodruff. Some new lower bounds for general locally decodable codes. *ECCC TR07-006*, 2007.

[20] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM*, 55(1):1–16, 2008.

[21] Sergey Yekhanin. *Locally decodable codes.* NOW Publishers, 2010.

[22] Sergey Yekhanin. Locally decodable codes: a brief survey. In *Proceedings of the 3rd International Workshop on Coding and Cryptography (IWCC)*, 2011.

A Appendix

A.1 Proof of Lemma 7

A simple, but important point used in the proof below is that, for any Q, the value of $(\mathbf{C}(x)+B)_Q$ is independent of the event A queries Q, since the decoder A is non-adaptive. Note however, that while E does not depend on the internal randomness of A, it may depend on the distribution B or the input x. Thus, the events we work with in general are not independent events.

Without loss of generality, assume A makes all of its coin flips in advance of querying any codeword positions. Let r denote the event that the outcome of these coin flips is a particular string r. Then we have:

$$\begin{aligned}&\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i)=v \mid A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q=s] = \\ &\qquad \sum_r \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i)=v \mid A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q=s; r]\cdot \\ &\qquad\quad \Pr_{x,B,A}[r \mid A \text{ queries } Q; E; (\mathbf{C}(x)+B)_Q=s]\end{aligned}$$

Since $|Q|=q$, for a fixed setting of the decoder's random bits, the output of the decoder is completely determined by the values $(\mathbf{C}(x)+B)_Q$, if A queries Q. Thus, $\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i)=v \mid A$ queries $Q; (\mathbf{C}(x)+$

$B)_Q = s; r]$ is either 0 or 1. An event with probability 0 or probability 1 remains of probability 0 or 1, respectively, under any conditioning. Therefore, we can remove the conditioning on E from $\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s; r]$ and get:

$$\begin{aligned}
&\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s] \\
&\quad = \sum_r \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = s; r] \cdot \\
&\qquad\qquad \Pr_{x,B,A}[r \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s]
\end{aligned}$$

Next, we note that for any r,

$$\begin{aligned}
&\Pr_{x,B,A}[r \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s] \\
&\quad = \frac{\Pr_{x,B,A}[r; A \text{ queries } Q]}{\Pr_{x,B,A}[A \text{ queries } Q]} \\
&\quad = \Pr_{x,B,A}[r \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = s] \ .
\end{aligned}$$

Above, we have used the fact that r and A queries Q is independent of E and the values of $\mathbf{C}(x) + B$ on the positions indexed by Q. Therefore,

$$\begin{aligned}
&\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; E; (\mathbf{C}(x) + B)_Q = s] \\
&\quad = \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) = v \mid A \text{ queries } Q; (\mathbf{C}(x) + B)_Q = s] \ ,
\end{aligned}$$

and this concludes the proof of the Lemma.

A.2 Helpful facts

The following bound was proved in [6].

Claim 24. *For large enough m,*

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} > \binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^2}{m}$$

We also use the following upper bound.

Claim 25. *For large enough m,*

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} < \binom{q}{k}\delta^k(1-\delta)^{q-k} + \frac{2q^2}{m}$$

Proof. When $0 \leq k < q$:

$$\begin{aligned}
&\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} = \binom{q}{k}\frac{(\delta m)!(m-\delta m)!}{m!}\frac{(m-q)!}{(\delta m-k)!(m-\delta m-q+k)!} = \binom{q}{k}\cdot \\
&\frac{\delta m(\delta m-1)...(\delta m-k+1)(m-\delta m)(m-\delta m-1)...(m-\delta m-q+k+1)}{m(m-1)...(m-q+1)} \\
&< \binom{q}{k}\frac{(\delta m)^k(m-\delta m)(m-\delta m-1)...(m-\delta m-q+k+1)}{m(m-1)...(m-q+1)} \\
&\leq \binom{q}{k}\frac{(\delta m)^k(m-\delta m)^{q-k}}{m^{q-k}(m-q+k)(m-q+k-1)...(m-q+1)} \\
&< \binom{q}{k}\frac{(\delta m)^k(m-\delta m)^{q-k}}{m^{q-k}(m-q)^k} \\
&= \binom{q}{k}\delta^k(1-\delta)^{q-k}\frac{1}{(1-\frac{q}{m})^k} \\
&< \binom{q}{k}\delta^k(1-\delta)^{q-k}\frac{1}{1-\frac{kq}{m}} \qquad \text{for } m \text{ large enough} \\
&< \binom{q}{k}\delta^k(1-\delta)^{q-k}(1+\frac{2kq}{m}) \qquad \text{for } m \text{ large enough} \\
&\leq \binom{q}{k}\delta^k(1-\delta)^{q-k} + \frac{2q^2}{m} \qquad \text{because } \binom{q}{k}\delta^k(1-\delta)^{q-k} \leq 1
\end{aligned}$$

When $k = q$:

$$\frac{\binom{m-q}{\delta m-q}}{\binom{m}{\delta m}} = \frac{\delta m(\delta m-1)...(\delta m-q+1)}{m(m-1)...(m-q+1)} < (\frac{\delta m}{m})^q = \delta^q$$

□

Claim 26. *Given that $\beta < \frac{1}{2}$, the function $\phi(a) = \frac{1}{\sqrt{2a}}(4\beta(1-\beta))^{a/2}$ is convex for positive a.*

Proof. Note that $\phi(a)$ can be rewritten as $\chi(b) \triangleq \frac{\gamma^b}{2\sqrt{b}}$ where $b \triangleq a/2$ and $\gamma \triangleq 4\beta(1-\beta) < 1$.

The first derivative of $\chi(b)$, with respect to b, is

$$\gamma^b(-\frac{1}{4}b^{-3/2} + \frac{1}{2}b^{-1/2}\log\gamma)$$

The second derivative of $\chi(b)$ is

$$\gamma^b(\frac{3}{8}b^{-5/2} - \frac{1}{2}b^{-3/2}\log\gamma + \frac{1}{2}b^{-1/2}(\log\gamma)^2)$$

Because $\log\gamma < 0$, $\gamma > 0$, and $b > 0$, each term of the last line is positive. So the last line as a whole is positive. Also note that these first and second derivatives are continuous over the domain of positive b. So $\chi(b)$ is convex. Because the derivative of b with respect to a is $\frac{1}{2} > 0$, then, by the chain rule, $\phi(a)$ is convex as well. □