On Relaxed Locally Decodable Codes for Hamming and Insertion-Deletion Errors

Alexander R. Block
University of Maryland, College Park, USA
Georgetown University, USA

Jeremiah Blocki
Department of Computer Science, Purdue University, USA

Kuan Cheng
Center on Frontiers of Computing Studies, Peking University, China

Elena Grigorescu
Department of Computer Science, Purdue University, USA

Xin Li
Department of Computer Science, Johns Hopkins University, USA

Yu Zheng
Meta Platforms, Inc., USA

Minshen Zhu
Department of Computer Science, Purdue University, USA

Abstract

Locally Decodable Codes (LDCs) are error-correcting codes $C : \Sigma^n \to \Sigma^m$, encoding *messages* in Σ^n to *codewords* in Σ^m, with super-fast decoding algorithms. They are important mathematical objects in many areas of theoretical computer science, yet the best constructions so far have codeword length m that is super-polynomial in n, for codes with constant query complexity and constant alphabet size.

In a very surprising result, Ben-Sasson, Goldreich, Harsha, Sudan, and Vadhan (SICOMP 2006) show how to construct a relaxed version of LDCs (RLDCs) with constant query complexity and almost linear codeword length over the binary alphabet, and used them to obtain significantly-improved constructions of Probabilistically Checkable Proofs.

In this work, we study RLDCs in the standard Hamming-error setting, and introduce their variants in the insertion and deletion (Insdel) error setting. Standard LDCs for Insdel errors were first studied by Ostrovsky and Paskin-Cherniavsky (*Information Theoretic Security, 2015*), and are further motivated by recent advances in DNA random access bio-technologies.

Our first result is an exponential lower bound on the length of Hamming RLDCs making 2 queries (even adaptively), over the binary alphabet. This answers a question explicitly raised by Gur and Lachish (SICOMP 2021) and is the first exponential lower bound for RLDCs. Combined with the results of Ben-Sasson et al., our result exhibits a "phase-transition"-type behavior on the codeword length for some constant-query complexity. We achieve these lower bounds via a transformation of RLDCs to standard Hamming LDCs, using a careful analysis of restrictions of message bits that fix codeword bits.

We further define two variants of RLDCs in the Insdel-error setting, a weak and a strong version. On the one hand, we construct weak Insdel RLDCs with almost linear codeword length and constant query complexity, matching the parameters of the Hamming variants. On the other hand, we prove exponential lower bounds for strong Insdel RLDCs. These results demonstrate that, while these variants are equivalent in the Hamming setting, they are significantly different in the insdel setting. Our results also prove a strict separation between Hamming RLDCs and Insdel RLDCs.

2012 ACM Subject Classification Theory of computation → Error-correcting codes; Mathematics of computing → Coding theory; Theory of computation → Lower bounds and information complexity

Keywords and phrases Relaxed Locally Decodable Codes, Hamming Errors, Insdel Errors, Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.CCC.2023.14

Related Version *Full Version*: https://arxiv.org/abs/2209.08688

Funding *Alexander R. Block*: NSF Award CCF-1910659 and DARPA agreement No. HR00112020022 and No. HR00112020025. The views, opinions, findings, conclusions and/or recommendations expressed in this material are those of the author and should not be interpreted as reflecting the position or policy of the Department of Defense or the U.S. Government, and no official endorsement should be inferred.
Jeremiah Blocki: NSF CAREER Award CNS-2047272 and NSF Award CCF-1910659
Kuan Cheng: A start-up fund from Peking University, Innovation Program for Quantum Science and Technology 2021ZD0302900
Elena Grigorescu: NSF CCF-1910659, NSF CCF-1910411, and NSF CCF-2228814
Xin Li: NSF CAREER Award CCF-1845349 and NSF Award CCF-2127575
Yu Zheng: NSF CAREER Award CCF-1845349
Minshen Zhu: NSF CCF-1910659, NSF CCF-1910411, and NSF CCF-2228814

Acknowledgements We are indebted to some anonymous reviewers who helped us improve the presentation of the paper.

1 Introduction

Locally Decodable Codes (LDCs) [55, 72] are error-correcting codes $C : \Sigma^n \to \Sigma^m$ that have super-fast decoding algorithms that can recover individual symbols of a *message* $x \in \Sigma^n$, even when worst-case errors are introduced in the *codeword* $C(x)$. Similarly, Locally Correctable Codes (LCCs) are error-correcting codes $C : \Sigma^n \to \Sigma^m$ for which there exist very fast decoding algorithms that recover individual symbols of the *codeword* $C(x) \in \Sigma^m$, even when worst-case errors are introduced. LDCs/LCCs were first discovered by Katz and Trevisan [55] and since then have proven to be crucial tools in many areas of computer science, including private information retrieval, probabilistically checkable proofs, self-correction, fault-tolerant circuits, hardness amplification, and data structures (e.g., [2, 4, 17, 18, 20, 28, 62] and surveys [36, 73]).

The *parameters* of interest of these codes are their *rate*, defined as the ratio between the message length n and the codeword length m, their *relative minimum distance*, defined as the minimum normalized Hamming distance between any pair of codewords, and their *locality* or *query complexity*, defined as the number of queries a decoder makes to a received word $y \in \Sigma^m$. Trade-offs between the achievable parameters of Hamming LDCs/LCCs have been studied extensively over the last two decades [8–11, 32–35, 37, 56, 57, 74, 75, 78, 79] (see also surveys by Yekhanin [79] and by Kopparty and Saraf [58]).

Specifically, for 2-query Hamming LDCs/LCCs it is known that $m = 2^{\Theta(n)}$ [6, 11, 37, 56]. However, for $q > 2$ queries, the current gap between upper and lower bounds is superpolynomial in n. In particular, the best constructions have super-polynomial codeword length [32, 34, 78], while the most general lower bounds for $q \geq 3$ are of the form $m = \Omega((\frac{n}{\log n})^{1+1/(\lceil \frac{q}{2} \rceil - 1)})$ [55, 56]. In particular, for $q = 3$, [55] showed an $m = \Omega(n^{3/2})$ bound, which was improved in [56] to $m = \Omega(n^2/\log^2 n)$. This was further improved by [75, 76] to $m = \Omega(n^2/\log n)$ for general codes and $m = \Omega(n^2)$ for linear codes. [11] used new combinatorial techniques to obtain the same $m = \Omega(n^2/\log n)$ bound. A very recent paper [1] breaks the quadratic barrier and proves that $m = \Omega(n^3/\operatorname{poly}\log n)$. We note that the exponential lower bound on the length of 3-query LDCs from [35] holds only for some restricted parameter regimes, and do not apply to the natural ranges of the known upper bounds.

Motivated by this large gap in the constant-query regime, as well as by applications in constructions of Probabilistically Checkable Proofs (PCPs), Ben-Sasson, Goldreich, Harsha,

Sudan, and Vadhan [7] introduced a relaxed version of LDCs for Hamming errors. Specifically, the decoder is allowed to output a "decoding failure" answer (marked as "$\perp$"), as long as it errs with some small probability. More precisely, a $(q, \delta, \alpha, \rho)$-*relaxed LDC* is an error-correcting code satisfying the following properties.

▶ **Definition 1.** *A $(q, \delta, \alpha, \rho)$-Relaxed Locally Decodable Code $C : \Sigma^n \to \Sigma^m$ is a code for which there exists a decoder that makes at most q queries to the received word y, and satisfies the following further properties:*

1. *(Perfect completeness) For every $i \in [n]$, if $y = C(x)$ for some message x then the decoder, on input i, outputs x_i with probability* 1.[1]
2. *(Relaxed decoding) For every $i \in [n]$, if y is such that $dist(y, C(x)) \leq \delta$ for some unique $C(x)$, then the decoder, on input i, outputs x_i or $\perp$ with probability $\geq \alpha$.*
3. *(Success rate) For every y such that $dist(y, C(x)) \leq \delta$ for some unique $C(x)$, there is a set I of size $\geq \rho n$ such that for every $i \in I$ the decoder, on input i, correctly outputs x_i with probability $\geq \alpha$.*

We will call an RLDC that satisfies all 3 *conditions by the notion of* strong *RLDC, and one that satisfies just the first* 2 *conditions by the notion of* weak *RLDC, in which case it is called a (q, δ, α)-RLDC. Furthermore, if the q queries are made in advance, before seeing entries of the codeword, then the decoder is said to be* non-adaptive*; otherwise, it is called* adaptive*.*

The above definition is quite general, in the sense that $dist(a, b)$ can refer to several different distance metrics. In the most natural setting, we use $dist(a, b)$ to mean the "relative" Hamming distance between $a, b \in \Sigma^m$, namely $dist(a, b) = |\{i \colon a_i \neq b_i\}|/m$. This corresponds to the standard RLDCs for Hamming errors. As it will be clear from the context, we also use $dist(a, b)$ to mean the "relative" Edit distance between $a, b \in \Sigma^*$, namely $dist(a, b) = \mathsf{ED}(a, b)/(|a| + |b|)$, where $\mathsf{ED}(a, b)$ is the minimum number of insertions and deletions to transform string a into b. This corresponds to the new notion introduced and studied here, which we call *Insdel RLDCs*. Throughout this paper, we only consider the case where $\Sigma = \{0, 1\}$.

Definition 1 has also been extended recently to the notion of *Relaxed Locally Correctable Codes (RLCCs)* by Gur, Ramnarayan, and Rothblum [40]. RLDCs and RLCCs have been studied in a sequence of exciting works, where new upper and lower bounds have emerged, and new applications to probabilistic proof systems have been discovered [3, 27, 29, 38–40].

Surprisingly, [7] constructs strong RLDCs with $q = O(1)$ queries and $m = n^{1+O(1/\sqrt{q})}$, and more recently Asadi and Shinkar [3] improve the bounds to $m = n^{1+O(1/q)}$, in stark contrast with the state-of-the-art constructions of standard LDCs. Gur and Lachish [39] show that these bounds are in fact tight, as for every $q \geq 2$, every weak q-query RLDC must have length $m = n^{1+1/O(q^2)}$ for non-adaptive decoders. We remark that the lower bounds of [39] hold even when the decoder does not have perfect completeness and in particular valid message bits are decoded with success probability 2/3. Dall'Agnon, Gur, and Lachish [30] further extend these bounds to the setting where the decoder is adaptive, with $m = n^{1+1/O(q^2 \log^2 q)}$.

[1] We remark that the initial definition in [7] only requires that x_i is output with probability 2/3 when there are no errors. However, to the best of our knowledge, all constructions of RLDCs (and LDCs) from the literature do satisfy perfect completeness. Moreover, some lower bounds (e.g., [11]) only hold with respect to perfect completeness.

1.1 Our results

As discussed before, since the introduction of RLDCs, unlike standard LDCs, they displayed a behaviour amenable to nearly linear-size constructions, with almost matching upper and lower bounds. However, recently [39] conjecture that for $q = 2$ queries, there is in fact an exponential lower bound, matching the bounds for standard LDCs.

In this paper, our first contribution is a proof of their conjecture, namely to show that Hamming 2-query RLDCs require exponential length. In fact, our exponential lower bound for $q = 2$ applies even to weak RLDCs, which only satisfy the first two properties (perfect completeness and relaxed decoding), and even for adaptive decoders.

▶ **Theorem 2.** *Let $C\colon \{0,1\}^n \to \{0,1\}^m$ be a weak adaptive $(2, \delta, 1/2 + \varepsilon)$-RLDC. Then $m = 2^{\Omega_{\delta,\varepsilon}(n)}$.*

Our results are the first exponential bounds for RLDCs. Furthermore, combined with the constructions with nearly linear codeword length for some constant number of queries [3, 7], our results imply that RLDCs experience a "phase transition"-type phenomena, where the codeword length drops from being exponential at $q = 2$ queries to being almost linear at $q = c$ queries for some constant $c > 2$. In particular, this also implies that there is a query number q where the codeword length drops from being super-polynomial at q to being polynomial at $q + 1$. Finding this exact threshold query complexity is an intriguing open question.

As our second contribution, we introduce and study the notion of RLDCs correcting *insertions and deletions*, namely Insdel RLDCs. The non-relaxed variants of Insdel LDCs were first introduced in [68], and were further studied in [12, 13, 26]. Local decoding in the Insdel setting is motivated in DNA storage [77], and in particular [5] show recent advances in bio-technological aspects of random access to data in these precise settings.

In [13, 68], the authors give Hamming to Insdel reductions which transform any Hamming LDC into an Insdel LDC with rate reduced by a constant multiplicative factor, and locality increased by a polylog(m) multiplicative factor. Unfortunately, these compilers do not imply constant-query Insdel LDCs, whose existence is still an open question.

The results of [14] show strong lower bounds on the length of constant-query Insdel LDCs. In particular, they show that linear Insdel LDCs with 2 queries do no exist, general Insdel LDCs for $q = 3$ queries must have $m = \exp(\Omega(\sqrt{n}))$, and for $q \geq 4$ they must have $m = \exp(n^{\Omega(1/q)})$.

In this work we continue the study of locally decodable codes in insertion and deletion channels by proving the first upper and lower bounds regarding the relaxed variants of Insdel LDCs. We first consider strong Insdel RLDCs, which satisfy all three properties of Definition 1 and where the notion of distance is now that of relative edit distance. We adapt and extend the results of [14] to establish strong lower bounds on the codeword length of strong Insdel RLDCs. In particular, we prove that $m = \exp(n^{\Omega(1/q)})$ for any strong Insdel RLDC with locality q.

▶ **Theorem 3.** *Let $C\colon \{0,1\}^n \to \{0,1\}^m$ be a non-adaptive strong $(q, \delta, 1/2 + \beta, \rho)$-Insdel RLDC where $\beta > 0$. Then for every $q \geq 2$ there is a constant $c_1 = c_1(q, \delta, \beta, \rho)$ such that*

$$m = \exp\left(c_1 \cdot n^{\Omega_\rho(\beta^2/q)}\right).$$

Furthermore, the same bound holds even if C does not have perfect completeness. If C has an adaptive decoder, the same bound holds with β replaced by $\beta/2^{q-1}$. Formally, there exists

a constant $c_2 = c_1(q, \delta, \beta/2^{q-1}, \rho)$ *such that*

$$m = \exp\left(c_2 \cdot n^{\Omega_\rho(\beta^2/(q2^{2q}))}\right).$$

Our reduction shown in the proof of Theorem 2, together with the impossibility results of standard *linear* or *affine* 2-query Insdel LDCs from [14] show a further impossibility result for linear and for affine 2-query Insdel RLDCs. A linear code of length m is defined over a finite field $\mathbb{F}$ and it is a linear subspace of the vector space $\mathbb{F}^m$, while an affine code is an affine subspace of $\mathbb{F}^m$.

We then consider *weak* Insdel RLDCs that only satisfy the first two properties (perfect completeness and relaxed decoding). In contrast with Theorem 3, we construct weak Insdel RLDCs with constant locality $q = O(1)$ and length $m = n^{1+\gamma}$ for some constant $\gamma \in (0, 1)$. To the best of our knowledge, this is the first positive result in the constant-query regime and the Insdel setting. However, the existence of a constant-query standard Insdel LDC (or even a constant-query strong Insdel RLDC) with any rate remains an open question. Finally, it is easy to see that our exponential lower bound for weak Hamming RLDCs with locality $q = 2$ still applies in the Insdel setting, since Insdel errors are more general than Hamming error. Thus, in the Insdel setting we discover the same "phase transition"-type phenomena as for Hamming RLDCs.

▶ **Theorem 4.** *For any* $\gamma > 0$ *and* $\varepsilon \in (0, 1/2)$*, there exist constants* $\delta \in (0, 1/2)$ *and* $q = q(\delta, \varepsilon, \gamma)$*, and non-adaptive weak* $(q, \delta, 1/2+\varepsilon)$*-Insdel RLDCs* $C\colon \{0,1\}^n \to \{0,1\}^m$ *with* $m = O(n^{1+\gamma})$.

We remark that in the Hamming setting, [7] shows that the first two properties of Definition 1 imply the third property for codes with constant query complexity and which can withstand a constant fraction of errors. Our results demonstrate that, in general, unlike in the Hamming case, the first two properties do not imply the third property for Insdel RLDCs from Definition 1. Indeed, while for strong Insdel RLDCs we have $m = \exp(n^{\Omega(1/q)})$ for codes of locality q, there exists $q = O(1)$ for which we have constructions of weak Insdel RLDCs with $m = n^{1+\gamma}$. This observation suggests that there are significant differences between Hamming RLDCs and Insdel RLDCs.

We note that our construction of weak Insdel RLDCs can be modified to obtain strong Insdel Relaxed Locally Correctable Codes (Insdel RLCCs). Informally, an Insdel RLCC is a code for which codeword entries can be decoded to the correct value or $\perp$ with high probability, even in the presence of insdel errors (see the full version for a formal definition of RLCC). We have the following corollary.

▶ **Corollary 5.** *For any* $\gamma > 0$ *and* $\varepsilon \in (0, 1/2)$*, there exist constants* $\delta \in (0, 1/2)$ *and* $q = q(\delta, \varepsilon, \gamma)$*, and non-adaptive strong* $(q, \delta, 1/2+\varepsilon, 1/2)$*-Insdel RLCCs* $C\colon \{0,1\}^n \to \{0,1\}^m$ *with* $m = O(n^{1+\gamma})$.

1.2 Overview of techniques

1.2.1 Exponential Lower Bound for Weak Hamming RLDCs with $q = 2$

To simplify the presentation, we assume a non-adaptive decoder in this overview. While the exact same arguments do not directly apply to adaptive decoders[2], with a bit more care they can be adapted to work in those settings.

[2] For standard LDCs Katz and Trevisan [55] observed that an adaptive decoder could be converted into a non-adaptive decoder by randomly guessing the output y_j of the first query j to learn the second query

At a high level we prove our lower bound by transforming any non-adaptive 2-query weak Hamming RLDC for messages of length n and δ fraction of errors into a standard 2-query Hamming LDC for messages of length $n' = \Omega(n)$, with slightly reduced error tolerance of $\delta/2$. Kerenidis and de Wolf [56] proved that any 2-query Hamming LDC for messages of length n must have codeword length $m = \exp(\Omega(n))$. Combining this result with our transformation, it immediately follows that any 2-query weak Hamming RLDC must also have codeword length $m = \exp(\Omega(n))$. While our transformation does not need the third property (success rate) of a strong RLDC, we crucially rely on the property of *perfect completeness*, and that the decoder only makes $q = 2$ queries.

Let $C\colon \{0,1\}^n \to \{0,1\}^m$ be a weak $(2, \delta, 1/2 + \varepsilon)$-RLDC. For simplicity (and without loss of generality), let us assume the decoder Dec works as follows. For message x and input $i \in [n]$, the decoder non-adaptively makes 2 random queries $j, k \in [m]$, and outputs $f^i_{j,k}(y_j, y_k) \in \{0, 1, \perp\}$, where y_j, y_k are answers to the queries from a received word y, and $f^i_{j,k}\colon \{0,1\}^2 \to \{0,1,\perp\}$ is a deterministic function. When there is no error, we have $y_j = C(x)_j$ and $y_k = C(x)_k$.

We present the main ideas below, and refer the readers to Section 4 for full details.

1.2.1.1 Fixable codeword bits

The starting point of our proof is to take a closer look at those functions $f^i_{j,k}$ with $\perp$ entries in their truth tables. It turns out that when $f^i_{j,k}$ has at least one $\perp$ entry in the truth table, $C(x)_j$ can be fixed to a constant by setting either $x_i = 0$ or $x_i = 1$, and same for $C(x)_k$. To see this, note that the property of perfect completeness forces $f^i_{j,k}$ to be 0 or 1 whenever $x_i = 0$ or $x_i = 1$ and there is no error. Thus if neither $x_i = 0$ nor $x_i = 1$ fixes $C(x)_j$, then there must be two entries of 0 and two entries of 1 in the truth table of $f^i_{j,k}$, which leaves no space for $\perp$ (see Claim 13). Thus, when there is at least one $\perp$ entry in the truth table of $f^i_{j,k}$, we say that $C(x)_j$ and $C(x)_k$ are *fixable* by x_i.

This motivates the definition of the set S_i, which contains all indices $j \in [m]$ such that the codeword bits $C(x)_j$ are fixable by x_i; and the definition of T_j, the set of all indices $i \in [n]$ such that $C(x)_j$ is fixable by the message bits x_i. It is also natural to pay special attention to queries j, k that are not both contained in S_i, since in this case the function $f^i_{j,k}$ never outputs $\perp$.

1.2.1.2 The query structure

In general, a query set $\{j, k\}$ falls into one of the following three cases: (1) both j, k lie inside S_i; (2) both j, k lie outside of S_i; (3) one of them lies inside S_i and the other lies outside of S_i. It turns out that case (3) essentially never occurs for a decoder with perfect completeness. The reason is that when, say, $j \in S_i$ and $k \notin S_i$, one can effectively pin down every entry in the truth table of $f^i_{j,k}$ by using the perfect completeness property, and observe that the output of $f^i_{j,k}$ does not depend on y_k at all (see Claim 14). Thus in this case we can equivalently view the decoder as only querying y_j where $j \in S_i$, which leads us back to case

---

k. Now we non-adaptively query the received codeword for both y_j and y_k. If our guess for y_j was correct then we continue simulating the adaptive decoder. Otherwise, we simply guess the output x_i. If the adaptive decoder succeeds with probability at least $p \geq 1/2 + \epsilon$ then the non-adaptive decoder succeeds with probability $p' \geq 1/4 + p/2 \geq 1/2 + \epsilon/2$. Unfortunately, this reduction does not preserve perfect completeness as required by our proofs for relaxed 2-query Hamming RLDCs i.e., if $p = 1$ then $p' = 3/4$.

(1). In what follows, we denote by E_1 the event that case (1) occurs, and by E_2 the event that case (2) occurs.

1.2.1.3 The transformation by polarizing conditional success probabilities

We now give a high level description of our transformation from a weak RLDC to a standard LDC. Let y be a string which contains at most $\delta m/2$ errors from the codeword $C(x)$. We have established that the success probability of the weak RLDC decoder on y is an average of two conditional probabilities

$$\Pr[\mathsf{Dec}(i,y) \in \{x_i, \perp\}] = p_1 \cdot \Pr[\mathsf{Dec}(i,y) \in \{x_i, \perp\} \mid E_1] + p_2 \cdot \Pr[\mathsf{Dec}(i,y) \in \{x_i, \perp\} \mid E_2],$$

where $p_1 = \Pr[E_1]$ and $p_2 = \Pr[E_2]$. Let us assume for the moment that S_i has a small size, e.g., $|S_i| \leq \delta m/2$. The idea in this step is to introduce additional errors to the S_i-portion of y, in a way that drops the conditional success probability $\Pr[\mathsf{Dec}(i,y) \in \{x_i, \perp\} \mid E_1]$ to 0 (see Lemma 15). In particular, we modify the bits in S_i to make it consistent with the encoding of any message $\hat{x}$ with $\hat{x}_i = 1 - x_i$. Perfect completeness thus forces the decoder to output $1 - x_i$ conditioned on E_1. Note that we have introduced at most $\delta m/2 + |S_i| \leq \delta m$ errors in total, meaning that the decoder should still have an overall success probability of $1/2 + \varepsilon$. Furthermore, now the conditional probability $\Pr[\mathsf{Dec}(i,y) \in \{x_i, \perp\} \mid E_2]$ takes all credits for the overall success probability. Combined with the observation that Dec never outputs $\perp$ given E_2, this suggests the following natural way to decode x_i in the sense of a standard LDC: sample queries j, k according to the conditional probability given E_2 (i.e., both j, k lie outside S_i) and output $f^i_{j,k}(y_j, y_k)$. This gives a decoding algorithm for standard LDC, with success probability $1/2 + \varepsilon$ and error tolerance $\delta m/2$ (see Lemma 16), modulo the assumption that $|S_i| \leq \delta m/2$.

1.2.1.4 Upper bounding $|S_i|$

The final piece in our transformation from weak RLDC to standard LDC is to address the assumption that $|S_i| \leq \delta m/2$. This turns out to be not true in general, but it would still suffice to prove that $|S_i| \leq \delta m/2$ for $n' = \Omega(n)$ of the message bits i. If we could show that $|T_j|$ is small for most $j \in [m]$, then a double counting argument shows that $|S_i|$ is small for most $i \in [n]$. Unfortunately, if we had $C(x)_j = \bigwedge_{i=1}^n x_i$ for $m/2$ of the codeword bits j then we also have $|T_j| = n$ for $m/2$ codeword bits and $|S_i| \geq m/2 \geq \delta m/2$ for all message bits $i \in [n]$. We address this challenge by first arguing that any weak RLDC for n-bit messages can be transformed into another weak RLDC for $\Omega(n)$-bit messages for which we have $|T_j| \leq 3\ln(8/\delta)$ for all but $\delta m/4$ codeword bits. The transformation works by fixing some of the message bits and then eliminating codeword bits that are fixed to constants. Intuitively, if some $C(x)_j$ is fixable by many message bits, it will have very low entropy (e.g., $C(x)_j$ is the AND of many message bits) and hence contain very little information and can (likely) be eliminated. We make this intuition rigorous through the idea of random restriction: for each $i \in [n]$, we fix $x_i = 0$, $x_i = 1$, or leave x_i free, each with probability $1/3$. The probability that $C(x)_j$ is not fixed to a constant is at most $(1 - 1/3)^{|T_j|} \leq \delta/8$, provided that $|T_j| \geq 3\ln(8/\delta)$. After eliminating codeword bits that are fixed to constants, we show that with probability at least $1/2$ at most $\delta m/4$ codeword bits $C(x)_j$ with $|T_j| \geq 3\ln(8/\delta)$ survived[3]. Note that with high probability the random restriction leaves at least $n/6$ message

[3] We are oversimplifying a bit for ease of presentation. In particular, the random restriction process may cause a codeword bit $C(x)_j$ to be fixable by a new message bit x_i that did not belong to T_j before the

bits free. Thus, there must exist a restriction which leaves at least $n/6$ message bits free ensuring that $|T_j| \geq 3\ln(8/\delta)$ for at most $\delta m/4$ of the remaining codeword bits $C(x)_j$. We can now apply the double counting argument to conclude that $|S_i| \leq \delta m/2$ for $\Omega(n)$ message bits, completing the transformation.

1.2.1.5 Adaptive decoders

For possibly adaptive decoders, we are going to follow the same proof strategy. The new idea and main difference is that we focus on the first query made by the decoder, which is always non-adaptive. We manage to show that the first query determines a similar query structure, which is the key to the transformation to a standard LDC. More details can be found in Section 4.2.

1.2.2 Lower Bounds for Strong Insdel RLDCs

We recall that a strong Insdel RLDC C is a weak Insdel RLDC which satisfies an additional property: for every $x \in \{0,1\}^n$ and $y \in \{0,1\}^{m'}$ such that $\mathsf{ED}(C(x), y) \leq \delta \cdot 2m$, there exists a set $I_y \subseteq [n]$ of size $|I_y| \geq \rho n$ such that for every $i \in I_y$, we have $\Pr[\mathsf{Dec}(i, y) = x_i] \geq \alpha$. In other words, for ρ-fraction of the message bits, the decoder can correctly recover them with high probability, just like in a standard Insdel LDC. Towards obtaining a lower bound on the codeword length m, a natural idea would be to view C as a standard Insdel LDC just for that ρ-fraction of message bits, and then apply the exponential lower bound for standard Insdel LDCs from [14]. This idea would succeed if the message bits correctly decoded with high probability were the same for all potential corrupted codewords y. However, it could be the case that $i \in I_y$ for some strings y, whereas $i \notin I_{y'}$ for other strings y'. Indeed, allowing the set I_y to depend on y is the main reason why very short constant-query Hamming RLDCs exist.

We further develop this observation to obtain our lower bound. We use an averaging argument to show the existence of a *corruption-independent* set I of message bits with $|I| = \Omega(n)$, which the decoder can recover with high probability. To this end, we need to open the "black box" of the lower bound result of Blocki et al. [14]. The proof in [14] starts by constructing an error distribution $\mathcal{E}$ with several nice properties, and deduce the exponential lower bound based solely on the fact that the Insdel LDC should, on average (i.e., for a uniformly random message x), correctly recover each bit with high probability under $\mathcal{E}$. One of the nice properties of $\mathcal{E}$ is that it is oblivious to the decoding algorithm Dec. Therefore, it makes sense to consider the average success rate against $\mathcal{E}$, i.e., $\Pr[\mathsf{Dec}(i, y) = x_i]$, where $i \in [n]$ is a uniformly random index, $x \in \{0,1\}^n$ is a uniformly random string, and y is a random string obtained by applying $\mathcal{E}$ to $C(x)$. By replacing $\perp$ with a uniformly random bit in the output of Dec, the average success rate is at least $\rho\alpha + (1-\rho)\alpha/2 = (1+\rho)\alpha/2$, since there is a ρ-fraction of indices for which Dec can correctly recover with probability α, and for the remaining $(1-\rho)$-fraction of indices the random guess provides an additional success rate of at least $\alpha/2$. Assuming α is sufficiently close to 1, which we can achieve by repeating the queries independently for a constant number of times and doing something similar to a majority vote, the average success rate against $\mathcal{E}$ is strictly above $1/2$. Therefore, there exist a constant fraction of indices for which the success rate against $\mathcal{E}$ is still strictly above $1/2$,

restriction – We thank an anonymous reviewer for pointing this out to us. Nevertheless, for our purpose it is sufficient to eliminate codeword bits that initially have a large $|T_j|$. See the formal proof for more details.

and the number of queries remains a constant. This is sufficient for the purpose of applying the argument in [14] to get an exponential lower bound.

1.2.3 Constant-Query Weak Insdel RLDC

Our construction of a constant query weak Insdel RLDC uses code concatenation and two building blocks: a weak Hamming RLDC (as the outer code) with constant query complexity, constant error-tolerance, and codeword length $k = O(n^{1+\gamma})$ for any $\gamma > 0$ [7], and the Schulman-Zuckerman [69] (from now on denoted by SZ) Insdel codes[4] (as the inner code). We let $C_{\text{out}} \colon \{0,1\}^n \to \{0,1\}^k$ and $C_{\text{in}} \colon [k] \times \{0,1\} \to \{0,1\}^t$ denote the outer and inner codes, respectively. Our final concatenation code C will have codewords in $\{0,1\}^m$ for some m (to be determined shortly), will have constant query complexity, and will tolerate a constant fraction of Insdel errors.

1.2.3.1 Code construction

Given a message $x \in \{0,1\}^n$, we first apply the outer code to obtain a Hamming codeword $y = y_1 \circ \cdots \circ y_k = C_{\text{out}}(x)$ of length k, where each $y_i \in \{0,1\}$ denotes a single bit of the codeword. Then for each index i, we compute $c_i = C_{\text{in}}(i, y_i) \in \{0,1\}^t$ as the encoding of the message (i, y_i) via the inner code. Finally, we output the codeword $C(x) \coloneqq c_1 \circ 0^t \circ c_2 \circ \cdots \circ 0^t \circ c_k$, where 0^t denotes a string of t zeros (which we later refer to as a buffer). Note that the inner code is a constant-rate code, i.e., $t = O(\log(k))$, and has constant error-tolerance $\delta_{\text{in}} \in (0, 1/2)$. Thus, the final codeword has length $m \coloneqq (2t-1)k = O(k \log(k))$ bits. For any constant $\gamma > 0$ we have a constant query outer code with length $k = O(n^{1+\gamma})$. Plugging this into our construction we have codeword length $m = O(n^{1+\gamma} \log n)$ which is $O(n^{1+\gamma'})$ for any constant $\gamma' > \gamma$.

1.2.3.2 Decoding algorithm: intuition and challenges

Intuitively, our relaxed decoder will simulate the outer decoder. When the outer decoder requests y_i, the natural approach would be to find and decode the block c_i to obtain (i, y_i). There are two challenges in this approach. First, if there were insertions or deletions, then we do not know where the block c_i is located; moreover, searching for this block can potentially blow-up the query complexity by a multiplicative polylog(m) factor [13, 68]. Second, even if we knew where c_i were located, because $t = O(\log k)$ and we want the decoder to have constant locality, we cannot afford to recover the entire block c_i.

We address the first challenge by attempting to locate block c_i under the optimistic assumption that there are no corruptions. If we detect any corruptions, then we may immediately abort and output $\perp$ since our goal is only to obtain a weak Insdel RLDC. Assuming that there were no corruptions, we know exactly where the block c_i is located, and we know that c_i can only take on two possible values: it is either the inner encoding of $(i, 0)$ or the inner encoding of $(i, 1)$. If we find anything inconsistent with the inner encoding of either $(i, 0)$ or $(i, 1)$, then we can immediately output $\perp$.

Checking consistency with the inner encodings of $(i, 0)$ and $(i, 1)$ is exactly how we address the second challenge. In place of reading the entire block c_i, we instead only need to determine whether (1) c_i is (close to) the inner encoding of $(i, 0)$, (2) c_i is (close to) the inner encoding of $(i, 1)$, or (3) c_i is not close to either string. In either case (1) or case

[4] In particular, these are classical/non-local codes.

(2), we simply output the appropriate bit, and in case (3), we simply output $\perp$. Thus, our Insdel RLDC decoder simulates the outer decoder. Whenever the outer decoder request y_i, we determine the expected location for c_i, randomly sub-sample a constant number of indices from this block and compare with the inner encodings of $(i, 0)$ and $(i, 1)$ at the corresponding indices. To ensure perfect completeness, we always ensure that *at least one* of the sub-sampled indices is for a bit where the inner encodings of $(i, 0)$ and $(i, 1)$ differ. If there are no corruptions, then whenever the simulated outer decoder requests y_i we will always respond with the correct bit. Perfect completeness of our Insdel RLDC now follows immediately from the perfect completeness of the outer decoder. Choosing a constant number of indices to sub-sample ensures that the locality of our weak Insdel RLDC decoder is a constant multiplicative factor larger than the outer decoder, which gives our Insdel RLDC decoder constant locality overall.

1.2.3.3 Analysis of the decoding algorithm

The main technical challenge is proving that our Insdel RLDC still satisfies the second condition of Definition 1, when the received word is not a correct encoding of the message x. Recall that $c_i = C_{\text{in}}(i, y_i)$, and suppose $\widetilde{c}_i \neq c_i$ is the block of the received word that we are going to check for consistency with the inner encodings of $(i, 0)$ and $(i, 1)$. Then, the analysis of our decoder falls into three cases. In the first case, if $\widetilde{c}_i$ is not too corrupted (i.e., $\mathsf{ED}(\widetilde{c}_i, c_i)$ is not too large), then we can argue that the decoder outputs the correct bit y_i or $\perp$ with good probability. In the second case, if $\widetilde{c}_i$ has high edit distance from both $C_{\text{in}}(i, 0)$ and $C_{\text{in}}(i, 1)$, then we can argue that the decoder outputs $\perp$ with good probability. The third case is the most difficult case, which we describe as "dangerous". We say that the block $\widetilde{c}_i$ is *dangerous* if the edit distance between $\widetilde{c}_i$ and $C_{\text{in}}(i, 1 - y_i)$ is not too large; i.e., $\widetilde{c}_i$ is close to the encoding of the opposite bit $1 - y_i$.

The key insight to our decoding algorithm is that as long as the number of dangerous blocks $\widetilde{c}_i$ is upper bounded, then we can argue the overall probability that our decoder outputs y_i or $\perp$ satisfies the relaxed decoding condition of Definition 1. Intuitively, we can we think of our weak Insdel RLDC decoder as running the outer decoder on a string $\widetilde{y} = \widetilde{y}_1 \circ \ldots \circ \widetilde{y}_k$, where each $\widetilde{y}_i \in \{0, 1, \perp\}$ and the outer decoder has been modified to output $\perp$ whenever it queries for y_i and receives $\perp$. Observe that if δ_{out} is the error-tolerance of the outer decoder, then as long as the set $\left|\{i \ : \ \widetilde{y}_i \neq \perp \wedge \widetilde{y}_i \neq y_i\}\right| \leq \delta_{\text{out}} k$, the modified outer decoder, on input $j \in [n]$, will output either the correct value x_j or $\perp$ with high probability (for appropriate choices of parameters). Intuitively, if a block is "dangerous" then we can view $\widetilde{y}_i = 1 - y_i$, and otherwise we have $\widetilde{y}_i \in \{y_i, \perp\}$ with reasonably high probability. Thus, as long as the number of "dangerous" block is at most $\delta_{\text{out}} k/2$, then our relaxed Insdel decoder will satisfy the second property of Definition 1 and output either x_j or $\perp$ with high probability for any $j \in [n]$.

1.2.3.4 Upper bounding the number of dangerous blocks

To upper bound the number of "dangerous" blocks we utilize a matching argument based on the longest common sub-sequence (LCS) between the original codeword and the received (corrupted) word. Our matching argument utilizes a key feature of the SZ Insdel code. In particular, the Hamming weight (i.e., number of non-zero symbols) of every substring c' of an SZ codeword is at least $\lfloor |c'|/2 \rfloor$. This ensures that the buffers 0^t cannot be matched with large portions of any SZ codeword. We additionally leverage a key lemma (full version, Lemma 9) which states that the edit distance between the codeword $C_{\text{in}}(i, 1 - y_i)$ and *any*

substring of length less than $2t$ of the uncorrupted codeword $C(x)$ has relative edit distance at least $\delta_{\text{in}}/2$. We use these two properties, along with key facts about the LCS matching, to yield an upper bound on the number of dangerous blocks, completing the analysis of our decoder.

1.2.3.5 Extension to relaxed locally correctable codes for insdel errors

Our construction also yields a strong Insdel Relaxed Locally Correctable Code (RLCC) with constant locality if the outer code is a weak Hamming RLCC. First, observe that bits of the codeword corresponding to the 0^t buffers are very easy to predict without even making any queries to the corrupted codeword. Thus, if we are asked to recover the j'th bit of the codeword and j corresponds to a buffer 0^t, we can simply return 0 without making any queries to the received word. Otherwise, if we are asked to recover the j'th bit of the codeword and j corresponds to block c_i, we can simulate the Hamming RLCC decoder (as above) on input i to obtain y_i (or $\perp$). Assuming that $y_i \in \{0,1\}$, we can compute the corresponding SZ encoding of (i, y_i) and obtain the original value of the block c_i and then recover the j'th bit of the original codeword. The analysis of the RLCC decoder is analogous to the RLDC decoder. See Section 6 in the full version for details on both our weak Insdel RLDC and strong Insdel RLCC constructions.

▶ Remark 6. The "adaptiveness" of our constructed Insdel RLDC/RLCC decoder is identical to that of the outer Hamming RLDC/RLCC decoder. In particular, the weak Hamming RLDC of Ben-Sasson et al. [7] has a non-adaptive decoder, making our final decoder non-adaptive as well. Similarly, we use a weak Hamming RLCC due to Asadi and Shinkar [3] for our Insdel LCC, which is also a non-adaptive decoder.

2 Open Questions

Exact "phase-transition" thresholds

Our results show that both in the Hamming and Insdel setting there is a constant q such that every q-query RLDC requires super-polynomial codeword length, while there exists a $(q+1)$-query RLDC of polynomial codeword length. Finding the precise q remains an intriguing open question. Further, a more refined understanding of codeword length for RLDCs making $3, 4, 5$ queries is another important question, which has lead to much progress in the understanding of the LDC variants.

Constant-query strong Insdel RLDCs/RLCCs

While we do construct the first weak RLDCs in the Insdel setting, the drawback of our constructions is the fact that our codes do not satisfy the third property of Definition 1. Building strong Insdel RLDCs remains an open question. We note that our lower bounds imply that for a constant number of queries, such codes (if they exist) must have exponential codeword length.

Applications of local Insdel codes

As previously mentioned, Hamming LDCs/RLDCs have so far found many applications such as private information retrieval, probabilistically checkable proofs, self-correction, fault-tolerant circuits, hardness amplification, and data structures. Are there analogous or new applications of the Insdel variants in the broader computing area?

Lower bounds for Hamming RLDCs/LDCs

Our 2-query lower bound for Hamming RLDCs crucially uses the perfect completeness property of the decoder. An immediate question is whether the bound still holds if we allow the decoder to have imperfect completeness. We also note that the argument in our exponential lower bounds for 2-query Hamming RLDCs fail to hold for alphabets other than the binary alphabet, and we leave the extension to larger alphabet sizes as an open problem. Another related question is to understand if one can leverage perfect completeness and/or random restrictions to obtain improved lower bounds for $q \geq 3$-query standard Hamming LDCs. Perfect completeness has been explicitly used before to show exponential lower bounds for 2-query LCCs [11].

2.1 Further discussion about related work

Insdel codes

The study of error correcting codes for insertions and deletions was initiated by Levenstein [59]. While progress has been slow because constructing codes for insdel errors is strictly more challenging than for Hamming errors, strong interest in these codes lately has led to many exciting results [19, 21–25, 41–43, 45–49, 51, 61, 63, 69] (See also the excellent surveys of [50, 64, 66, 71]).

Insdel LDCs

[67] gave private-key constructions of LDCs with $m = \Theta(n)$ and locality polylog(n). [16] extended the construction from [67] to settings where the sender/decoder do not share randomness, but the adversarial channel is resource bounded. [12] applied the [13] compiler to the private key Hamming LDC of [67] (resp. resource bounded LDCs of [16]) to obtain private key Insdel LDCs (resp. resource bounded Insdel LDCs) with constant rate and polylog(n) locality.

Insdel LDCs have also been recently studied in *computationally bounded channels*, introduced in [60]. Such channels can perform a bounded number of adversarial errors, but do not have unlimited computational power as the general Hamming channels. Instead, such channels operate with bounded resources. As expected, in many such limited-resource settings one can construct codes with strictly better parameters than what can be done generally [31, 44, 65, 70]. LDCs in these channels under Hamming error were studied in [15, 16, 52–54, 67]. [12] applied the [13] compiler to the Hamming LDC of [16] to obtain a constant rate Insdel LDCs with polylog(n) locality for resource bounded channels. The work of [26] proposes the notion of locally decodable codes with randomized encoding, in both the Hamming and edit distance regimes, and in the setting where the channel is oblivious to the encoded message, or the encoder and decoder share randomness. For edit error they obtain codes with $m = O(n)$ or $m = n \log n$ and polylog(n) query complexity. However, even in settings with shared randomness or where the channel is oblivious or resource bounded, there are no known constructions of Insdel LDCs with constant locality.

Locality in the study of insdel codes was also considered in [49], which constructs explicit synchronization strings that can be locally decoded.

2.2 Organization

The remainder of the paper is organized as follows. We give general preliminaries and recall some prior results used in our results in Section 3. Due to space limit, we only present the

proof of Theorem 2 in Section 4. The readers are pointed to the full version for proofs of Theorem 3, Theorem 4 and Corollary 5.

3 Preliminaries

For natural number $n \in \mathbb{N}$, we let $[n] \coloneqq \{1, 2, \ldots, n\}$. We let "$\circ$" denote the standard string concatenation operation. For a string $x \in \{0,1\}^*$ of finite length, we let $|x|$ denote the length of x. For $i \in [|x|]$, we let $x[i]$ denote the i-th bit of x. Furthermore, for $I \subseteq [|x|]$, we let $x[I]$ denote the subsequence $x[i_1] \circ x[i_2] \circ \cdots \circ x[i_\ell]$, where $i_j \in I$ and $\ell = |I|$. For two strings $x, y \in \{0,1\}^n$ of length n, we let $\mathsf{HAM}(x,y)$ denote the *Hamming Distance* between x and y; i.e., $\mathsf{HAM}(x,y) \coloneqq \big|\{i \in [n] \colon x_i \neq y_i\}\big|$. Similarly, we let $\mathsf{ED}(x,y)$ denote the *Edit Distance* between x and y; i.e., $\mathsf{ED}(x,y)$ is the minimum number of insertions and deletions needed to transform string x into string y. We often discuss the *relative Hamming Distance* (resp., *relative Edit Distance*) between x and y, which is simply the Hamming Distance normalized by n, i.e., $\mathsf{HAM}(x,y)/n$ (resp., the Edit Distance normalized by $|x| + |y|$, i.e., $\mathsf{ED}(x,y)/(|x| + |y|)$). Finally, the *Hamming weight* of a string x is the number of non-zero entries of x, which we denote as $\mathsf{wt}(x) \coloneqq |\{i \in [|x|] \colon x_i \neq 0\}|$.

For completeness, we recall the definition of a classical locally decodable code, or just a *locally decodable code*.

▶ **Definition 7** (Locally Decodable Codes). *A (q, δ, α)-Locally Decodable Code $C \colon \Sigma^n \to \Sigma^m$ is a code for which there exists a randomized decoder that makes at most q queries to the received word y and satisfies the following property: for every $i \in [n]$, if y is such that* $\mathrm{dist}(y, C(x)) \leq \delta$ *for some unique $C(x)$, then the decoder, on input i, outputs x_i with probability $\geq \alpha$. Here, the randomness is taken over the random coins of the decoder, and* dist *is a normalized metric.*

If dist *is the relative Hamming distance, then we say that the code is a Hamming LDC; similarly, if* dist *is the relative edit distance, then we say that the code is an Insdel LDC.*

We recall the general 2-query Hamming LDC lower bound [6, 56].

▶ **Theorem 8** ([6, 56]). *For constants $\delta, \varepsilon \in (0, 1/2)$ there exists a constant $c = c(\delta, \varepsilon) \in (0,1)$ such that if $C \colon \{0,1\}^n \to \{0,1\}^m$ is a $(2, \delta, 1/2 + \varepsilon)$ Hamming LDC then $m \geq 2^{cn-1}$.*

In our weak Insdel RLDC construction, we utilize a weak Hamming RLDC due to [7].

▶ **Lemma 9** ([7]). *For constants $\varepsilon, \delta \in (0, 1/2)$ and $\gamma \in (0,1)$, there exists a constant $q = O_{\delta,\varepsilon}(1/\gamma^2)$ and a weak $(q, \delta, 1/2 + \varepsilon)$-Hamming RLDC $C \colon \{0,1\}^n \to \{0,1\}^m$ with $m = O(n^{1+\gamma})$. Moreover, the decoder of this code is non-adaptive.*

Our construction additionally utilizes the well-known Schulman-Zuckerman Insdel codes [69].

▶ **Lemma 10** (Schulman-Zuckerman (SZ) Code [69]). *There exists constants $\beta \geq 1$ and $\delta > 0$ such that for large enough values of $t > 0$, there exists a code $C \colon \{0,1\}^t \to \{0,1\}^{\beta t}$ capable of decoding from δ-fraction of Insdel errors and the additional property that for every $x \in \{0,1\}^t$ and $y = C(x)$, every substring y' of y with length at least 2 has Hamming weight $\geq \lfloor |y'|/2 \rfloor$.*

Our strong Insdel RLCC construction relies on a weak Hamming RLCC. We utilize the following weak Hamming RLCC implicit in [3].

▶ **Lemma 11** (Implied by Theorem 1 of [3]). *For every sufficiently large $q \in \mathbb{N}$ and $\varepsilon \in (0, 1/2)$, there is a constant δ such that there exists a weak $(q, \delta, 1/2 + \varepsilon)$-relaxed Hamming Locally Correctable Code $C \colon \{0,1\}^n \to \{0,1\}^m$ with $m = n^{1+O(1/q)}$. Moreover, the decoder of this code is non-adaptive.*

4 Lower Bounds for 2-Query Hamming RLDCs

We prove Theorem 2 in this section. As a reminder, a weak (q, δ, α)-RLDC satisfies the first two conditions in Definition 1, and non-adaptive means the decoder makes queries according to a distribution which is independent of the received string y. Here we are interested in the case $q = 2$ and $\alpha = 1/2 + \varepsilon$.

To avoid overloading first-time readers with heavy notations, we first present a proof of the lower bound for *non-adaptive* decoders, i.e., decoders with a query distribution independent of the received string. This proof will be easier to follow, while the crucial ideas behind it remain the same. The proof for the most general case is presented in the last subsection, with an emphasis on the nuances in dealing with adaptivity.

4.1 A Warm-up: the lower bound for non-adaptive decoders

In the following, we fix a relaxed decoder Dec for C. In this subsection, we assume that Dec is non-adaptive, and that it has the first two properties specified in Definition 1. To avoid technical details, we also assume Dec always makes exactly 2 queries (otherwise add dummy queries to make the query count exactly 2).

Given an index $i \in [n]$ and queries j, k made by $\mathsf{Dec}(i, \cdot)$, in the most general setting the output could be a random variable which depends on i and y_j, y_k, where y_j, y_k are the answers to queries j, k, respectively. An equivalent view is that the decoder picks a random function f according to some distribution and outputs $f(y_j, y_k)$. Let $\mathtt{DF}^i_{j,k}$ be the set of all decoding functions $f\colon \{0,1\}^2 \to \{0, 1, \perp\}$ which are selected by $\mathsf{Dec}(i, \cdot)$ with non-zero probability when querying j, k. We partition the queries into the following two sets

$$F^0_i \coloneqq \left\{\{j,k\} \subseteq [m] \colon \forall f \in \mathtt{DF}^i_{j,k} \text{ the truth table of } f \text{ contains no “}\perp\text{”}\right\},$$

$$F^{\geq 1}_i \coloneqq \left\{\{j,k\} \subseteq [m] \colon \exists f \in \mathtt{DF}^i_{j,k} \text{ the truth table of } f \text{ contains at least 1 “}\perp\text{”}\right\}.$$

Notations

Given a string $w \in \{0,1\}^m$ and a subset $S \subseteq [m]$, we denote $w[S] \coloneqq (w_i)_{i \in S} \in \{0,1\}^{|S|}$. Given a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, and $\sigma \in \{0,1\}$, we write $f \restriction_{x_i = \sigma}$ to denote the restriction of f to the domain $\left\{\mathbf{x} \in \{0,1\}^n : x_i = \sigma\right\}$. For a sequence of restrictions, we simply write $f \restriction_{(x_{j_1}, \ldots, x_{j_k}) = (\sigma_1, \ldots, \sigma_k)}$, or $f_{J|\sigma}$ where $J = [n] \setminus \{j_1, \ldots, j_k\}$ and $\sigma = (\sigma_1, \ldots, \sigma_k)$. Note that $f_{J|\sigma}$ is a Boolean function over the domain $\{0,1\}^J$.

We will identify the encoding function of C as a collection of m Boolean functions

$$\mathcal{C} \coloneqq \left\{C_1, \ldots, C_m \colon \forall j \in [m], C_j \colon \{0,1\}^n \to \{0,1\}\right\}.$$

Namely, $C(x) = (C_1(x), C_2(x), \ldots, C_m(x))$ for all $x \in \{0,1\}^n$.

For $j \in [m]$, we say C_j is *fixable* by x_i if at least one of the restrictions $C_j \restriction_{x_i = 0}$ and $C_j \restriction_{x_i = 1}$ is a constant function. Denote

$$S_i \coloneqq \left\{j \in [m] \colon C_j \text{ is fixable by } x_i\right\}, \quad T_j \coloneqq \left\{i \in [n] \colon C_j \text{ is fixable by } x_i\right\},$$

and $w_j \coloneqq |T_j|$. Let

$$W \coloneqq \left\{j \in [m] \colon w_j \geq 3\ln(8/\delta)\right\}.$$

For $i \in [n]$ define the sets $S_{i,+} \coloneqq S_i \cap W$, and $S_{i,-} \coloneqq S_i \cap \overline{W}$.

Let $J \subseteq [n]$ and $\rho \in \{0,1\}^{\overline{J}}$. A code $C\colon \{0,1\}^n \to \{0,1\}^m$ restricted to $\mathbf{x}_{\overline{J}} = \rho$, denoted by $C_{J|\rho}$, is specified by the following collection of Boolean functions

$$\mathcal{C}_{J|\rho} := \left\{ C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho} \colon j \in [m], C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho} \text{ is not a constant function} \right\}.$$

Namely, we restrict each function C_j in $\mathcal{C}$ to $\mathbf{x}_{\overline{J}} = \rho$, and eliminate those that have become constant functions. $C_{J|\rho}$ encodes n'-bit messages into m'-bit codewords, where $n' = |J|$ and $m' = \left|\mathcal{C}_{J|\rho}\right| \leq m$.

We note that the local decoder Dec for C can also be used as a local decoder for $C_{J|\rho}$, while preserving all the parameters. This is because, Dec never needs to really read a codeword bit which has become a constant function under the restriction $J|\rho$.

The lemma below will be useful later in the proof. It shows that a constant fraction of the message bits can be fixed so that most codeword bits C_j with large w_j become constants.

▶ **Lemma 12.** *There exist a set $J \subseteq [n]$ and assignments $\rho \in \{0,1\}^{\overline{J}}$ such that $|J| \geq n/6$, and $|W \setminus A| \leq \delta m/4$, where $A \subseteq W$ collects all codeword bits $j \in W$ such that $C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho}$ is a constant function.*

Proof. Let J be a random subset formed by selecting each $i \in [n]$ independently with probability $1/3$. For each $j \in \overline{J}$, set $\rho_j = 0$ or $\rho_j = 1$ with probability $1/2$. We have $\mathbb{E}[|J|] = n/3$, and hence the Chernoff bound shows that $|J| < n/6$ with probability $\exp(-\Omega(n))$. Furthermore, for each $j \in W$, $C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho}$ becomes a constant function except with probability $\delta/8$. This is because for each $i \in T_j$, $C_j \restriction_{x_i=0}$ or $C_j \restriction_{x_i=1}$ is a constant function, and either case happens with probability $1/3$. Therefore

$$\Pr\left[C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho} \text{ is not constant}\right] \leq \left(1 - \frac{1}{3}\right)^{|T_j|} < e^{-|T_j|/3} \leq \frac{\delta}{8},$$

where the last inequality is due to $w_j = |T_j| \geq 3\ln(8/\delta)$, since $j \in W$.

By linearity of expectation and Markov's inequality, we have

$$\begin{aligned}
&\Pr\left[\sum_{j\in W} \mathbf{1}\left\{C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho} \text{ is not constant}\right\} \geq \frac{\delta}{4}|W|\right] \\
\leq &\frac{\mathbb{E}\left[\sum_{j\in W} \mathbf{1}\left\{C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho} \text{ is not constant}\right\}\right]}{\delta|W|/4} \\
= &\frac{\sum_{j\in W} \Pr\left[C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho} \text{ is not constant}\right]}{\delta|W|/4} \\
\leq &\frac{\delta/8 \cdot |W|}{\delta|W|/4} \leq \frac{1}{2}.
\end{aligned}$$

Applying a union bound gives

$$\begin{aligned}
&\Pr\left[(|J| < n/6) \vee \left(\sum_{j\in W} \mathbf{1}\left\{C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho} \text{ is not constant}\right\} \geq \frac{\delta}{4}|W|\right)\right] \\
\leq &\exp\left(-\Omega(n)\right) + \frac{1}{2} < 1.
\end{aligned}$$

Finally, we can conclude that there exist $J \subseteq [n]$ and $\rho \in \{0,1\}^{\overline{J}}$ such that $|J| \geq n/6$, and $C_j \restriction_{\mathbf{x}_{\overline{J}}=\rho}$ becomes a constant function for all but $\delta/4$ fraction of $j \in W$. ◀

Let $J \subseteq [n]$ and $\rho \in \{0,1\}^{\overline{J}}$ be given by the Lemma 12, and consider the restricted code $C_{J|\rho}$. By rearranging the codeword bits, we may assume $J = [n']$ where $n' = |J| \geq n/6$.

Let $A \subseteq [m]$ be the set of codeword bits which get fixed to constants under $J|\rho$. We denote $W' \coloneqq W \setminus A$, $S'_i \coloneqq S_i \setminus A$, $S'_{i,-} \coloneqq S_{i,-} \setminus A$, and $S'_{i,+} \coloneqq S_{i,+} \setminus A$. Note that $|W'| = |W \setminus A| \leq \delta m/4$, and thus $|S'_{i,+}| = |S_{i,+} \cap W'| \leq \delta m/4$ for all $i \in [n']$. We emphasize that S'_i does not necessarily contain all codeword bits fixable by x_i in the restricted code $C_{J|\rho}$, as fixing some message bits may cause more codeword bits to be fixable by x_i.

We first show that the queries of C must have certain structures. The following claim characterizes the queries in $F_i^{\geq 1}$.

▷ Claim 13. Suppose $\{j,k\} \in F_i^{\geq 1}$. Then we must have $j,k \in S_i$.

Proof. Let $\{j,k\} \in F_i^{\geq 1}$. Suppose for the sake of contradiction that $j \notin S_i$. This implies there are partial assignments $\sigma_{00}, \sigma_{01}, \sigma_{10}, \sigma_{11} \in \{0,1\}^{n-1}$ such that

$$C_j\left(\mathbf{x}_{-i} = \sigma_{00}, x_i = 0\right) = 0, \quad C_j\left(\mathbf{x}_{-i} = \sigma_{01}, x_i = 1\right) = 0,$$
$$C_j\left(\mathbf{x}_{-i} = \sigma_{10}, x_i = 0\right) = 1, \quad C_j\left(\mathbf{x}_{-i} = \sigma_{11}, x_i = 1\right) = 1,$$

where $\mathbf{x}_{-i}$ is defined as $\left(x_t \colon t \in [n] \setminus \{i\}\right)$.

Let $C_{00}, C_{01}, C_{10}, C_{11}$ be encodings of the corresponding assignments mentioned above. Since the relaxed decoder has perfect completeness, when $\mathsf{Dec}(i,\cdot)$ is given access to C_{00} or C_{10} it must output $x_i = 0$. Note that the j-th bit is different in C_{00} and C_{10}. Similarly, when $\mathsf{Dec}(i,\cdot)$ is given access to C_{01} or C_{11} it must output $x_i = 1$. However, this already takes up 4 entries in the truth table of any decoding function $f \in \mathtt{DF}^i_{j,k}$, leaving no space for any "$\perp$" entry. This contradicts with the assumption $\{j,k\} \in F_i^{\geq 1}$. ◀

Here is another way to view Claim 13 which will be useful later: Suppose $\{j,k\}$ is a query set such that $j \notin S_i$ (or $k \notin S_i$), then $\{j,k\} \in F_i^0$. In other words, conditioned on the event that some query is not contained in S_i, the decoder never outputs $\perp$.

The following claim characterizes the queries in F_i^0.

▷ Claim 14. Suppose $\{j,k\} \in F_i^0$, and $j \in S_i$. Then one of the following three cases occur: (1) $k \in S_i$, (2) $C_j = x_i$, or (3) $C_j = \neg x_i$.

Proof. Since $j \in S_i$, we may, without loss of generality, assume that $C_j \restriction_{x_i=0}$ is a constant function. Let us further assume it is the constant-zero function. The proofs for the other cases are going to be similar.

Denote by $f(y_j, y_k)$ the function returned by $\mathsf{Dec}(i,\cdot)$ conditioned on reading $\{j,k\}$. Any function $f \in \mathtt{DF}^i_{j,k}$ takes values in $\{0,1\}$ since $\{j,k\} \in F_i^0$. Suppose case (1) does not occur, meaning that $C_k \restriction_{x_i=0}$ is not a constant function. Then there must be partial assignments $\sigma_{00}, \sigma_{01} \in \{0,1\}^{n-1}$ such that

$$C_k(x_i = 0, \mathbf{x}_{-i} = \sigma_{00}) = 0, \quad C_k(x_i = 0, \mathbf{x}_{-i} = \sigma_{01}) = 1.$$

Let C_{00} and C_{01} be the encodings of the corresponding assignments mentioned above. Due to perfect completeness of Dec, it must always output $x_i = 0$ when given access to C_{00} or C_{01}. That means $f(0,0) = f(0,1) = 0$.

Now we claim that $C_j \restriction_{x_i=1}$ must be the constant-one function. Otherwise there is a partial assignment $\sigma_{10} \in \{0,1\}^{n-1}$ such that

$$C_j(x_i = 1, \mathbf{x}_{-i} = \sigma_{10}) = 0.$$

Let C_{10} be the encoding of this assignment. On the one hand, due to perfect completeness $\mathsf{Dec}(i,\cdot)$ should always output $x_i = 1$ when given access to C_{10}. On the other hand, $\mathsf{Dec}(i,\cdot)$ outputs $f((C_{10})_j, 0) = f(0,0) = 0$. This contradiction shows that $C_j \restriction_{x_i=1}$ must be the constant-one function. Therefore $C_j = x_i$, i.e., case (2) occurs.

Similarly, when $C_j \restriction_{x_i=0}$ is the constant-one function, we can deduce that $C_j = \neg x_i$, i.e., case (3) occurs. ◀

We remark that Claim 13 and Claim 14 jointly show that for any query set $\{j,k\}$ made by $\mathsf{Dec}(i,\cdot)$ there are 2 essentially different cases: (1) both j,k lie inside S_i, and (2) both j,k lie outside S_i. The case $j \in S_i, k \notin S_i$ ($k \in S_i, j \notin S_i$, resp.) means that k (j, resp.) is a dummy query which is not used for decoding. Furthermore, conditioned on case (2), the decoder never outputs $\perp$.

Another important observation is that all properties of the decoder discussed above hold for the restricted code $C_{J|\rho}$, with S_i replaced by S'_i. This is because $C_{J|\rho}$ uses essentially the same decoder, except that it does not actually query any codeword bit which became a constant.

For a subset $S \subseteq [m]$, we say "$\mathsf{Dec}(i,\cdot)$ reads S" if the event "$j \in S$ and $k \in S$" occurs where $j,k \in [m]$ are the queries made by $\mathsf{Dec}(i,\cdot)$. The following lemma says that conditioned on $\mathsf{Dec}(i,\cdot)$ reads some subset S, there is a way of modifying the bits in S that flips the output of the decoder.

▶ **Lemma 15.** *Let $S \subseteq [m]$ be a subset such that $\Pr[\mathsf{Dec}(i,\cdot) \text{ reads } S] > 0$. Then for any string $s \in \{0,1\}^m$ and any bit $b \in \{0,1\}$, there exists a string $z \in \{0,1\}^m$ such that $z[[m] \setminus S] = s[[m] \setminus S]$, and*

$$\Pr\left[\mathsf{Dec}(i,z) = 1-b \mid \mathsf{Dec}(i,\cdot) \text{ reads } S\right] = 1.$$

Proof. Let $x \in \{0,1\}^n$ be a string with $x_i = 1-b$. Let $z \in \{0,1\}^m$ be the string satisfying

$$z[S] = C(x)[S], \quad z[[m] \setminus S] = s[[m] \setminus S].$$

Since Dec has perfect completeness, we have

$$1 = \Pr\left[\mathsf{Dec}(i, C(x)) = x_i \mid \mathsf{Dec}(i,\cdot) \text{ reads } S\right] = \Pr\left[\mathsf{Dec}(i,z) = 1-b \mid \mathsf{Dec}(i,\cdot) \text{ reads } S\right].$$

◀

The next lemma is a key step in our proof. It roughly says that there is a local decoder for x_i in the standard sense as long as the size of S_i is not too large.

▶ **Lemma 16.** *Suppose $i \in [n]$ is such that $|S_i| \le \delta m/2$. Then there is a $(2, \delta/2, 1/2+\varepsilon)$-local decoder D_i for i. In other words, for any $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$ such that $\mathsf{HAM}(C(x), y) \le \delta m/2$, we have*

$$\Pr\left[D_i(y) = x_i\right] \ge \frac{1}{2} + \varepsilon,$$

and D_i makes at most 2 queries into y.

Proof. Let $i \in [n]$ be such that $|S_i| \le \delta m/2$. The local decoder D_i works as follows. Given $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$ such that $\mathsf{HAM}(C(x), y) \le \delta m/2$, D_i obtains a query set Q according to the query distribution of $\mathsf{Dec}(i,\cdot)$ conditioned on $Q \subseteq [m] \setminus S_i$. Then D_i finishes by outputting the result returned by $\mathsf{Dec}(i,\cdot)$.

Denote by E_i the event "$\mathsf{Dec}(i,\cdot)$ reads $[m]\setminus S_i$", i.e., both two queries made by $\mathsf{Dec}(i,\cdot)$ lie outside S_i. In order for the conditional distribution to be well-defined, we need to argue that E_i occurs with non-zero probability. Suppose this is not the case, meaning that $Q \cap S_i \neq \varnothing$ for all possible query set Q. Let $z \in \{0,1\}^m$ be the string obtained by applying Lemma 15 with $S = S_i$, $s = C(x)$ and $b = x_i$. Claim 13 and Claim 14 jointly show that either $Q \subseteq S_i$, or the decoder's output does not depend on the answers to queries in $Q \setminus S_i$. In any case, the output of $\mathsf{Dec}(i,z)$ depends only on $z[S_i]$. However, by the choice of z we now have a contradiction since

$$\frac{1}{2} + \varepsilon \leq \Pr\left[\mathsf{Dec}(i,z) \in \{x_i, \perp\}\right] = \Pr\left[\mathsf{Dec}(i,z) \in \{x_i,\perp\} \mid \mathsf{Dec}(i,\cdot) \text{ reads } S_i\right] = 0,$$

where the first inequality is due to $\mathsf{HAM}(C(x),z) \leq |S_i| < \delta m$ and the relaxed decoding property of Dec.

By definition of D_i, it makes at most 2 queries into y. Its success rate is given by

$$\Pr[D_i(y) = x_i] = \Pr[\mathsf{Dec}(i,y) = x_i \mid E_i].$$

Therefore it remains to show that

$$\Pr\left[\mathsf{Dec}(i,y) = x_i \mid E_i\right] \geq \frac{1}{2} + \varepsilon.$$

Let z be the string obtained by applying Lemma 15 with $S = S_i$, $s = y$ and $b = x_i$. From previous discussions we see that conditioned on $\overline{E_i}$ (i.e., the event E_i does not occur), the output of $\mathsf{Dec}(i,z)$ only depends on $z[S_i]$. Therefore

$$\Pr\left[\mathsf{Dec}(i,z) \in \{x_i,\perp\} \mid \overline{E_i}\right] = 1 - \Pr\left[\mathsf{Dec}(i,z) = 1 - x_i \mid \overline{E_i}\right] = 0. \tag{1}$$

We also have that z is close to $C(x)$ since

$$\mathsf{HAM}(z,C(x)) \leq \mathsf{HAM}(z,y) + \mathsf{HAM}(y,C(x)) \leq |S_i| + \delta m/2 \leq \delta m.$$

Thus, the relaxed decoding property of Dec gives

$$\Pr\left[\mathsf{Dec}(i,z) \in \{x_i,\perp\}\right] \geq \frac{1}{2} + \varepsilon.$$

On the other hand, we also have

$$\begin{aligned}
&\Pr\left[\mathsf{Dec}(i,z) \in \{x_i,\perp\}\right] \\
=&\Pr\left[\mathsf{Dec}(i,z) \in \{x_i,\perp\} \mid \overline{E_i}\right] \cdot \Pr\left[\overline{E_i}\right] + \Pr\left[\mathsf{Dec}(i,z) \in \{x_i,\perp\} \mid E_i\right] \cdot \Pr\left[E_i\right] \\
=&\Pr\left[\mathsf{Dec}(i,z) \in \{x_i,\perp\} \mid \overline{E_i}\right] \cdot \Pr\left[\overline{E_i}\right] + \Pr\left[\mathsf{Dec}(i,y) \in \{x_i,\perp\} \mid E_i\right] \cdot \Pr\left[E_i\right] \\
&\qquad\qquad (z[[m]\setminus S_i] = y[[m]\setminus S_i]) \\
=&\Pr\left[\mathsf{Dec}(i,y) \in \{x_i,\perp\} \mid E_i\right] \cdot \Pr\left[E_i\right] \qquad \text{(Equation (1))} \\
\leq&\Pr\left[\mathsf{Dec}(i,y) \in \{x_i,\perp\} \mid E_i\right].
\end{aligned}$$

Note that by Claim 13, conditioned on E_i, $\mathsf{Dec}(i,\cdot)$ never outputs "$\perp$". We thus have

$$\Pr\left[\mathsf{Dec}(i,y) = x_i \mid E_i\right] \geq \frac{1}{2} + \varepsilon.$$

◀

We remark once again that the above lemma holds for the restricted code $C_{J|\rho}$, with S_i replaced by S'_i.

Below we prove an exponential lower bound for non-adaptive 2-query Hamming RLDCs.

▶ **Proposition 17.** *Let $C\colon \{0,1\}^n \to \{0,1\}^m$ be a non-adaptive weak $(2,\delta,1/2+\varepsilon)$-RLDC. Then $m = 2^{\Omega_{\delta,\varepsilon}(n)}$.*

Proof. Let $C_{J|\rho}\colon \{0,1\}^{n'} \to \{0,1\}^{m'}$ be the restricted code where $J|\rho$ is given by Lemma 12, and $A \subseteq [m]$ be the set of codeword bits which get fixed to constants. We also let $S'_i := S_i \setminus A$, $S'_{i,-} = S_{i,-} \setminus A$, $S'_{i,+} = S_{i,+} \setminus A$.

Denote $T'_j := \{i \in [n']\colon j \in S'_i\}$. Since $S'_i \subseteq S_i$ for each i, we also have $T'_j \subseteq T_j$ for each j. In particular, for each $j \notin W' \subseteq W$, we have $|T'_j| \le |T_j| \le 3\ln(8/\delta)$. Therefore

$$\mathop{\mathbb{E}}_{i\in[n']}\left[|S'_{i,-}|\right] = \frac{1}{n'}\sum_{i=1}^{n'}|S'_{i,-}| = \frac{1}{n'}\sum_{j\in[m']\setminus W'}|T'_j| \le 3\ln(8/\delta)\cdot\frac{m'}{n'}.$$

Therefore by Markov's inequality,

$$\Pr_{i\in[n']}\left[|S'_{i,-}| > \delta m'/4\right] \le \frac{12\ln(8/\delta)}{\delta n'} = O_\delta\left(\frac{1}{n'}\right).$$

In other words, there exists $I \subseteq [n']$ of size $|I| \ge n' - O_\delta(1)$ such that $|S'_{i,-}| \le \delta m'/4$ for all $i \in I$. For any such $i \in I$, we have $|S'_i| = |S'_{i,-}| + |S'_{i,+}| \le \delta m'/4 + \delta m'/4 = \delta m'/2$. By Lemma 16, we can view $C_{J|\rho}$ as a $(2,\delta/2,1/2+\varepsilon)$-LDC for message bits in I (for instance, we can arbitrarily fix the message bits outside I), where $|I| > n' - O_\delta(1) = \Omega(n)$. Finally, the statement of the proposition follows from Theorem 8. ◀

4.2 Lower bounds for adaptive 2-Query Hamming RLDCs

Now we turn to the actual proof, which still works for possibly adaptive decoders. Let C be a weak $(2,\delta,1/2+\varepsilon)$-RLDC with perfect completeness. We fix a relaxed decoder Dec for C. Without loss of generality, we assume Dec works as follows: on input $i \in [n]$, $\mathsf{Dec}(i,\cdot)$ picks the first query $j \in [m]$ according to a distribution $\mathcal{D}_i$. Let $b \in \{0,1\}$ be the answer to this query. Then Dec picks the second query $k \in [m]$ according to a distribution $\mathcal{D}_{i;j,b}$, and obtains an answer $b' \in \{0,1\}$. Finally, Dec outputs a random variable $X_{i;j,b,k,b'} \in \{0,1,\perp\}$.

We partition the support of $\mathcal{D}_i$ into the following two sets:

$$F_i^0 := \left\{j \in \mathrm{supp}(\mathcal{D}_i)\colon \forall b,b' \in \{0,1\}, k \in \mathrm{supp}(\mathcal{D}_{i;j,b,k,b'}), \Pr[X_{i;j,b,k,b'} = \perp] = 0\right\},$$
$$F_i^{>0} := \left\{j \in \mathrm{supp}(\mathcal{D}_i)\colon \exists b,b' \in \{0,1\}, k \in \mathrm{supp}(\mathcal{D}_{i;j,b,k,b'}), \Pr[X_{i;j,b,k,b'} = \perp] > 0\right\}.$$

We will still apply the restriction guaranteed by Lemma 12 to C. The sets S_i, T_j, W, $S_{i,-}$, $S_{i,+}$ (are their counterparts for $C_{J|\rho}$) are defined in the exact same way.

The following claim is adapted from Claim 13.

▷ Claim 18. $(\mathrm{supp}(\mathcal{D}_i) \setminus S_i) \subseteq F_i^0$.

Proof. Let $j \in \mathrm{supp}(\mathcal{D}_i) \setminus S_i$ and we will show $j \in F_i^0$. By the definition of S_i, $j \notin S_i$ means that there are partial assignments $\sigma_{00},\sigma_{01},\sigma_{10},\sigma_{11} \in \{0,1\}^{n-1}$ such that

$$C_j(\mathbf{x}_{-i} = \sigma_{00}, x_i = 0) = 0, \quad C_j(\mathbf{x}_{-i} = \sigma_{01}, x_i = 1) = 0,$$
$$C_j(\mathbf{x}_{-i} = \sigma_{10}, x_i = 0) = 1, \quad C_j(\mathbf{x}_{-i} = \sigma_{11}, x_i = 1) = 1,$$

where $\mathbf{x}_{-i}$ is defined as $\big(x_t \colon t \in [n] \setminus \{i\}\big)$.

Let $C_{00}, C_{01}, C_{10}, C_{11}$ be encodings of the corresponding assignments mentioned above. Consider an arbitrary query $k \in \mathrm{supp}(\mathcal{D}_{i;j,0})$, and let b_1', b_2' be the k-th bit of C_{00} and C_{01}, respectively. We note that $X_{i;j,0,k,b_1'}$ is the output of $\mathsf{Dec}(i, C_{00})$ conditioned on the queries j, k, and $X_{i;j,0,k,b_2'}$ is the output of $\mathsf{Dec}(i, C_{01})$ conditioned on the queries j, k. Due to perfect completeness of Dec, we have

$$\Pr[X_{i;j,0,k,b_1'} = 0] = 1, \quad \Pr[X_{i;j,0,k,b_2'} = 1] = 1.$$

Therefore, it must be the case that $b_1' \neq b_2'$, which implies that $\Pr[X_{i;j,0,k,b'} = \perp] = 0$ for any $b' \in \{0,1\}$.

An identical argument shows that $\Pr[X_{i;j,1,k,b'} = \perp] = 0$ for any $k \in \mathrm{supp}(\mathcal{D}_{i;j,1})$ and $b' \in \{0,1\}$. Thus we have shown $j \in F_i^0$. ◀

We remark that the above claim also implies $F_i^{>0} \subseteq S_i$, since $\mathrm{supp}(\mathcal{D}_i)$ is a disjoint union of F_i^0 and $F_i^{>0}$. In other words, conditioned on the event that the first query j is not contained in S_i, the decoder never outputs $\perp$.

The next claim is adapted from Claim 14.

▷ Claim 19. Let $j \in \mathrm{supp}(\mathcal{D}_i) \cap S_i$. For any $b \in \{0,1\}$ one of the following three cases occurs:

1. $\mathrm{supp}(\mathcal{D}_{i;j,b}) \subseteq S_i$;
2. For any $k \in \mathrm{supp}(\mathcal{D}_{i;j,b}) \setminus S_i$, $\Pr[X_{i;j,b,k,0} = b] = \Pr[X_{i;j,b,k,1} = b] = 1$;
3. For any $k \in \mathrm{supp}(\mathcal{D}_{i;j,b}) \setminus S_i$, $\Pr[X_{i;j,b,k,0} = 1-b] = \Pr[X_{i;j,b,k,1} = 1-b] = 1$.

Proof. Since $j \in S_i$, we may, without loss of generality, assume that $C_j \restriction_{x_i=0}$ is a constant function. Let us further assume $C_j \restriction_{x_i=0} \equiv 0$. The proofs for the other cases are going to be similar.

Suppose $\mathrm{supp}(\mathcal{D}_{i;j,0}) \not\subseteq S_i$, and let $k \in \mathrm{supp}(\mathcal{D}_{i;j,0}) \setminus S_i$. By the definition of S_i, $k \notin S_i$ means that there are partial assignments $\sigma_{00}, \sigma_{01} \in \{0,1\}^{n-1}$ such that

$$C_k(x_i = 0, \mathbf{x}_{-i} = \sigma_{00}) = 0, \quad C_k(x_i = 0, \mathbf{x}_{-i} = \sigma_{01}) = 1.$$

Let C_{00} and C_{01} be the encodings of the corresponding assignments mentioned above. We note that $X_{i;j,0,k,0}$ and $X_{i;j,0,k,1}$ are the outputs of $\mathsf{Dec}(i, C_{00})$ and $\mathsf{Dec}(i, C_{01})$, respectively, conditioned on the queries j, k. Due to perfect completeness of Dec, we must have

$$\Pr[X_{i;j,0,k,0} = 0] = \Pr[X_{i;j,0,k,1} = 0] = 1,$$

since both C_{00} and C_{01} encode messages with $x_i = 0$.

Now we claim that $C_j \restriction_{x_i=1} \equiv 1$ must hold. Otherwise there is a partial assignment $\sigma_{10} \in \{0,1\}^{n-1}$ such that

$$C_j(x_i = 1, \mathbf{x}_{-i} = \sigma_{10}) = 0.$$

Let C_{10} be the encoding of this assignment, and let $b' \in \{0,1\}$ be the k-th bit of C_{10}. On the one hand, $X_{i;j,0,k,b'}$ is the output $\mathsf{Dec}(i, C_{10})$ conditioned on the queries j, k, and we have just established

$$\Pr[X_{i;j,0,k,b'} = 0] = 1.$$

On the other hand, $\mathsf{Dec}(i, C_{10})$ should output $x_i = 1$ with probability 1 due to perfect completeness. This contradiction shows that $C_j \restriction_{x_i=1} \equiv 1$.

Similarly, suppose $\mathrm{supp}(\mathcal{D}_{i;j,1}) \not\subseteq S_i$ and let $k \in \mathrm{supp}(\mathcal{D}_{i;j,1}) \setminus S_i$, meaning that there are partial assignments $\sigma_{10}, \sigma_{11} \in \{0,1\}^{n-1}$ such that

$$C_k(x_i = 1, \mathbf{x}_{-i} = \sigma_{10}) = 0, \quad C_k(x_i = 1, \mathbf{x}_{-i} = \sigma_{11}) = 1.$$

Let C_{10} and C_{11} be the corresponding encodings, and note that $X_{i;j,1,k,0}$ and $X_{i;j,1,k,1}$ are the outputs of $\mathsf{Dec}(i, C_{10})$ and $\mathsf{Dec}(i, C_{11})$, respectively, conditioned on the queries j, k. Perfect completeness of Dec implies

$$\Pr[X_{i;j,1,k,0} = 1] = \Pr[X_{i;j,1,k,1} = 1] = 1,$$

since both C_{10} and C_{11} encode messages with $x_i = 1$.

So far we have shown that for any $b \in \{0,1\}$ such that $\mathrm{supp}(\mathcal{D}_{i;j,b}) \not\subseteq S_i$, it holds that

$$\forall k \in \mathrm{supp}(\mathcal{D}_{i;j,b}) \setminus S_i, \quad \Pr[X_{i;j,b,k,0} = b] = \Pr[X_{i;j,b,k,1} = b] = 1,$$

provided that $C_j \restriction_{x_i=0} \equiv 0$. In case of $C_j \restriction_{x_i=0} \equiv 1$, we can use an identical argument to deduce that for any $b \in \{0,1\}$ such that $\mathrm{supp}(\mathcal{D}_{i;j,b}) \not\subseteq S_i$, it holds that

$$\forall k \in \mathrm{supp}(\mathcal{D}_{i;j,b}) \setminus S_i, \quad \Pr[X_{i;j,b,k,0} = 1 - b] = \Pr[X_{i;j,b,k,1} = 1 - b] = 1.$$

◀

Here is another way to view Claim 19: conditioned on the event that the first query j is contained in S_i, either the second query k is also contained in S_i, or the output $X_{i;j,b,k,b'}$ is independent of the answer b' to query k. In either case, the decoder's output depends solely on the S_i-portion of the received string.

Once again, the conclusions of Claim 18 and Claim 19 hold for $C_{J|\rho}$, with S_i replaced by S'_i.

Finally, we are ready to prove Theorem 2. We recall the Theorem below.

▶ **Theorem 2.** *Let $C\colon \{0,1\}^n \to \{0,1\}^m$ be a weak adaptive $(2, \delta, 1/2 + \varepsilon)$-RLDC. Then $m = 2^{\Omega_{\delta,\varepsilon}(n)}$.*

Proof. The proof is almost identical to the one for Proposition 17. First, we can show that there exists $I \subseteq [n']$ of size $|I| \geq n' - O_\delta(1) = \Omega(n)$ such that $|S'_{i,-}| \leq \delta m/4$ for all $i \in I$, and hence $|S'_i| = |S'_{i,-}| + |S'_{i,+}| \leq \delta m/2$. Second, similar to the proof of Lemma 16, for each $i \in I$ we can construct a decoder D_i for x_i as follows. D_i restarts $\mathsf{Dec}(i, \cdot)$ until it makes a first query $j \in [m'] \setminus S'_i$. Then D_i finishes simulating $\mathsf{Dec}(i, \cdot)$ and returns its output. With the help of Claim 18 and Claim 19, the same analysis in Lemma 16 shows that D_i never returns $\perp$, and that the probability of returning x_i is at least $1/2 + \varepsilon$. Finally, the theorem follows from Theorem 8. ◀

References

1 Omar Alrabiah, Venkatesan Guruswami, Pravesh Kothari, and Peter Manohar. A near-cubic lower bound for 3-query locally decodable codes from semirandom csp refutation. *Electron. Colloquium Comput. Complex.*, 2022. URL: https://eccc.weizmann.ac.il/report/2022/101/.

2 Alexandr Andoni, Thijs Laarhoven, Ilya P. Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *SODA*, pages 47–66, 2017.

3 Vahid R. Asadi and Igor Shinkar. Relaxed locally correctable codes with improved parameters. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021*, volume 198 of *LIPIcs*, pages 18:1–18:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

4 László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC*, pages 21–31, 1991.

5 James L. Banal, Tyson R. Shepherd, Joseph Berleant, Hellen Huang, Miguel Reyes, Cheri M. Ackerman, Paul C. Blainey, and Mark Bathe. Random access dna memory using boolean search in an archival file storage system. *Nature Materials*, 20:1272–1280, 2021. doi:10.1101/2020.02.05.936369.

6 Avraham Ben-Aroya, Oded Regev, and Ronald de Wolf. A hypercontractive inequality for matrix-valued functions with applications to quantum computing and ldcs. In *FOCS*, pages 477–486. IEEE Computer Society, 2008.

7 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006. A preliminary version appeared in the Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC).

8 Arnab Bhattacharyya, L. Sunil Chandran, and Suprovat Ghoshal. Combinatorial lower bounds for 3-query ldcs. In *ITCS*, volume 151 of *LIPIcs*, pages 85:1–85:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

9 Arnab Bhattacharyya, Zeev Dvir, Shubhangi Saraf, and Amir Shpilka. Tight lower bounds for linear 2-query lccs over finite fields. *Comb.*, 36(1):1–36, 2016.

10 Arnab Bhattacharyya and Sivakanth Gopi. Lower bounds for constant query affine-invariant lccs and ltcs. *ACM Trans. Comput. Theory*, 9(2):7:1–7:17, 2017.

11 Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query lccs over large alphabet. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2017.

12 Alexander R. Block and Jeremiah Blocki. Private and resource-bounded locally decodable codes for insertions and deletions. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 1841–1846, 2021. doi:10.1109/ISIT45174.2021.9518249.

13 Alexander R. Block, Jeremiah Blocki, Elena Grigorescu, Shubhang Kulkarni, and Minshen Zhu. Locally decodable/correctable codes for insertions and deletions. In *FSTTCS*, volume 182 of *LIPIcs*, pages 16:1–16:17, 2020.

14 Jeremiah Blocki, Kuan Cheng, Elena Grigorescu, Xin Li, Yu Zheng, and Minshen Zhu. Exponential lower bounds for locally decodable and correctable codes for insertions and deletions. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 739–750, 2022. doi:10.1109/FOCS52979.2021.00077.

15 Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. *IEEE Transactions on Information Theory*, 67(7):4338–4360, 2021. doi:10.1109/TIT.2021.3076396.

16 Jeremiah Blocki, Shubhang Kulkarni, and Samson Zhou. On Locally Decodable Codes in Resource Bounded Channels. In Yael Tauman Kalai, Adam D. Smith, and Daniel Wichs, editors, *1st Conference on Information-Theoretic Cryptography (ITC 2020)*, volume 163, pages 16:1–16:23, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12121, doi:10.4230/LIPIcs.ITC.2020.16.

17 Manuel Blum and Sampath Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, 1995.

18 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993.

19 Joshua Brakensiek, Venkatesan Guruswami, and Samuel Zbarsky. Efficient low-redundancy codes for correcting multiple deletions. *IEEE Trans. Inf. Theory*, 64(5):3403–3410, 2018.

20 Victor Chen, Elena Grigorescu, and Ronald de Wolf. Error-correcting data structures. *SIAM J. Comput.*, 42(1):84–111, 2013.

21 Kuan Cheng, Venkatesan Guruswami, Bernhard Haeupler, and Xin Li. Efficient linear and affine codes for correcting insertions/deletions. In *SODA*, pages 1–20. SIAM, 2021.

22 Kuan Cheng, Bernhard Haeupler, Xin Li, Amirbehshad Shahrasbi, and Ke Wu. Synchronization strings: Highly efficient deterministic constructions over small alphabets. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2185–2204. SIAM, 2019.

23 Kuan Cheng, Zhengzhong Jin, Xin Li, and Ke Wu. Deterministic document exchange protocols, and almost optimal binary codes for edit errors. In Mikkel Thorup, editor, *FOCS*, pages 200–211, 2018.

24 Kuan Cheng, Zhengzhong Jin, Xin Li, and Ke Wu. Block edit errors with transpositions: Deterministic document exchange protocols and almost optimal binary codes. In *ICALP*, volume 132 of *LIPIcs*, pages 37:1–37:15, 2019.

25 Kuan Cheng and Xin Li. Efficient document exchange and error correcting codes with asymmetric information. In *SODA*, pages 2424–2443. SIAM, 2021.

26 Kuan Cheng, Xin Li, and Yu Zheng. Locally decodable codes with randomized encoding. *CoRR*, abs/2001.03692, 2020. URL: `https://arxiv.org/abs/2001.03692`.

27 Alessandro Chiesa, Tom Gur, and Igor Shinkar. Relaxed locally correctable codes with nearly-linear block length and constant query complexity. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1395–1411. SIAM, 2020.

28 Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

29 Gil Cohen and Tal Yankovitz. Relaxed locally decodable and correctable codes: Beyond tensoring. *Electron. Colloquium Comput. Complex.*, TR22-045, 2022. URL: `https://eccc.weizmann.ac.il/report/2022/045`, `arXiv:TR22-045`.

30 Marcel Dall'Agnol, Tom Gur, and Oded Lachish. A structural theorem for local algorithms with applications to coding, testing, and privacy. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1651–1665. SIAM, 2021.

31 Yan Ding, Parikshit Gopalan, and Richard Lipton. Error correction against computationally bounded adversaries. Manuscript, 2004.

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

33 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Superquadratic lower bound for 3-query locally correctable codes over the reals. *Theory Comput.*, 13(1):1–36, 2017.

34 Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM J. Comput.*, 41(6):1694–1703, 2012.

35 Anna Gál and Andrew Mills. Three-query locally decodable codes with higher correctness require exponential length. *ACM Trans. Comput. Theory*, 3(2):5:1–5:34, 2012.

36 William I. Gasarch. A survey on private information retrieval (column: Computational complexity). *Bulletin of the EATCS*, 82:72–107, 2004.

37 Oded Goldreich, Howard J. 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.

38 Tom Gur and Oded Lachish. A lower bound for relaxed locally decodable codes. *arXiv preprint arXiv:1904.08112*, 2019.

39 Tom Gur and Oded Lachish. On the power of relaxed local decoding algorithms. *SIAM J. Comput.*, 50(2):788–813, 2021.

40 Tom Gur, Govind Ramnarayan, and Ron Rothblum. Relaxed locally correctable codes. *Theory Comput.*, 16:1–68, 2020.

983 **41** Venkatesan Guruswami, Bernhard Haeupler, and Amirbehshad Shahrasbi. Optimally resilient
984 codes for list-decoding from insertions and deletions. In Konstantin Makarychev, Yury
985 Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *STOC*, pages
986 524–537. ACM, 2020.
987 **42** Venkatesan Guruswami and Ray Li. Coding against deletions in oblivious and online models.
988 In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on*
989 *Discrete Algorithms*, pages 625–643. SIAM, 2018.
990 **43** Venkatesan Guruswami and Ray Li. Polynomial time decodable codes for the binary deletion
991 channel. *IEEE Trans. Inf. Theory*, 65(4):2171–2178, 2019.
992 **44** Venkatesan Guruswami and Adam Smith. Optimal rate code constructions for computationally
993 simple channels. *J. ACM*, 63(4):35:1–35:37, September 2016. URL: `http://doi.acm.org/10.`
994 `1145/2936015`, `doi:10.1145/2936015`.
995 **45** Venkatesan Guruswami and Carol Wang. Deletion codes in the high-noise and high-rate
996 regimes. *IEEE Transactions on Information Theory*, 63(4):1961–1970, 2017.
997 **46** Bernhard Haeupler. Optimal document exchange and new codes for insertions and deletions.
998 In David Zuckerman, editor, *FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*,
999 pages 334–347, 2019.
1000 **47** Bernhard Haeupler, Aviad Rubinstein, and Amirbehshad Shahrasbi. Near-linear time insertion-
1001 deletion codes and (1+ϵ)-approximating edit distance via indexing. In Moses Charikar and
1002 Edith Cohen, editors, *STOC*, pages 697–708. ACM, 2019.
1003 **48** Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: codes for insertions
1004 and deletions approaching the singleton bound. In Hamed Hatami, Pierre McKenzie, and
1005 Valerie King, editors, *STOC*, pages 33–46. ACM, 2017.
1006 **49** Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: explicit construc-
1007 tions, local decoding, and applications. In Ilias Diakonikolas, David Kempe, and Monika
1008 Henzinger, editors, *STOC*, pages 841–854. ACM, 2018.
1009 **50** Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings and codes for
1010 insertions and deletions – a survey, 2021. `arXiv:2101.00711`.
1011 **51** Bernhard Haeupler, Amirbehshad Shahrasbi, and Madhu Sudan. Synchronization strings: List
1012 decoding for insertions and deletions. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel
1013 Marx, and Donald Sannella, editors, *ICALP*, volume 107 of *LIPIcs*, pages 76:1–76:14, 2018.
1014 **52** Brett Hemenway and Rafail Ostrovsky. Public-key locally-decodable codes. In *Advances in*
1015 *Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Proceedings*,
1016 pages 126–143, 2008.
1017 **53** Brett Hemenway, Rafail Ostrovsky, Martin J. Strauss, and Mary Wootters. Public key
1018 locally decodable codes with short keys. In *14th International Workshop, APPROX, and 15th*
1019 *International Workshop, RANDOM, Proceedings*, pages 605–615, 2011.
1020 **54** Brett Hemenway, Rafail Ostrovsky, and Mary Wootters. Local correctability of expander
1021 codes. *Inf. Comput.*, 243:178–190, 2015.
1022 **55** Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-
1023 correcting codes. In *STOC*, pages 80–86, 2000.
1024 **56** Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable
1025 codes via a quantum argument. *J. Comput. Syst. Sci.*, 69(3):395–420, 2004.
1026 **57** Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable
1027 and locally testable codes with sub-polynomial query complexity. *J. ACM*, 64(2):11:1–11:42,
1028 2017.
1029 **58** Swastik Kopparty and Shubhangi Saraf. Guest column: Local testing and decoding of high-rate
1030 error-correcting codes. *SIGACT News*, 47(3):46–66, 2016.
1031 **59** Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and
1032 reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. Doklady Akademii Nauk SSSR, V163
1033 No4 845-848 1965.
1034 **60** Richard J. Lipton. A new approach to information theory. In *STACS*, pages 699–708, 1994.

61 Shu Liu, Ivan Tjuawinata, and Chaoping Xing. On list decoding of insertion and deletion errors. *CoRR*, abs/1906.09705, 2019. URL: http://arxiv.org/abs/1906.09705.

62 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

63 Jiri Matousek Marcos Kiwi, Martin Loebl. Expected length of the longest common subsequence for large alphabets. *Advances in Mathematics*, 197(2):480–498, 2005.

64 Hugues Mercier, Vijay K. Bhargava, and Vahid Tarokh. A survey of error-correcting codes for channels with symbol synchronization errors. *IEEE Communications Surveys and Tutorials*, 12, 2010.

65 Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson. Optimal error correction against computationally bounded noise. In *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, pages 1–16, 2005.

66 Michael Mitzenmacher. A survey of results for deletion channels and related synchronization channels. *Probability Surveys*, 6:1–3, 07 2008.

67 Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private locally decodable codes. In *ICALP*, pages 387–398, 2007.

68 Rafail Ostrovsky and Anat Paskin-Cherniavsky. Locally decodable codes for edit distance. In Anja Lehmann and Stefan Wolf, editors, *Information Theoretic Security*, pages 236–249, Cham, 2015. Springer International Publishing.

69 L. J. Schulman and D. Zuckerman. Asymptotically good codes correcting insertions, deletions, and transpositions. *IEEE Transactions on Information Theory*, 45(7):2552–2557, 1999.

70 Ronen Shaltiel and Jad Silbak. Explicit list-decodable codes with optimal rate for computationally bounded channels. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 45:1–45:38, 2016.

71 N.J.A. Sloane. On single-deletion-correcting codes. *arXiv: Combinatorics*, 2002.

72 Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma (abstract). In *CCC*, page 4, 1999.

73 Luca Trevisan. Some applications of coding theory in computational complexity. *CoRR*, cs.CC/0409044, 2004.

74 Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 1424–1436. Springer, 2005.

75 David P. Woodruff. New lower bounds for general locally decodable codes. Technical report, Weizmann Institute of Science, Israel, 2007.

76 David P. Woodruff. A quadratic lower bound for three-query linear locally decodable codes over any field. *J. Comput. Sci. Technol.*, 27(4):678–686, 2012.

77 S. M. Hossein Tabatabaei Yazdi, Ryan Gabrys, and Olgica Milenkovic. Portable and error-free dna-based data storage. *Scientific Reports*, 7:2045–2322, 2017. doi:https://doi.org/10.1038/s41598-017-05188-1.

78 Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1):1:1–1:16, 2008.

79 Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.