Exponential Lower Bounds for Locally Decodable and Correctable Codes for Insertions and Deletions

Jeremiah Blocki*[1], Kuan Cheng[2,3], Elena Grigorescu†[1], Xin Li ‡[4], Yu Zheng §[4], and Minshen Zhu[1]

[1]Department of Computer Science, Purdue University
[2]Center on Frontiers of Computing Studies, Peking University
[3]Advanced Institute of Information Technology, Peking University
[4]Department of Computer Science, Johns Hopkins University
{jblocki, elena-g, zhu628}@purdue.edu
ckkcdh@pku.edu.cn
{lixints, yuzheng}@cs.jhu.edu

Abstract

Locally Decodable Codes (LDCs) are error-correcting codes for which individual message symbols can be quickly recovered despite errors in the codeword. LDCs for Hamming errors have been studied extensively in the past few decades, where a major goal is to understand the amount of redundancy that is necessary and sufficient to decode from large amounts of error, with small query complexity.

Motivated by new progress in DNA-storage technologies (Banal et al., *Nature Materials, 2021*), in this work we study LDCs for *insertion* and *deletion* errors, called *Insdel LDCs*. Their study was initiated by Ostrovsky and Paskin-Cherniavsky (*Information Theoretic Security, 2015*), who gave a reduction from Hamming LDCs to Insdel LDCs with a small blowup in the code parameters. On the other hand, the only known lower bounds for Insdel LDCs come from those for Hamming LDCs, thus there is no separation between them. Here we prove new, strong lower bounds for the existence of Insdel LDCs. In particular, we show that 2-query *linear* Insdel LDCs do not exist, and give an exponential lower bound for the length of all q-query Insdel LDCs with constant q. For $q \geq 3$ our bounds are exponential in the existing lower bounds for Hamming LDCs. Furthermore, our exponential lower bounds continue to hold for adaptive decoders, and even in private-key settings where the encoder and decoder share secret randomness. This exhibits a strict separation between Hamming LDCs and Insdel LDCs.

Our strong lower bounds also hold for the related notion of Insdel LCCs (except in the private-key setting), due to an analogue to the Insdel notions of a reduction from Hamming LCCs to LDCs.

Our techniques are based on a delicate design and analysis of hard distributions of insertion and deletion errors, which depart significantly from typical techniques used in analyzing Hamming LDCs.

*Supported by NSF CAREER Award CNS-2047272 and NSF Awards CCF-1910659 and CNS-1931443,
†Supported by NSF CCF-1910659 and NSF CCF-1910411.
‡Supported by NSF CAREER Award CCF-1845349 and NSF Award CCF-2127575.
§Supported by NSF CAREER Award CCF-1845349.

1 Introduction

Error correcting codes are fundamental mathematical objects in both theory and practice, whose study dates back to the pioneering work of Shannon and Hamming in the 1950's. While the study of classical codes focuses on unique decoding from *Hamming* errors, many exciting variants have emerged ever since, such as list-decoding, which can go beyond the half distance barrier, and *local decoding*, which can decode any message symbol by querying only a few codeword symbols. These variants have proved to be closely connected to diverse areas in computer science.

Similarly, another line of work studies *synchronization* errors, namely *insertions* and *deletions* (*insdels*, for short), which are strictly more general than Hamming errors and happen frequently in various applications such as text/speech processing, media access, and communication systems. The study of codes for such errors (*insdel codes*, for short) also has a long history that goes back to the work of Levenstein [Lev66] in the 1960's.

This paper focuses on locally decodable codes correcting insertions and deletions, which we call *Insdel LDCs*. We prove the first non-trivial lower bounds for such codes, which in turn provide a strong separation between Hamming LDCs and Insdel LDCs. Furthermore, these results imply similar strong bounds for the related notion of locally correctable codes correcting insertions and deletions, which we call *Insdel LCCs*.

More formally, Locally Decodable Codes (LDCs) are error-correcting codes $C : \Sigma^n \to \Sigma^m$ that allow very fast recovery of individual symbols of a *message* $x \in \Sigma^n$, even when worst-case errors are introduced in the encoded message, called *codeword* $C(x)$. Similarly, Locally Correctable Codes (LCCs) are error-correcting codes $C : \Sigma^n \to \Sigma^m$ that allow very fast recovery of individual symbols of the codeword $C(x) \in \Sigma^m$, even when worst-case errors are introduced. In what follows, for ease of presentation, we will discuss our results and related work by focusing on the notion of LDCs, and we will return to the implications to Insdel LCCs in Section 1.1.3. We remark that the previous lower bounds for Hamming LCCs are asymptotically the same as for LCCs due to a folklore reduction between the two notions (e.g. formalized in [KV10, BGT16]).

The important *parameters* of LDCs are their *rate*, defined as the ratio between the message length n and the codeword length m, measuring the amount of redundancy in the encoding; their *relative minimum distance*, defined as the minimum normalized Hamming distance between any pair of codewords, a parameter relevant to the fraction of correctable errors; and their *locality* or *query complexity*, defined as the number of queries a decoder makes to a received word $y \in \Sigma^m$ in order to decode the symbol at location $i \in [n]$ of the message, namely x_i.

Since they were introduced in [KT00, STV99], LDCs have found many applications in private information retrieval, probabilistically checkable proofs, self-correction, fault-tolerant circuits, hardness amplification, and data structures (e.g., [BFLS91, LFKN92, BLR93, BK95, CKGS98, CGdW13, ALRW17] and surveys [Tre04, Gas04]), and the tradeoffs between the achievable parameters of Hamming LDCs has been studied extensively [KdW04, WdW05, GKST06, Woo07, Yek08, Yek12, DGY11, Efr12, GM12, BDSS16, BG17, DSW17, KMRS17, BCG20] (see also surveys by Yekhanin [Yek12] and by Kopparty and Saraf [KS16]). This sequence of results has brought up exciting progress regarding the necessary and sufficient rate for codes with small query complexity that can withstand a constant fraction of errors. Nevertheless, many important parameter regimes leave wide gaps in our current understanding of LDCs. For example, even for 3-query Hamming LDCs the gap between constructions and lower bounds is superpolynomial [Yek08, DGY11, Efr12, KT00, Woo07, Woo12]. (Note: [GM12] established an exponential lower bound on the length of 3-query LDCs for some parameter regimes, but it does not rule out the

possibility of a 3-query LDC with polynomial length in natural parameter ranges.)

More specifically, for 2-query Hamming LDCs we have matching upper and lower bounds of $m = 2^{\Theta(n)}$, where the upper bound is achieved by the simple Hadamard code while the lower bound is established in [KdW04, BRdW08]. In the constant-query regime where the decoder makes 2^t many queries, for some $t > 1$, the best known constructions of Hamming LDCs are based on matching-vector codes, and give codes that map n symbols into $m = \exp(\exp((\log n)^{1/t}(\log\log n)^{1-1/t}))$ symbols [Yek08, DGY11, Efr12], while the best general lower bound for q-query LDC is $\Omega(n^{\frac{q+1}{q-1}})/\log n$ when $q \geq 3$ [Woo07]. In the polylog(n)-query regime, Reed-Muller codes are examples of $\log^c n$-query Hamming LDCs of block length $n^{1+\frac{1}{c-1}+o(1)}$ for some $c > 0$ (e.g., see [Yek12]). Finally, there exist sub-polynomial (but super logarithmic)-query Hamming LDCs with constant rate [KMRS17]. These latter constructions improve upon the previous constant-rate codes in the n^ϵ-query regime achieved by Reed-Muller codes, and upon the more efficient constructions of [KSY14]. In a different model, if we assume that the encoder and decoder have shared secret randomness [Lip94], then it becomes much easier to construct LDCs. For example, [OPS07] constructs private-key Hamming LDCs with constant rate (i.e., $m = \Theta(n)$) and query complexity polylog(n), and a simple modification yields a private-key Hamming LDC with rate $m = \tilde{\Theta}(n)$ and query complexity 1 — see details in Appendix A.

Regarding insdel codes, following the work of Levenstein [Lev66], the progress has historically been slow, due to the fact that synchronization errors often result in the loss of index information. Indeed, constructing codes for insdel errors is strictly more challenging than for Hamming errors. However the interest in these codes has been rekindled lately, leading to a wave of new results [SZ99, Kiw05, GW17, HS17, GL18, HSS18, HS18, BGZ18, CJLW18, CHL+19, CJLW19, GL19, HRS19, Hae19, LTX19, GHS20, CGHL21, CL21] (See also the excellent surveys of [Slo02, MBT10, Mit08, HS21]) with almost optimal parameters in various settings, and the variant of "list-decodable" insdel codes, that can withstand a larger fraction of errors while outputting a small list of potential codewords [HSS18, GHS20, LTX19]. However, none of these works addresses insdel LDCs, which we believe are natural objects in the study of insdel codes, since such codes are often used in applications involving large data sets.

Insdel LDCs were first introduced in [OPC15] and further studied in [BBG+20, BB21, CLZ20]. In [OPC15, BBG+20] the authors give Hamming to Insdel reductions which transform any Hamming LDC into an Insdel LDC. These reductions decrease the rate by a constant multiplicative factor and increase the locality by a $\log^{c'}(m)$ multiplicative factor for a fixed constant $c' > 1$. Applying the compilers to the above-mentioned constructions of Reed-Muller codes gives $(\log n)^{c+c'}$-query Insdel LDCs of length $m = n^{1+\frac{1}{c-1}+o(1)}$, for any $c > 1$. Also, applying the compilers to the LDCs in [KMRS17] yields Insdel LDCs of constant rate and $\exp(\tilde{O}(\sqrt{\log n}))$-query complexity.

Unfortunately, these compilers do not imply constant-query Insdel LDCs, and in fact, even after this work, we do not know if constant-query Insdel LDCs exist in general. In the private-key setting, applying the compilers to [OPS07] yields a private-key Insdel LDC with constant rate and locality polylog(n) [CLZ20, BB21].

We now formally define the notion of Insdel LDCs. See Appendix A for further discussion.

Definition 1. *[Insdel Locally Decodable Codes (Insdel LDCs)] Fix an integer q and constants $\delta \in [0,1]$, $\varepsilon \in (0, \frac{1}{2}]$. We say $C\colon \{0,1\}^n \to \Sigma^m$ is a (q,δ,ε)-*locally decodable insdel code *if there exists a probabilistic algorithm* Dec *such that:*

- *For every $x \in \{0,1\}^n$ and $y \in \Sigma^{m'}$ such that $\mathsf{ED}\left(C(x), y\right) \leq \delta \cdot 2m$, and for every $i \in [n]$, we*

have

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

where the probability is taken over the randomness of Dec*, and* $\mathsf{ED}\left(C(x), y\right)$ *denotes the minimum number of insertions/deletions necessary to transform* $C(x)$ *into* y*.*

- *In every invocation,* Dec *reads at most* q *symbols of* y*. We say that* Dec *is* non-adaptive *if the distribution of queries of* $\mathsf{Dec}(y, m', i)$ *is independent of* y*.*

Note that in this definition we allow the decoder to have as an input m', the length of the string y. This only makes our lower bounds stronger. We can also extend the definition to private-key LDC, where the encoder and decoder share secret randomness, and we relax the requirement that $\Pr\left[\mathsf{Dec}(y, m', i) = x_i\right] \geq \frac{1}{2} + \varepsilon$ for all y s.t. $\mathsf{ED}\left(C(x), y\right) \leq \delta \cdot 2m$. Instead, we require that any attacker who does not have the secret randomness (private-key) cannot produce y such that $\mathsf{ED}\left(C(x), y\right) \leq \delta \cdot 2m$ and $\Pr\left[\mathsf{Dec}(y, m', i) = x_i\right] < \frac{1}{2} + \varepsilon$ except with negligible probability — see Appendix B.

In this work we focus on *binary* Insdel LDCs and give the first non-trivial lower bounds for such codes. In most cases, such as constant-query Insdel LDCs, our bounds are exponential in the message length. We note that prior to our work, the only known lower bounds for Insdel LDCs come from the lower bounds for Hamming LDCs (since Hamming erros can be implemented by insdel errors), and thus there is no separation of Insdel LDC and Hamming LDC. In particular, these bounds don't even preclude the possibility of a 3-query Insdel LDC with $m = \Theta(n^2)$. We also note that we mainly prove lower bounds for Insdel LDCs with non-adaptive decoders. However, by using a reduction suggested in [KT00] we obtain almost the same lower bounds for Insdel LDCs with adaptive decoders.

Our results provide a strong separation between Insdel LDCs and Hamming LDCs in several contexts. First, many of our exponential lower bounds continue to apply in the setting of private-key LDCs, while in such settings it is easy to construct private-key Hamming LDCs with $m = \tilde{O}(n)$ and locality 1. Second, our exponential lower bounds hold for any constant q, while even for $q = 3$ we have constructions of Hamming LDCs with sub-exponential length. Finally, for $q = 2$ we rule out the possibility of linear Insdel LDCs, while the Hadamard code is a simple 2-query Hamming LDC. This separation is in sharp contrast to the situation of unique decoding with codes for Hamming errors vs. codes for insdel errors, where they have almost the same parameter tradeoffs.

Motivation of Insdel LDCs in DNA storage DNA storage [YGM17] is a storage medium that harnesses the biology of DNA sequences, to store and transmit not only genetic information, but also any arbitrary digital data, despite the presence of insertion and deletion errors. It has the potential of becoming the storage medium of the future, due to its superior scaling properties, provided new techniques for random data access are developed. Recent progress towards achieving effective and reliable DNA random access technology is motivated by the fact that a *"crucial aspect of data storage systems is the ability to efficiently retrieve specific files or arbitrary subsets of files."* [BSB+21]. This is also precisely the real-world goal formalized by the notion of Insdel LDCs, which motivates a systematic theoretical study of such codes and of their limitations.

1.1 Our results

1.1.1 Lower bounds for 2 query Insdel LDCs

We first present our result for *linear* codes. Linear codes are defined over a finite field $\Sigma = \mathbb{F}$, and the codewords form a linear subspace in $\mathbb{F}^m$. Similarly, the codewords of an *affine* code form an affine subspaces in $\mathbb{F}^m$. Lower bounds for the length of 2-query linear Hamming LDCs were first studied in [GKST06], where the authors proved an exponential bound. This is matched by the Hadamard code.

In [Woo12] Woodruff give a $m = \Omega(n^2)$ bound for 3-query linear codes, which is still the best known for any linear code with $q \geq 3$.

Furthermore, the best upper bounds of [Yek08, DGY11, Efr12, KMRS17], and, to the best of our knowledge, all known constructions of Hamming LDCs are achieved by linear codes. As further motivation for studying linear LDCs, lower bounds for linear 2-query (Hamming) LDCs are useful in polynomial identity testing [DS07], and they are known to imply lower bounds on matrix rigidity [Dvi10]. In addition, a recent work [CGHL21] has initiated a systematic study on linear insdel codes.

We first show that 2-query linear insdel LDCs do not exist.

Theorem 1. *For any $(2, \delta, \varepsilon)$ linear or affine insdel LDC $C : \{0,1\}^n \to \{0,1\}^m$, we have $n = O_{\delta,\varepsilon}(1)$.*

More generally, we show an exponential lower bound for general 2-query insdel LDCs.

Theorem 2. *For any $(2, \delta, \varepsilon)$ insdel LDC $C : \{0,1\}^n \to \{0,1\}^m$, we have $m = \exp(\Omega_{\delta,\varepsilon}(n))$.*

We remark that, as previously mentioned, the lower bound for 2-query Hamming LDCs from [KdW04] also holds for 2-query Insdel LDCs. However, that proof uses sophisticated quantum arguments, and an important quest in the area has been providing non-quantum proofs for the same result. Indeed, the proof from [KdW04] was adapted to classical arguments by [BRdW08], but the arguments still retained a strong quantum-style flavor. Our arguments here do not resemble those proofs and are purely classical. Furthermore, in contrast to the lower bounds from [KdW04, BRdW08], our lower bounds in Theorems 1 and 2 extend to the private-key setting where the encoder and decoder share private randomness. We note that one *can* easily obtain private-key Hamming LDCs with $m = \tilde{O}(n)$ and locality 1 by modifying the construction of [OPS07] — see details in Appendix B.

By contrast, for any constants $\epsilon, \delta > 0$ our results rule out the possibility of 2-query Insdel LDCs with $m = \exp(o(n))$ even in the private-key setting.

1.1.2 Lower bounds for $q \geq 3$ query Insdel LDCs

We prove the following general bound for $q \geq 3$ queries.

Theorem 3. *For any non-adaptive (q, δ, ε) insdel LDC C: $\{0,1\}^n \to \{0,1\}^m$ with $q \geq 3$, we have the following bounds.*

- *For arbitrary adversarial channels,*

$$m = \begin{cases} \exp(\Omega_{\delta,\varepsilon}(\sqrt{n})) \text{ for } q = 3\text{; and} \\ \exp\left(\Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)} \cdot (\varepsilon^3 n)^{1/(2q-4)}\right)\right) \text{ for } q \geq 4. \end{cases}$$

- *For the private-key setting,*

$$m = \exp\left(\Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)} \cdot \left(\varepsilon^3 n\right)^{1/(2q-3)}\right)\right).$$

As a comparison, for general Hamming LDCs the best known lower bounds for $q \geq 3$ in [Woo07] give $m = \Omega(n^2/\log n)$ for $q = 3$, and $m = \Omega(n^{1+1/\lceil (q-1)/2 \rceil}/\log n)$ for $q > 3$. Thus, in the constant-query regime, the bounds from Theorem 3 are essentially exponential in the existing bounds for Hamming LDCs. Moreover, these bounds also give a separation between constant-query Hamming LDCs, which can have length $\exp(n^{o(1)})$, and constant-query insdel LDCs.

Lower bounds for adaptive decoders It is well-known [KT00] that a (q, δ, ε) adaptive Hamming LDC can be converted into a non-adaptive $(\frac{|\Sigma|^q - 1}{|\Sigma| - 1}, \delta, \varepsilon)$ Hamming LDC, and also into a non-adaptive $(q, \delta, \varepsilon/|\Sigma|^{q-1})$ Hamming LDC, and hence lower bounds for non-adaptive decoders imply lower bounds for adaptive decoders, with the respective loss in parameters. It is easy to verify that the same reduction works for Insdel LDCs.[1] In particular our lower bounds imply the respective lower bounds for adaptive decoders.

Corollary 1. *For any (possibly adaptive) (q, δ, ε) insdel LDC $C\colon \{0,1\}^n \to \{0,1\}^m$ with $q \geq 3$, we have the following bounds.*

- *For arbitrary adversarial channels,*

$$m = \begin{cases} \exp(\Omega_{\delta,\varepsilon}(\sqrt{n})) \text{ for } q = 3\text{; and} \\ \exp\left(\Omega\left(\frac{\delta}{(q+\ln(q/\varepsilon))^2} \cdot \left(\varepsilon^3 n\right)^{1/(2q-4)}\right)\right) \text{ for } q \geq 4. \end{cases}$$

- *For the private-key setting,*

$$m = \exp\left(\Omega\left(\frac{\delta}{(q+\ln(q/\varepsilon))^2} \cdot \left(\varepsilon^3 n\right)^{1/(2q-3)}\right)\right).$$

Corollary 1 is obtained by plugging $\epsilon' = \epsilon/2^{q-1}$ into Theorem 3 and applying the average case reduction from a (q, δ, ϵ) (adaptive) Insdel LDC to a $(q, \delta, \epsilon/2^{q-1})$ (non-adaptive) Insdel LDC [KT00]. Corollary 1 also implies lower bounds in regimes where q is slightly super-constant (but $o(\log n)$).

Corollary 2. *For any (possibly adaptive) (q, δ, ε) insdel LDC $C\colon \{0,1\}^n \to \{0,1\}^m$, the following bounds hold.*

[1] For example, our non-adaptive decoder can pick $r_1, \ldots, r_{q-1} \in \Sigma$ randomly and simulate the adaptive (q, δ, ϵ)-decoder responding to the first $q-1$ queries with $r_1, \ldots, r_{q-1}$. This allows the non-adaptive decoder to extract a set $(j_1, \ldots, j_q)$ of queries representing the set of queries that the adaptive decoder would have asked given the first $q-1$ responses. The queries $(j_1, \ldots, j_q)$ can then be asked non-adaptively to obtain $y[j_1], \ldots, y[j_q]$. With probability $|\Sigma|^{-q+1}$ we will have $y[j_i] = r_i$ for each $i \leq q-1$ and we can finish simulating the adaptive decoder to obtain a prediction x_i which will be correct with probability at least $\frac{1}{2} + \varepsilon$. Otherwise, our non-adaptive decoder randomly guesses the output bit x_i. Thus, the non-adaptive decoder is successful with probability at least $\frac{1}{2} + \epsilon|\Sigma|^{-q+1}$.

- *If* $q = O(\log \log n)$*, then* $m = \exp\left(\exp(\Omega_{\delta,\varepsilon}(\log n/\log \log n))\right)$.
- *If* $q = \log n/(2c \log \log n)$ *for some* $c > 3$*, then* $m = \exp(\Omega(\log^{c-2} n))$*. In turn, if* $m = \text{poly}(n)$*, then* $q = \Omega(\log n/\log \log n)$.

Moreover, the lower bounds hold even in private-key settings.

We remark that the lower bound for $q = O(\log \log n)$ queries is even larger than the Hamming LDC upper bound of $\exp(\exp((\log n)^{1/t}(\log \log n)^{1-1/t}))$ due to [Yek08, DGY11, Efr12] for $q = 2^t$ being a constant number of queries.

Furthermore, we get a super-polynomial lower bound even if $q = \log n/(8 \log \log n)$. Thus to get any polynomial length Insdel LDC one needs $q = \Omega(\log n/\log \log n)$. This can be compared to the Insdel LDC constructions in [OPC15, BBG+20], which give $m = o(n^2)$ with $q = (\log n)^C$ for some $C > 2$ (or to the private-key Insdel LDC construction in [CLZ20, BB21] which gives constant rate $m = \Theta(n)$ and $q = (\log n)^C$ for some $C > 2$). Both the lower bound and the upper bound are for an adaptive Insdel LDC, so our lower bound on the query complexity almost matches the upper bound for polynomial length Insdel LDCs. This also implies that there is a "phase transition" phenomenon in the $q = \text{polylog}(n)$ regime, where the length of the Insdel LDC transits from super-polynomial to polynomial.

1.1.3 Implications to lower bounds for Insdel LCCs

As mentioned above, the lower bounds for Insdel LDCs extend to Insdel LCCs due an analogue to Insdel errors of a reduction [KV10, BGT16] between the two notions in the Hamming error model. More specifically, in [KV10, BGT16], the authors show that in the standard Hamming error case, any q-query LCC can be converted into a q-query LDC with only a constant loss in rate, and preserving the other relevant parameters. In [KV10], Kaufman and Viderman show that the two notions are not equivalent in some specific sense, as there exist LDCs that are not LCCs [KV10].

We start with a formal definition.

Definition 2. *[Insdel Locally Correctable Codes (Insdel LCCs)] Fix an integer* q *and constants* $\delta \in [0, 1]$, $\varepsilon \in (0, \frac{1}{2}]$*. We say* $C\colon \{0,1\}^n \to \Sigma^m$ *is a* (q, δ, ε)-locally correctable insdel code *if there exists a probabilistic algorithm* Dec *such that:*

- *For every* $x \in \{0,1\}^n$ *and* $y \in \Sigma^{m'}$ *such that* $\mathsf{ED}\left(C(x), y\right) \le \delta \cdot 2m$*, and for every* $i \in [m]$*, we have*

$$\Pr\left[\mathsf{Dec}(y, m', i) = C(x)_i\right] \ge \frac{1}{2} + \varepsilon,$$

  where the probability is taken over the randomness of Dec*, and* $\mathsf{ED}\left(C(x), y\right)$ *denotes the minimum number of insertions/deletions necessary to transform* $C(x)$ *into* y.

- *In every invocation,* Dec *reads at most* q *symbols of* y*. We say that* Dec *is* non-adaptive *if the distribution of queries of* $\mathsf{Dec}(y, m', i)$ *is independent of* y.

We note that if C is a linear/affine insdel LCCs then C is also an insdel LDC. Indeed, linear/affine codes are *systematic* codes, and hence the message bits appear as codeword bits. This is also the case in the private-key setting. For completeness, we include a proof in Appendix D. Hence our lower

bounds about linear/affine insdel LDCs apply to linear/affine insdel LCCs, even in the private-key setting.

Our results can be extended to non-linear LCCs and LDCs (but not in the private-key setting), using the following theorem, which we prove in Appendix D via small adaptations to the proof of [BGT16].

Theorem 4. *Let $C : \{0,1\}^k \to \Sigma^m$ be a (q, δ, ε)-insdel LCC, then there exists a (q, δ, ε)-insdel LDC $C' : \{0,1\}^{k'} \to \Sigma^m$ with*

$$k' = \Omega\left(\frac{k}{\log(1/\delta)}\right).$$

We conclude the following about Insdel LCCs.

Corollary 3. *The asymptotic lower bounds for Insdel LDCs in Theorems 1, 2, 3 (for arbitrary adversarial channels only), and the respective corollaries, also hold for Insdel LCCs.*

1.1.4 A stronger version of the lower bound

Our lower bounds above hold against adversarial channels, where the channel may first inspect the codeword and then introduce worst-case error. Our techniques, however, work for a more innocuous channel, namely one that is oblivious to both the codeword and the decoder. We formalize the stronger version of our results below. We believe in this form they may be more easily applicable to other settings.

Definition 3 (Channel)**.** *A channel $\mathfrak{C}$ for m-bit strings is a Markov chain on $\Omega = \{0,1\}^m$. Equivalently, it is a collection of distributions $\left\{\mathfrak{C}(s) \colon s \in \{0,1\}^m\right\}$ over $\{0,1\}^m$.*

We remark that this definition does allow the output of a channel to depend on its input. However, since it is fixed for any decoding algorithm, the following notion of "decodable on average" is well-defined. We note that a similar notion would not make sense for adversarial channels, as the channel can be adaptive to the decoding strategy.

Definition 4 (Channel decoding on average)**.** *A code $C \colon \{0,1\}^n \to \{0,1\}^m$ is (q, δ, ε)-*locally decodable on average for channel $\mathfrak{C}$ *if there exists a probabilistic algorithm* Dec *such that*

- *For every $i \in [n]$, we have*

$$\Pr_{\substack{x \in \{0,1\}^n \\ y \sim \mathfrak{C}(C(x))}} \left[\mathsf{Dec}(y, m, i) = x_i\right] \geq \frac{1}{2} + \varepsilon,$$

  where the probability is taken over the uniform random choice of $x \in \{0,1\}^n$, the randomness of $\mathfrak{C}$, and the randomness of Dec.

- Dec *makes at most q queries to y in each invocation.*

We now state our lower bound in the strongest form. Its proof can be obtained by inspecting the proof of Theorem 3 (for the private-key setting).

Theorem 5. *There exists a channel $\mathfrak{D}$ for m-bit strings such that:*

- *For every $s \in \{0,1\}^m$, $\Pr_{s' \sim \mathfrak{D}(s)}[\mathsf{ED}\left(s', s\right) > \delta \cdot 2m] < \mathsf{negl}(m)$.*
- *Let $C \colon \{0,1\}^n \to \{0,1\}^m$ be a code that is (q, δ, ε)-locally decodable on average for $\mathfrak{D}$. Then for $q \geq 3$ we have $m = \exp\left(\Omega_{q,\delta,\varepsilon}(n^{1/(2q-3)})\right)$.*

1.2 Overview of techniques

Here we give an informal overview of the key ideas and techniques used in our proofs. We always assume a non-adaptive decoder in the following discussion.

Prior strategies for Hamming LDC lower bounds We start by discussing the proof strategies in lower bounds for Hamming LDCs. Essentially all such proofs[2] begin by observing that the code needs to be *smooth* in the sense that for any target message bit, the decoder cannot query a specific index with very high probability. Using this property, one can show that if we represent the queries used by the decoder as edges in a hypergraph with m vertices, then for any target message bit the hypergraph contains a matching of size $\Omega(m/q)$. The key idea in the proof is now to analyze this matching, where one uses various tools such as (quantum) information theory [KT00, KdW04, Woo07], matrix hypercontractivity [BRdW08], combinatorial arguments [KT00, BCG20], and reductions from q-query to 2-query [Woo07, Woo12].

For our proofs, however, the matching turns out to be not the right object to look at. Indeed, by simply analyzing the matching it is hard to prove any strong lower bounds for $q \geq 3$, as evidenced by the lack of progress for Hamming LDCs. Intuitively, a matching does not capture the essence of insdel errors (e.g., position shifts), which are strictly more general and powerful than Hamming errors. Therefore, we instead need to look at a different object.

The Good queries For a q-query insdel LDC, the correct object turns out to be the set of all *good* q-tuples in the codeword that are potentially useful for decoding a target message bit. When we view the bits in the codeword as functions of the message, we define a q-tuple to be good for the i'th message bit if there exists a Boolean function $f : \{0,1\}^q \to \{0,1\}$ which can predict the i'th message bit with a non-trivial advantage (e.g., with probability at least $1/2 + \varepsilon/4$, see Definition 5), using these q bits. It is a straightforward application of information theory (e.g., Theorem 2 in [KT00]) that any q-tuple cannot be good for too many message bits. Therefore, intuitively, if we can show that any message bit requires a lot of good tuples to decode, then we can conclude that there must be many tuples and thus the codeword must be long. In the extreme case, if we can show that any message bit requires a *constant fraction* of all tuples to decode, then we can conclude that there can be at most a constant number of message bits, regardless of the length of the codeword.

Towards this end, we consider the effect of insdels on the tuples. Suppose the decoder originally queries some q-tuple A. After some insdels (e.g., deletions) the positions of the tuples will change, and the actual tuple the decoder queries using A now may correspond to some other tuple B in the original codeword. B may not be a good tuple, in which case it's not useful for decoding the message bit. However, since the decoder always succeeds with probability $1/2 + \varepsilon$ when the number of errors is bounded, the decoder should still hit good tuples with a decent probability (e.g., $3\varepsilon/2$). Intuitively, this already implies in some sense that there should be many good tuples, except that this depends on the decoder's probability distribution. For example, if the decoder queries one tuple with probability 1, then for any fixed error pattern one just needs to make sure that one specific tuple is good.

To leverage the above point, we turn to a probabilistic analysis and use random errors. Specifically, we carefully design a probability distribution on the insdel errors. For any q-tuple A, this distribution also induces another probability distribution for the q-tuple B which A corresponds to

[2] Except the proof in [GM12] which gives a lower bound for 3-query Hamming LDC in a special range of parameters.

in the original codeword. The key ingredient in all our proofs is to design the error distribution such that it ensures certain nice properties of the induced distribution of any q-tuple, which will allow us to establish our bounds. This can be viewed as a conceptual contribution of our work, as we have reduced the problem of proving lower bounds of insdel LDCs to the problem of designing appropriate error distributions.

Designing the insdel error distribution What is the best insdel error distribution for our proof? It turns out the ideal case for the induced distribution of a q-tuple is the uniform distribution. Indeed, the hitting property discussed above implies that for any message bit, there is at least one q-tuple in the support of the decoder's queries which would still be good with constant probability under the induced distribution. If we can design an error distribution such that for *any* q-tuple, the induced distribution is the uniform distribution on all q-tuples, this means that for any message bit, there are at least a constant fraction of all q-tuples that are good for this bit, which would in turn imply that there can be at most a constant number of message bits.

However, it appears hard to design an error distribution with the above property even for $q = 2$, since we have a bound on the total number of errors allowed, and errors allocated to one tuple will affect the number of errors available for other tuples. Instead, our goal is to design the error distribution such that the induced distribution of any q-tuple is as "close" to the uniform distribution as possible. We first illustrate our ideas for the case of $q = 2$.

The case of $q = 2$ A simple idea is to start with a random number (up to $\Omega(m)$) of deletions at the beginning of the codeword, we call this deletion *type* **1**. This results in a random shift of any pair of indices. However, a crucial observation is that the *distance* between any pair of indices stays the same (for a pair of indices $i, j \in [m]$, their distance is $|i - j|$), which makes the induced distribution far from being uniform. Indeed, under such error patterns the Hadamard code seems to be a good candidate for insdel LDC. This is because any codeword bit of the Hadamard code is the inner product of a vector $v \in \{0, 1\}^n$ with the message, and to decode the i'th message bit the decoder queries a pair of inner products for v and $v + e_i$ (e_i is the i'th standard basis vector) where v is a uniform vector. If we arrange the codeword bits in the natural lexicographical order according to v, then all pairs used in queries for the i'th message bit have a fixed distance of 2^{i-1}. In fact we show in the appendix that a simple variant of the Hadamard code does give a LDC under deletion type **1**. However, our Theorem 1 implies that it is not an insdel LDC in general. The point here is that we need a different operation to change the distance of any pair, which is a phenomenon unique to insdel LDC and never happens in Hamming LDC.

To achieve this, we introduce *random deletions* of each message bit on top of the previous operation. Specifically, imagine that we fix a constant $p < \delta$ and delete each bit of the codeword independently with probability p. Under this error distribution, any pair of queries with distance d will correspond to a pair with distance $\frac{d}{1-p}$ in expectation (since we expect to delete p fraction of bits in any interval). However, the independent deletions lead to a concentration around the mean. Thus the probability of any distance around $\frac{d}{1-p}$ is $\Theta(\frac{1}{\sqrt{d}})$ and the distribution resembles that of a binomial distribution (it is called a negative binomial distribution), which is not flat enough compared to the uniform distribution. Therefore, we add another twist by first picking the parameter p uniformly from an interval (e.g., $[\frac{\delta}{8}, \frac{\delta}{4}]$) and then delete each bit of the codeword independently with probability p. We call this deletion *type* **2**. Somewhat magically, the compound distribution now effectively "flattens" the original distribution, and we can show that the probability

mass of any distance is now $O(\frac{1}{d})$. Intuitively, this is because the distance in the induced distribution is now roughly equally likely to appear in the interval $[\frac{d}{1-\delta/8}, \frac{d}{1-\delta/4}]$. Combined with the deletions at the beginning, we can conclude the following two properties for any pair with original distance d in the induced distribution: (1) The probability mass of any element in the support is $O(\frac{1}{md})$, and (2) With high probability, the corresponding pair will have distance in $[d, cd]$ for some constant $c = c(\delta, \varepsilon)$ (See Lemma 3 for the formal statement).

While this is not exactly the uniform distribution, it is already enough to establish non-trivial bounds. To do this, we divide all pairs of queries into $O(\log m)$ intervals based on their distances, where the j'th interval P_j consists of all pairs with distance in $[c^{j-1}, c^j)$. By the hitting property discussed before, for any message bit, there is at least one q-tuple in the support of the decoder's queries which is still good with constant probability under the induced distribution. By (1) and (2) above, there must be at least $\Omega(md)$ good pairs with distance in $[d, cd]$, and this further implies that there exists a j such that P_j contains a constant fraction of good pairs. Now a packing argument implies that $n = O(\log m)$.

We remark that the random deletion channel (described above) that we use to establish the lower bound does not depend on anything about the codeword or the entire coding and decoding scheme. Thus, in contrast to the same bound for Hamming LDC, our lower bound continues to apply in private-key settings where the encoder and decoder share secret randomness.

Linear 2-query LDC The case of linear/affine codes is more involved. Here, we first use Fourier analysis to argue that if a pair of codeword bits is good for decoding a message bit, then the message bit must have non-trivial correlation with some parity of the codeword bits. However, since the code itself is linear or affine, this non-trivial correlation must be 1. By the hitting property, for any i'th message bit there exists a j_i such that a constant fraction of the pairs in P_{j_i} are good for i. By rearranging the message bits, without loss of generality we can assume that $j_1 \le j_2 \le \cdots \le j_n$.

Now, for any i and P_{j_i} we have two cases: the message bit can have correlation 1 either with a single codeword bit, or with the parity of the two codeword bits. By averaging, at least one case consists of a constant fraction of the pairs in P_{j_i}. By another averaging, at least a constant fraction of the message bits fall into one of the above cases, so eventually we have two cases: (a) a constant fraction of the message bits each has correlation 1 with a constant fraction of all codeword bits, or (b) a constant fraction of the message bits each has correlation 1 with the parity of a constant fraction of the pairs in P_{j_i}.

The first case is easy since any codeword bit cannot simultaneously have correlation 1 with two different message bits, hence this implies we can only have a constant number of message bits. The second case is harder, where we use a delicate combinatorial argument to reduce to the first case. Specifically, for any such message bit i we can consider the bipartite graph G_i on $2m$ vertices induced by the good pairs in P_{j_i}, thus any such graph has bounded degree (since the distance of the pairs is bounded) and is dense in the sense that the edges take up a constant fraction of all possible edges. For simplicity let us assume that having correlation 1 means that the two bits are the same as functions. Roughly, we use the dense property of these graphs to show the following: (c) there is an index $i = \Omega(n)$ and a right vertex $W \in G_i$ which is connected to a set T of $\Omega(c^{j_i})$ left vertices in G_i, and (d) there are $\Omega(n)$ indices $i' \le i$ such that in each $G_{i'}$, the same set T is connected to a set $U_{i'}$ of $\Omega(c^{j_i})$ neighbors. By (c), all the codeword bits in T must be the same, and they are all contained in an interval of length c^{j_i}. Then by (d), all the codeword bits in $U_{i'}$ for different i' must be disjoint, since the parity of them with some bits in T equals a different message bit. Now notice

that for any $i' \le i$, all pairs in $P_{j'_i}$ have distance at most $c^{j'_i} \le c^{j_i}$. This implies all the bits of all $U_{i'}$ are contained in an interval of length $2c^{j_i}$, which readily gives that $n = O(1)$.

The case of $q \ge 3$ We now generalize the above strategy to the case of $q \ge 3$. Consider the case of $q = 3$ for example. Now any query is a triple and we use (d_1, d_2) to stand for the distances of the two adjacent intervals in the query. If we can show similar properties as before, i.e., for any triple with distance (d_1, d_2) in the induced distribution: (1) The probability mass of any element is $O(\frac{1}{md_1d_2})$, and (2) With high probability, the corresponding triple will have distance (d'_1, d'_2) such that $d'_1 \in [d_1, cd_1], d'_2 \in [d_2, cd_2]$ for some constant $c = c(\delta, \varepsilon)$, then a similar argument would yield the bound of $n = O(\log^2 m)$, and for general q (at least constant q) the bound of $n = O(\log^{q-1} m)$.

However, unlike the case of $q = 2$, another tricky issue arises here. The issue is that with the error distribution discussed above, while we can ensure that for any *pair* of indices in the q-tuple, its marginal distribution behaves as before, the joint distribution of the q-tuple in the induced distribution behaves differently than what we expect. The reason is that (e.g., for $q = 3$) the two intervals with distance d_1 and d_2 are correlated under the error distribution. Specifically, the random deletion of each codeword bit again leads to a concentration phenomenon, thus conditioned on the number of deletions in the first interval, the parameter p is no longer uniformly distributed in the interval $[\frac{\delta}{8}, \frac{\delta}{4}]$, but rather pretty concentrated in a much smaller interval. This in turn affects the induced distribution of the second interval. Specifically, under this error distribution the bound on the probability in (1) becomes $O(\frac{\sqrt{d}}{md_1d_2})$, where $d = d_1 + d_2$. If we simply apply this bound, it will lead to (coincidentally or uncoincidentally) almost exactly the same bound as for Hamming LDC, thus we don't get any significant improvement.

To get around this and prove strong lower bounds for insdel LDCs, we introduce additional random deletion processes to "break" the correlations discussed above. Towards this, we add another layer of deletions on top of the previous two operations: we first divide the codeword evenly into blocks of size s, and then for each block, we independently pick a parameter p uniformly from $[\frac{\delta}{8}, \frac{\delta}{4}]$ and delete each bit of this block independently with probability p. The idea is that, if for a 3-query it happens that one block is completely contained in one interval, then since the deletion process in that block is independent of the other blocks, the induced distribution of that interval is also more or less independent of the other interval.

However, this comes with another tricky issue: how to pick the size s. If s is too large, then for queries with small intervals, both intervals can be contained in the same block, and the deletion process would be exactly the same as before, which defeats the purpose of using blocks. On the other hand, if s is too small, then for queries with large intervals, the concentration and correlation phenomenon will happen again, which also defeats the purpose of using blocks. Since the intervals of the queries can have arbitrary distance, our solution is to actually use $O(\log m)$ layers of deletions, where for the j'th layer we use a block size of say 2^j. This ensures that for any query there is an appropriate block size, and in the analysis we can first condition on the fixing of all other layers, and argue about this layer.

Yet there is another price to pay here: since we are only allowed at most δm deletions, in each layer we cannot delete each bit with constant probability. Therefore for these layers we need to pick a parameter p uniformly from $[\frac{\delta}{8 \log m}, \frac{\delta}{4 \log m}]$. We call this deletion *type* **3**. This blows up our bound of the probability in (1) by a polylog factor (see Corollary 4 for a formal statement), and we get a bound of $n = O(\log^{2q-3} m)$.

We note that in all the discussions so far, our error distributions do not depend on anything

about the codeword or the entire coding and decoding scheme, thus all these results apply in settings where the encoder and decoder share secret randomness (private-key), which makes our lower bounds stronger. On the other hand, by exploiting the decoder's strategy, we can actually improve our bounds for the case of $q \geq 3$ (but the improved lower bounds no longer hold in the private-key setting). This time, we add another $O(\log m)$ layers of deletions on top of the previous three operations, where for the j'th layer we again use a block size of say 2^j. However, for these $O(\log m)$ layers the deletion parameter p is not picked from $[\frac{\delta}{8\log m}, \frac{\delta}{4\log m}]$, but rather uniformly from $[\frac{\delta p_j}{8}, \frac{\delta p_j}{4}]$, where p_j is the probability that the decoder uses a query whose first interval has distance in $[2^{j-1}, 2^j)$. We call this deletion *type* **4**. Notice that since $\sum_j p_j = 1$ the expected number of total deletions for this operation is still at most $\frac{\delta m}{4}$.

To get some intuition of why this helps us, consider the extreme case where all the queries used by the decoder have exactly the same distance for the first interval. Since there is no other distance for the first interval, we should not assign any probability mass of deletions to blocks of a different size, but should instead use the same block size, and delete each bit with probability p chosen uniformly from say $[\frac{\delta}{8}, \frac{\delta}{4}]$. This corresponds to the case where some $p_j = 1$, and the above strategy is a natural generalization. In the meantime, we still need all previous deletion types to take care of the other intervals. We show that under this deletion process we can replace one $\log m$ factor in the probability of (1) by $1/p_j$ (see Corollary 5 for a formal statement), and overall this leads to a bound of $n = O(\log^{2q-4} m)$ for $q \geq 3$.

1.3 Open Problems

Better lower bounds and/or explicit constructions An immediate open problem concerns the existence of constant-query Insdel LDCs. While our lower bounds here are exponential in the message length, it is still conceivable that such Insdel LDCs do not actually exist, a conjecture raised in [BBG+20]. If on the other hand such codes do exist, it would be very intriguing to have better lower bounds and/or have explicit constructions, even for $q = O(\log n / \log\log n)$. Surprisingly, it appears to be highly non-trivial to construct such codes, which is in contrast to Hamming LDCs, where the simple Hadamard code is a classic example of a constant-query (in fact 2-query) LDC, and by now we have several constructions of explicit LDCs.

Relaxed Insdel LDCs/LCCs Relaxed (Hamming) LDCs/LCCs are variants in which the decoder is allowed some small probability of outputing a "don't know" answer, while it should answer with the correctly decoded bit most of the time. [BGH+06] proposed these variants and gave constructions with constant query complexity and codeword length $m = n^{1+\varepsilon}$. More recently [GRR18] extended the notion to LCCs, and proved similar bounds, which are tight [GL21]. An open problem here is to understand tight bounds for the relaxed insdel variants of LDCs/LCCs.

Larger alphabet size We believe our proofs generalize to larger alphabet sizes, and leave the precise bounds in terms of the alphabet size as an open problem. All the above directions may also be asked for larger alphabet sizes.

1.4 Further discussion about related work

[OPS07] gave private key constructions of LDCs with constant rate $m = \Theta(n)$ and locality polylog(n). [BKZ20] extended the construction from [OPS07] to settings where the sender/decoder do not share

randomness, but the adversarial channel is resource bounded i.e., there is a safe-function that can be evaluated by the encoder/decoder but not by the channel due to resource constraints (space, computation depth, etc.). By contrast, in the classical setting with no shared randomness and a computationally unbounded channel there are no known constructions with constant rate $m = \Theta(n)$ and locality polylog(n). [BB21] applied the [BBG$^+$20] compiler to the private key Hamming LDC of [OPS07] (resp. resource bounded LDCs of [BKZ20]) 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 [Lip94]. 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: for example, they might only behave like probabilistic polynomial time machines, or log space machines, or they may only corrupt codewords while being oblivious to the encoder's random coins, or they might have to deal with settings in which the sender and receiver exchange cryptographic keys. As expected, in many such limited-resource settings one can construct codes with strictly better parameters than what can be done generally [DGL04, MPSW05, GS16, SS16]. LDCs in these channels under Hamming error were studied in [OPS07, HO08, HOSW11, HOW15, BGGZ19, BKZ20].

[BB21] applied the [BBG$^+$20] compiler to the Hamming LDC of [BKZ20] to obtain a constant rate Insdel LDCs with polylog(n) locality for resource bounded channels. The work of [CLZ20] 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 [HS18], which constructs explicit synchronization strings that can be locally decoded.

Synchronization strings are powerful ingredients that have been used extensively in constructions of insdel codes. In fact, by combining locally decodable synchronization strings with Hamming LDCs, it seems possible to get similar reductions to Insdel LDCs as those in [OPS07, BBG$^+$20].

1.5 Organization

In Section 2 we give some basic notations and lemmas. In section 3, we show our lower bound for 2 query insdel LDCs. In Section 4, we describe more general error distributions and their induced properties. In Section 5 we show our lower bound for q-query insdel LDCs for the private key setting. In Section 6, we show our stronger lower bound for q-query insdel LDCs.

2 Notation and Preliminary Lemmas

Here we present some common notation and lemmas which we use throughout our proofs.

The indices i, j, k, ℓ are reserved for iterators; $c, \alpha, \beta, \gamma, \eta$ are reserved for constants; a, b, x, y, z are reserved for vectors or strings. For a string $y \in \{0,1\}^m$ and a subset $J \subseteq [m]$ of indices, we write $y_J \coloneqq \{y_j : j \in J\}$ for the restriction of y to J.

We may assume that decoder always queries exactly q indices. If some query uses a set of

indices $Q' \subset [m]$ such that $|Q'| = q' < q$, we can replace Q' by $Q = Q' \cup \{j_1, \ldots, j_{q'-q}\}$ where choices of $j_1, \ldots, j_{q'-q} \in [m] \setminus Q'$ are arbitrary. In the actual decoding, the decoder will just ignore the extra symbols. Given a tuple $\{k_0, \ldots, k_{q-1}\}$ with $k_0 < \cdots < k_{q-1}$, we also denote it by $(k_0, d_1, d_2, \ldots, d_{q-1})$ where $d_i = k_i - k_{i-1}$ for $i = 1, 2, \ldots, q-1$. Note that this induces a bijection $\psi_{m,q}$ between $\mathcal{S}_{m,q} = \{(k, d_1, \ldots, d_{q-1})\colon k, d_1, \ldots, d_{q-1} \geq 1, k + d_1 + \cdots + d_{q-1} \leq m\}$ and $\binom{[m]}{q}$. Sometimes we will abuse the notation and write $Q \subseteq [m]^q$ while we actually mean the image of Q under $\psi_{m,q}$ (e.g. when we write $A \cap B$ where $A \subseteq \binom{[m]}{q}$ and $B \subseteq \mathcal{S}_{m,q}$), and vice versa.

Given a distribution $\mathcal{D}$ over some space Ω, denote by $\mathrm{supp}(\mathcal{D}) = \{\omega \in \Omega\colon \mathcal{D}(\omega) > 0\}$ the *support* of $\mathcal{D}$.

All logs are in base 2 unless otherwise specified. We write $\mathcal{H}(x) = -x \log x - (1-x)\log(1-x)$ for the binary entropy function, and we use the following upper bound (Proposition 1). The proof can be obtained via expanding $\mathcal{H}(x)$ into Taylor series around $x = 1/2$.

Proposition 1. *For $x \in (0, 1/2)$, we have $\mathcal{H}(1/2 + x) \leq 1 - (2(\ln 2)^2/3)x^2$.*

Basic facts of Fourier analysis. We start with a Boolean function from $\{0,1\}^n \to \{0,1\}$ and transform it to the $\{1,-1\}^n \to \{1,-1\}$ space by the transformation $u \mapsto (-1)^u$ for any bit u in the input or output.

Let f, g be two Boolean functions from Fourier space. We define their correlation to be $\mathsf{Corr}(f,g) = |\mathbb{E}_x f(x)g(x)| = |\Pr_x[f(x) = g(x)] - \Pr_x[f(x) \neq g(x)]|$. For a function f, its Fourier expansion is $\sum_{S \subseteq [n]} \hat{f}_S \chi_S(x)$, where $\chi_S(x) = \prod_{i \in S} x_i$ and $\hat{f}_S = \langle f, \chi_S \rangle = \mathbb{E}_x f(x)\chi_S(x)$. By this definition, for Boolean functions f, we always have $|\hat{f}_S| \leq 1$, since $f(u), \chi_S(u) \in \{-1, 1\}$.

Proposition 2. *Let $f : \{-1,1\}^n \to \{-1,1\}$ and $C : \{-1,1\}^n \to \{-1,1\}^m$ be arbitrary functions. For every $Q \subseteq \binom{[m]}{q}$, if*

$$\sup_{S \subseteq Q} \left| \mathbb{E}_x f(x) \prod_{j \in S} y_j \right| < \frac{\varepsilon}{2^q},$$

where $y = C(x)$, then for any function $g : \{-1,1\}^q \to \{-1,1\}$, $\Pr_x\left[g(y_Q) = f(x)\right] < (1+\varepsilon)/2$.

Proof. We know that $g(y_Q) = \sum_{S \subseteq [q]} \hat{g}_S \chi_S(y_Q)$. So

$$\begin{aligned}
\left|\mathbb{E}_x g(y_Q) f(x)\right| &= \left| \mathbb{E}_x \sum_{S \subseteq [q]} \hat{g}_S \chi_S(y_Q) f(x) \right| \\
&= \left| \sum_{S \subseteq [q]} \mathbb{E}_x \hat{g}_S \chi_S(y_Q) f(x) \right| \\
&\leq \sum_{S \subseteq [q]} \left| \mathbb{E}_x \hat{g}_S \chi_S(y_Q) f(x) \right| \\
&\leq 2^q \sup_{S \subseteq [q]} \left| \mathbb{E}_x \hat{g}_S \chi_S(y_Q) f(x) \right| \\
&< 2^q \varepsilon / 2^q = \varepsilon.
\end{aligned}$$

So $\Pr_x\left[g(y_Q) = f(x)\right] < (1+\varepsilon)/2$. □

Our analysis is based on designing a specific error pattern and deriving the necessary properties the decoder needs to have in order to perform well against such errors. In a high level, the error pattern is going to be in the following form. Given a codeword $y \in \{0,1\}^m$, we first obtain the *augmented codeword* $y' \in \{0,1\}^{2m}$ by appending m bits to the end of y. These bits may be random, and most often they will be independent and uniformly random bits. Then the augmented codeword undergoes a random deletion process, which we describe in details later in Section 3 and Section 4. For now, think of it as generating a subset $D \subseteq [2m]$ according to some distribution $\mathcal{D}$, and then deleting all bits from y' with indices in D. Finally, the string output by the deletion process is truncated at length m to obtain the final output $\widetilde{y}$. We will argue that with high probability, $\widetilde{y}$ has length exactly m (i.e. there are at most m deletions in total) and is close to the original codeword y (i.e. only a small number of deletions are introduced to the first half of y').

One would observe that we could equivalently augment the codeword to length m *after* the deletion process, and indeed this gives the same distribution (if the padded bits are i.i.d.). However, it turns out that our argument becomes cleaner if we view the deletions as if they also occur in the augmented part. Specifically, in the following definition we view the augmented bits as part of the codeword, as it is possible that in some situation they actually help the decoder to decode some message bits.

Definition 5. *For $i \in [n]$, define the set Good_i as*

$$\mathsf{Good}_i \coloneqq \left\{ Q \in \binom{[2m]}{q} : \exists \text{a Boolean function } f\colon \{0,1\}^q \to \{0,1\} \text{ such that } \Pr[f(C'(x)_Q) = x_i] \geq \frac{1}{2} + \frac{\varepsilon}{4} \right\},$$

where the probability is over the uniform distribution of all messages and any possible randomness in the padded bits.

For $Q \in \binom{[2m]}{q}$, let $H_Q \subseteq [n]$ be a subset collecting all indices i for which Good_i contains Q. The following is a corollary to Theorem 2 in [KT00].

Proposition 3. $\forall Q \in \binom{[2m]}{q}, \left|H_Q\right| \leq q / \left(1 - \mathcal{H}(1/2 + \varepsilon/4)\right)$.

Proof. Let $I\left(\mathbf{x}_{H_Q}; C(\mathbf{x})_Q\right)$ denote the mutual information between $\mathbf{x}_{H_Q}$ and $C(\mathbf{x})_Q$. We have that

$$I\left(\mathbf{x}_{H_Q}; C(\mathbf{x})_Q\right) \leq \mathcal{H}\left(C(\mathbf{x})_Q\right) \leq q.$$

On the other hand,

$$\begin{aligned} I\left(\mathbf{x}_{H_Q}; C(\mathbf{x})_Q\right) &= \mathcal{H}\left(\mathbf{x}_{H_Q}\right) - \mathcal{H}\left(\mathbf{x}_{H_Q} \mid C(\mathbf{x})_Q\right) \\ &\geq \mathcal{H}\left(\mathbf{x}_{H_Q}\right) - \sum_{i \in H_Q} \mathcal{H}\left(x_i \mid C(\mathbf{x})_Q\right) \\ &\geq \left(1 - \mathcal{H}(1/2 + \varepsilon/4)\right) \cdot \left|H_Q\right|. \end{aligned}$$

Rearranging gives the result. □

A deletion pattern is a distribution $\mathcal{D}$ over subsets of $[2m]$. Let $D \subseteq [2m]$ be a set of deletions. We note that D induces a strictly increasing mapping $\phi_D\colon [2m - |D|] \to [2m]$, where $\phi_D(i) = \min\left\{ i' \in [2m] : \left|\overline{D} \cap [i']\right| \geq i \right\}$, or intuitively the index of i before the deletions are introduced.

Given $Q = \{k_0, \ldots, k_{q-1}\} \in \binom{[m]}{q}$, we denote $Q^D = \{\phi_D(k_0), \ldots, \phi_D(k_{q-1})\}$. Note that this is always well-defined when $|D| \leq m$. Most often we will work with a random $D \sim \mathcal{D}$ for some deletion pattern $\mathcal{D}$. In that case Q^D is a random variable, and sometimes we say that Q^D *corresponds to* Q under $\mathcal{D}$. If the event $Q^D \in \mathsf{Good}_i$ occurs, where Q is a random query of $\mathsf{Dec}(\cdot, m, i)$, we say that "$\mathsf{Dec}(\cdot, m, i)$ *hits* Good_i". In this paper this event will be independent of the string given to Dec since Dec is non-adaptive, and $\mathcal{D}$ will be oblivious to the codeword.

Lemma 1. *Given a (q, δ, ε) insdel LDC, for any deletion pattern $\mathcal{D}$ such that $|D \cap [m]| \leq \delta m$ and $|D| \leq m$ for any $D \in \mathrm{supp}(\mathcal{D})$, and any $i \in [n]$, the probability that $\mathsf{Dec}(\cdot, m, i)$ hits Good_i is at least $3\varepsilon/2$.*

Proof. Consider a uniformly random message $x \in \{0,1\}^n$ and $y = C(x) \in \{0,1\}^m$. Let $y' \in \{0,1\}^{2m}$ be an augment of y, and denote by y^D the string obtained by deleting from y' all bits with indices in D and truncating at length m. Formally, $y^D_j = y'_{\phi_D(j)}$ for $j = 1, \ldots, m$. Note that this is well defined if $|D| \leq m$.

Denote by $\mathcal{E}$ the event "$\mathsf{Dec}(\cdot, m, i)$ hits Good_i". Conditioned on $\overline{\mathcal{E}}$, the decoder successfully outputs x_i with probability at most $1/2 + \varepsilon/4$, by definition of Good_i (even in the case where the decoder may output a random function).

When $\left|D \cap [m]\right| \leq \delta m$, we have that $\mathsf{ED}(y, y^D) \leq \delta \cdot 2m$. By definition of a (q, δ, ε) insdel LDC, we have that

$$\begin{aligned}
\frac{1}{2} + \varepsilon &\leq \Pr\left[\mathsf{Dec}(y^D, m, i) = x_i\right] \\
&\leq \Pr\left[\mathsf{Dec}(y^D, m, i) = x_i \mid \mathcal{E}\right] \cdot \Pr\left[\mathcal{E}\right] + \Pr\left[\mathsf{Dec}(y^D, m, i) = x_i \mid \overline{\mathcal{E}}\right] \cdot \Pr\left[\overline{\mathcal{E}}\right] \\
&\leq \Pr\left[\mathcal{E}\right] + \left(\frac{1}{2} + \frac{\varepsilon}{4}\right) \cdot \left(1 - \Pr\left[\mathcal{E}\right]\right).
\end{aligned}$$

All probabilities above are over x, D, the randomness of the decoder and any possible randomness in the padded bits. Rearranging gives $\Pr[\mathcal{E}] \geq 3\varepsilon/(2-\varepsilon) \geq 3\varepsilon/2$. □

We will write $\mathbf{U}[a, b]$ for the uniform distribution over the interval $[a, b]$. For $n \in \mathbb{N}$ and $p \in [0, 1]$, we will write $B(n, p)$ for the binomial distribution with n trials and success probability p. When p is a random variable with distribution $\mathcal{D}$, we will denote the resulting compound distribution by $B(n, \mathcal{D})$.

We use the following anti-concentration bound for the compound distribution $B(n, \mathbf{U}[a, b])$.

Lemma 2. *Let $n \in \mathbb{N}$, and $0 \leq s < t \leq 1$. Let X be a random variable following a compound distribution $B(n, \mathbf{U}[s, t])$. Then for any $0 \leq k \leq n$, we have*

$$\Pr[X = k] \leq \frac{1}{(t-s)(n+1)}.$$

Proof. We can explicitly write the probability as

$$\Pr[X = k] = \frac{1}{t-s} \int_s^t \binom{n}{k} x^k (1-x)^{n-k} \, \mathrm{d}x \leq \frac{1}{t-s} \int_0^t \binom{n}{k} x^k (1-x)^{n-k} \, \mathrm{d}x.$$

Denoting

$$I_k = \binom{n}{k} \int_0^t x^k (1-x)^{n-k} \, \mathrm{d}x,$$

we are going to show that $I_k \leq 1/(n+1)$. Integration by parts gives

$$\begin{aligned}
I_k &= \frac{1}{k+1}\binom{n}{k}\left(x^{k+1}(1-x)^{n-k}\Big|_0^t + (n-k)\int_0^t x^{k+1}(1-x)^{n-k-1}\,\mathrm{d}x \right) \\
&= \frac{1}{k+1}\binom{n}{k} t^{k+1}(1-t)^{n-k} + \frac{n-k}{k+1}\binom{n}{k}\int_0^t x^{k+1}(1-x)^{n-k-1}\,\mathrm{d}x \\
&= \frac{1}{n+1}\binom{n+1}{k+1} t^{k+1}(1-t)^{n-k} + \binom{n}{k+1}\int_0^t x^{k+1}(1-x)^{n-k-1}\,\mathrm{d}x \\
&= \frac{1}{n+1}\binom{n+1}{k+1} t^{k+1}(1-t)^{n-k} + I_{k+1}.
\end{aligned}$$

Therefore by telescoping and the Binomial Theorem, we have

$$I_k = \sum_{j=k}^{n} (I_k - I_{k+1}) = \frac{1}{n+1}\sum_{j=k+1}^{n+1} \binom{n+1}{j} t^j (1-t)^{n+1-j} \leq \frac{1}{n+1}.$$

□

3 Bounds for 2-query Insdel LDCs

In this section, we prove lower bounds for 2-query insdel LDCs (Theorem 1 and Theorem 2).

We start by describing the error pattern. It is defined via the following random deletion process which is applied to the augmented codeword described in the last section i.e., we obtain the augmented codeword by appending m bits to the end of the codeword. Recall that after applying the random deletions below we can always truncate the final string back down to m bits.

Description of the error distribution

Step 1 Pick a real number $\beta \in [\frac{\delta}{8}, \frac{\delta}{4}]$ uniformly at random and then delete each bit $j \in [2m]$ independently with probability β.

Step 2 Pick an integer $e_2 \in \left\{0, 1, \ldots, \left\lfloor \frac{\delta m}{4} \right\rfloor\right\}$ uniformly at random and delete the first e_2 bits.

We remark that equivalently, the process can be thought of as maintaining a subset $D \subseteq [2m]$ of deletions and updating D in each step, and nothing is really deleted until the end of the process. We will sometimes take this view in later discussions. However, for readability we omitted the details as to how this set is updated.

The following proposition bounds the number of deletions introduced by the process.

Proposition 4. *Let $D \subseteq [2m]$ be a set of deletions generated by the process. Then we have*

- $\Pr\left[|D \cap [m]| > \delta m\right] \leq 2^{-\Omega(m)}$,
- $\Pr\left[|D| > m\right] \leq 2^{-\Omega(m)}$.

Proof. Let $D_1 \subseteq D$ be the subset of deletions introduced during Step 1. Since Step 2 introduces at most $\delta m/4$ deletions, it suffices to upper bound the probabilities of $\left|D_1 \cap [m]\right| > 3\delta m/4$ and $|D_1| > 3m/4$. Moreover, it suffices to prove the upper bounds for any fixed $\beta \in [\delta/8, \delta/4]$ picked in Step 1.

For the first item, notice that each bit $j \in [m]$ is deleted independently with probability $\beta \leq \delta/4$. Thus by Hoeffding's inequality

$$\Pr\left[\left|D_1 \cap [m]\right| \geq \left(\frac{\delta}{4} + \frac{\delta}{2}\right) m\right] \leq \exp\left(-\frac{\delta^2 m}{2}\right) = 2^{-\Omega(m)}.$$

The proof of the second item follows similarly from Hoeffding's inequality

$$\Pr\left[|D_1| \geq \frac{3m}{4}\right] \leq \Pr\left[|D_1| \geq \left(\frac{\delta}{4} + \frac{1}{8}\right) \cdot 2m\right] \leq \exp\left(-2\left(\frac{1}{8}\right)^2 \cdot 2m\right) = 2^{-\Omega(m)}.$$

□

In the following lemma, we fix an arbitrary query $\{k, \ell\} \in \binom{[m]}{2}$ of the decoder, with $k < \ell$, and represent it as (k, d) where $d = \ell - k$.

Let $(k', d') \in [2m] \times [2m]$ be the random pair that corresponds to (k, d) under the error distribution (see the discussion before Lemma 1). It should be clear that we always have $k' \geq k$ and $d' \geq d$. We prove some properties of the distribution of (k', d').

Lemma 3. *There exist two constants $c = c(\varepsilon) > 1$ and $c' = c'(\delta) > 0$ such that the following holds.*

- *The distribution of (k', d') is concentrated in the set $[2m] \times [d, cd]$ with probability $1 - \varepsilon$.*
- *Any support of (k', d') has probability at most $\frac{c'}{md}$.*

Proof. We prove the concentration result first. We will fix an arbitrary $\beta \in [\delta/8, \delta/4]$. By Hoeffding's inequality, we can take $c_0 = \sqrt{\ln(1/\varepsilon)/2}$ such that for any $n \in \mathbb{N}$ and $p \in [0, 1]$,

$$\Pr_{Y \sim B(n,p)}\left[Y \geq pn + c_0\sqrt{n}\right] \leq \varepsilon.$$

Take $c = 8c_0^2 = 4\ln(1/\varepsilon)$. Then $c > 1 + \left(c_0/(1-\beta)\right)^2$ for any $\beta \leq \delta/4 < 1/2$. Let X denote the number of deletions occurred in $[d+1, cd]$, which follows a binomial distribution $B((c-1)d, \beta)$. Then by the choice of c_0 we have

$$\Pr\left[d' > cd\right] \leq \Pr\left[X \geq (c-1)d\right] \leq \Pr\left[X \geq \beta(c-1)d + c_0\sqrt{(c-1)d}\right] \leq \varepsilon.$$

Note that this holds for any choice of $\beta \leq \delta/4$, and thus the concentration result follows.

Now we turn to the anti-concentration result. Denote by $k' \mapsto k$ the event that the k'-th bit is retained and has index k after the deletion, and denote by $\left(k', d'\right) \mapsto (k, d)$ the event $\left(k' \mapsto k\right) \wedge \left(k' + d' \mapsto k + d\right)$.

Write $\Pr_{S_1}[\cdot]$ for the error distribution after Step 1. Let X be the number deletions occurred in $\{k'+1,\ldots,k'+d'-1\}$, which follows a compound distribution $B(d'-1, \mathbf{U}[\delta/8, \delta/4])$. We have

$$\begin{aligned}
\sum_{k''=0}^{k'} \Pr_{S_1}\left[(k',d') \mapsto (k'',d)\right] &= \sum_{k''=0}^{k'} \Pr_{S_1}\left[k' \mapsto k''\right] \cdot \Pr_{S_1}\left[k'+d' \mapsto k''+d \;\middle|\; k' \mapsto k''\right] \\
&= \sum_{k''=0}^{k'} \Pr_{S_1}\left[k' \mapsto k''\right] \cdot \Pr_{S_1}\left[X = d'-d\right] \cdot \Pr_{S_1}\left[k'+d' \text{ is retained}\right] \\
&\le \Pr_{S_1}\left[k' \text{ is retained}\right] \cdot \frac{8}{\delta} \cdot \frac{1}{d'} \\
&\le \frac{8}{\delta} \cdot \frac{1}{d'}.
\end{aligned}$$

Here the first inequality is due to Lemma 2. Finally, averaging over e_2 gives

$$\begin{aligned}
\Pr\left[(k',d') \mapsto (k,d)\right] &= \frac{8}{\delta m} \sum_{e_2=0}^{\delta m/8} \Pr_{S_1}\left[(k',d') \mapsto (k+e_2,d)\right] \\
&\le \frac{8}{\delta m} \sum_{k''=0}^{k'} \Pr_{S_1}\left[(k',d') \mapsto (k'',d)\right] \\
&\le \frac{8}{\delta m} \cdot \frac{8}{\delta} \cdot \frac{1}{d} \\
&= \frac{64}{\delta^2} \cdot \frac{1}{md}.
\end{aligned}$$

Therefore we can take $c' = 64/\delta^2$. □

Before proceeding to prove the main theorems, we provide a dictionary of notations to facilitate the readers.

Notations. The sets S_i, S, T, U_i, V_i are subsets of $[m]$, where S_i, S, T, U_i are used to denote some set of the first indices (namely k for a pair $\{k,\ell\}$ with $k<\ell$), and V_i is used to denote some set of the second indices (namely ℓ for a pair $\{k,\ell\}$ with $k<\ell$). We have the following relation: $\forall i, U_i \subseteq T \subseteq S$.

The set Good_i is defined in Definition 5. The sets P_j, Q_i are subsets of $[2m]\times[2m]$, i.e., subsets of the pairs of indices that may or may not be used in the query. j is reserved for the index of some P_j.

The set $G_{k,i}$ is a subset of $[n]$, i.e., a subset of some indices of the message bits.

We recall the statement of our main theorem for 2-query linear insdel LDC.

Theorem 1. *For any $(2,\delta,\varepsilon)$ linear or affine insdel LDC $C : \{0,1\}^n \to \{0,1\}^m$, we have $n = O_{\delta,\varepsilon}(1)$.*

To prove this theorem we first establish the following claim, which works for any $(2,\delta,\varepsilon)$ insdel (even non-linear/affine) LDC. Let c be the constant from Lemma 3. Consider all pairs of the form (k,d) in $[2m]\times[2m]$, and partition them into $t = \lceil \log_c(2m) \rceil = O(\log m)$ subsets $\{P_j\}$, where for any $j \in [t]$, $P_j = [2m] \times [c^{j-1}, c^j)$.

Claim 1. *There exists a constant $\gamma = \gamma(\delta, \varepsilon) \le 1$ such that the following holds for any $(2, \delta, \varepsilon)$ insdel LDC. For any $i \in [n]$, there exists a $j \in [t]$ such that $|P_j \cap \mathsf{Good}_i| \ge \gamma m c^j$.*

Proof. Fix any $i \in [n]$. Let $D \subseteq [2m]$ be a random set of deletions generated by the random process. By Proposition 4, with probability $1 - 2^{-\Omega(m)} \ge 1 - \varepsilon/4$ for any large enough n (and thus also m), we have that $\big|D \cap [m]\big| \le \delta m$ and $|D| \le m$. Conditioned on this event, $\mathsf{Dec}(\cdot, m, i)$ hits Good_i with probability at least $3\varepsilon/2$ by Lemma 1. Therefore, unconditionally the probability that $\mathsf{Dec}(\cdot, m, i)$ hits Good_i is at least $3\varepsilon/2 - \varepsilon/4 = 5\varepsilon/4$

(if $|D| > m$ we simply assume that $\mathsf{Dec}(\cdot, m, i)$ never hits Good_i). This implies that for at least one (k, d) in the support of the queries of $\mathsf{Dec}(\cdot, m, i)$, the corresponding pair (k', d') hits Good_i with probability at least $5\varepsilon/4$.

Now by the first item of Lemma 3 and a union bound, $\mathsf{Dec}(\cdot, m, i)$ queries a pair in $\mathsf{Good}_i \cap \big([2m] \times [d, cd]\big)$ with probability at least $5\varepsilon/4 - \varepsilon = \varepsilon/4$. By the second item of Lemma 3, we must have

$$\Big|\mathsf{Good}_i \cap \big([2m] \times [d, cd]\big)\Big| \ge \frac{\varepsilon m d}{4c'}.$$

Choose j' such that $c^{j'-1} \le d < c^{j'}$. Noticing that $[2m] \times [d, cd] \subseteq P_{j'} \cup P_{j'+1}$, for some $j \in \big\{j', j'+1\big\}$ we must have $\big|\mathsf{Good}_i \cap P_j\big| \ge \varepsilon m d/(8c')$. Since $d \ge c^{j'-1} \ge c^{j-2}$, we can choose $\gamma = \varepsilon/(8c'c^2)$ and the claim follows. □

By the definition of Good_i and Proposition 2, if a pair $\{k, \ell\} \in \mathsf{Good}_i$ $(k < \ell)$, then one of the following cases must happen: (1) x_i has correlation at least $\varepsilon/8$ with y_k; (2) x_i has correlation at least $\varepsilon/8$ with y_ℓ; and (3) x_i has correlation at least $\varepsilon/8$ with $y_k \oplus y_\ell$. However, notice that the code is a linear or affine code, thus every bit in $C(x)$ is a linear or affine function of x, which has correlation either 1 or 0 with any x_i. Furthermore the inserted bits are independent, uniform random bits. Therefore in any of these cases, the correlation must be 1 and the bits involved must not contain any inserted bit.

Thus, for any $i \in [n]$ and the corresponding $j \in [t]$ guaranteed by Claim 1, by averaging we also have three cases: (1) P_j has at least $\gamma m c^j/4$ pairs such that the first bit has correlation 1 with x_i; (2) P_j has at least $\gamma m c^j/4$ pairs such that the second bit has correlation 1 with x_i; and (3) P_j has at least $\gamma m c^j/2$ pairs such that the parity of the pair of bits has correlation 1 with x_i.

By another averaging, we now have two cases: either (a) at least $n/4$ of the message bits fall into case (1) or (2) above, or (b) at least $n/2$ of the message bits fall into case (3) above. We prove Theorem 1 in each case.

Proof of Theorem 1 in case (a). In this case, without loss of generality assume that there is a subset $I \subseteq [n]$ with $|I| \ge n/4$ such that for any $i \in I$, the corresponding P_j has at least $\gamma m c^j/4$ pairs such that the first bit has correlation 1 with x_i. Notice that any bit in $C(x)$ can be the first bit for at most c^j pairs in P_j, this means that there must be at least $\gamma m/4$ different bits in $C(x)$ that has correlation 1 with x_i. Let this set be V_i and we have $|V_i| \ge \gamma m/4$.

Since for each $i \in I$ we have such a set V_i, and these sets must be disjoint (a bit cannot simultaneously have correlation 1 with x_i and $x_{i'}$ if $i \ne i'$), we have

$$\frac{n}{4} \cdot \frac{\gamma m}{4} \le \sum_{i \in I} |V_i| = \left|\bigcup_{i \in I} V_i\right| \le m.$$

This gives $n \leq 16/\gamma = O_{\delta,\varepsilon}(1)$. □

Proof of Theorem 1 in case (b). This is the harder part of the proof. Here, there is a subset $I \subseteq [n]$ with $|I| \geq n/2$ such that for any $i \in I$, the corresponding P_j has at least $\gamma m c^j/2$ pairs such that the parity of the pair of bits has correlation 1 with x_i. For each $i \in I$, let the set of these pairs be Q_i. Thus $|Q_i| \geq \gamma m c^{j_i}/2$, where for any $i \in I$, j_i is the corresponding index of P_j guaranteed by Claim 1. Let $|I| = n' \geq n/2$. By rearranging the message bits if necessary, without loss of generality we can assume that $I = [n']$ and $j_1 \leq j_2 \leq \cdots \leq j_{n'}$. Let S_i be the set of all first indices of Q_i which are connected to at least $\gamma c^{j_i}/4$ second indices. Formally, $S_i = \{k : \left|\{d : (k,d) \in Q_i\}\right| \geq \gamma c^{j_i}/4\}$.

Another way to view this is to consider the bipartite graph $G_i = \big([m], [m], Q_i\big)$ (since the pairs in Q_i can only involve bits in $C(x)$). Then G_i has at least $\gamma m c^{j_i}/2$ edges, and the left and right degrees of G_i are both at most c^{j_i}. Now S_i is the subset of left vertices with degree at least $\gamma c^{j_i}/4$.

We have the following claim.

Claim 2. *For any $i \in [n']$, $|S_i| \geq \gamma m/4$.*

Proof. Since $|Q_i| \geq \gamma m c^{j_i}/2$, and each index in S_i is connected to at most c^{j_i} other indices, the claim follows by a Markov type argument. □

Now, for any index $k \in [m]$ and any index $i \in [n']$, we define the set $G_{k,i}$ to be the set of all indices $i' \leq i$ such that $k \in S_{i'}$. Formally, $G_{k,i} = \{i' \leq i : k \in S_{i'}\}$. We have the following claim:

Claim 3. *There exists a constant $\eta = \eta(\delta,\varepsilon) = \gamma/8$, an index $i \in [n']$ and a set $S \subseteq S_i$, such that*

- *$|S| \geq \eta m$.*
- *For any $k \in S$, we have $|G_{k,i}| \geq \eta n'$.*

Proof. First notice that $\sum_{k\in[m]} |G_{k,n'}| = \sum_{i\in[n']} |S_i|$. For a pair (i,k) with $i \in [n']$ and $k \in [m]$, we say it is good if $k \in S_i$ and $|G_{k,i}| \geq \gamma n'/8$. For any fixed $k \in [m]$, there are at least $|G_{k,n'}| - \gamma n'/8$ indices $i \in [n']$ such that (i,k) is good (this number may be negative, but that's still fine for us). To see this, let i^* be the smallest index such that $\left|G_{k,i^*}\right| = \gamma n'/8$ and notice that $|G_{k,n'}| - \gamma n'/8 = |G_{k,n'}| - |G_{k,i^*}|$ counts the number of i such that $i^* < i \leq n'$ and $k \in S_i$, i.e. the number of good pairs.

Therefore the total number of good pairs is at least

$$\sum_{k\in[m]} \left(|G_{k,n'}| - \frac{\gamma n'}{8}\right) = \sum_{i\in[n']} |S_i| - \frac{\gamma m n'}{8} \geq \frac{\gamma m n'}{8},$$

since for any $i \in [n']$, we have $|S_i| \geq \gamma m/4$.

By averaging, this implies that $\exists i \in [n']$, such that there are at least $\gamma m/8$ good pairs for this fixed i. Let S be the set of all good indices of k for this i, then we must have $|S| \geq \gamma m/8$ and for any $k \in S$, we have $k \in S_i$ and $|G_{k,i}| \geq \gamma n'/8$. Thus the claim holds. □

Now consider the index i and the set S guaranteed by the above claim. Recall j_i is the index j of P_j corresponding to i. We have the following claim.

Claim 4. *There exists a set $T \subseteq S$ and two indices $k_0, \ell_0 \in [m]$ such that the following holds.*

- *$|T| \geq \frac{\eta\gamma c^{j_i}}{4}$.*

- $T \subseteq [k_0, k_0 + c^{j_i}]$.
- $\forall k \in T$, $C(x)_k \oplus C(x)_{\ell_0}$ *has correlation* 1 *with* x_i.

Proof. Consider all pairs of indices $\{k, \ell\} \in Q_i$ $(k < \ell)$ with $k \in S$, and view it as a bipartite graph $G = (A, B, E)$ with indices k on the left, and indices ℓ on the right. Formally, $G = (A, B, E)$ with $A = \{a_1, \ldots, a_m\}$, $B = \{b_1, \ldots, b_m\}$ and edge $E = \{(a_k, b_\ell) : \{k, \ell\} \in Q_i, k < \ell\}$. Since for any $k \in S$, we have $k \in S_i$, we know that any a_k has degree at least $\gamma c^{j_i}/4$. Notice that there are m right vertices in B. Therefore there must exist an $\ell_0 \in [m]$ such that the node b_{ℓ_0} is connected to at least $\eta\gamma c^{j_i}/4$ vertices on the left, and we can let the set of all these vertices be $T = \{k : (a_k, b_{\ell_0}) \in E\}$. Since for any pair in Q_i, the parity of this pair of bits has correlation 1 with x_i, we have that $C(x)_k \oplus C(x)_{\ell_0}$ has correlation 1 with x_i for all $k \in T$.

Since the vertices in T are all connected to ℓ_0, and the distance $d = \ell - k$ for all pairs in Q_i is in $[c^{j_i - 1}, c^{j_i}]$, we must have that all indices $k \in T$ are in the range $[\ell_0 - c^{j_i}, \ell_0]$. Taking $k_0 = \ell_0 - c^{j_i}$ and the claim follows. □

Now for any $i' \le i$, let $U_{i'} = S_{i'} \cap T$, and consider the set $V_{i'}$ of all indices $\ell \in [m]$ such that $\exists k \in U_{i'}$ with $\{k, \ell\} \in Q_{i'}$. In other words, $V_{i'}$ is set of neighbours of $U_{i'}$ in the bipartite graph $\big([m], [m], Q_{i'}\big)$. We have the following claim.

Claim 5. *For any* $i' \le i$, *we have*

- $V_{i'} \subseteq [k_0, k_0 + 2c^{j_i}]$.
- $|V_{i'}| \ge \gamma |U_{i'}|/4$.

Proof. Since $U_{i'} \subseteq T$, and every pair of query in $Q_{i'}$ has distance at most $c^{j_{i'}} \le c^{j_i}$, we have $V_{i'} \subseteq [k_0, k_0 + 2c^{j_i}]$. Furthermore, since every index in $U_{i'}$ is connected to at least $\gamma c^{j_{i'}}/4$ indices in $V_{i'}$, while every index in $V_{i'}$ is connected to at most $c^{j_{i'}}$ indices in $U_{i'}$, we must have $|V_{i'}| \ge \gamma |U_{i'}|/4$. □

Now, notice that for any $i' \le i$, and any $\ell \in V_{i'}$, there exists some $k \in U_{i'} \subseteq T$ such that $C(x)_k \oplus C(x)_\ell$ has correlation 1 with $x_{i'}$. By Claim 4, $C(x)_k \oplus C(x)_{\ell_0}$ has correlation 1 with x_i. Thus $C(x)_\ell \oplus C(x)_{\ell_0}$ has correlation 1 with $x_i \oplus x_{i'}$, and $C(x)_\ell$ has correlation 1 with $x_i \oplus x_{i'} \oplus C(x)_{\ell_0}$. This means that for any two $i_1, i_2 \le i$ with $i_1 \ne i_2$, we must have $V_{i_1} \cap V_{i_2} = \emptyset$. Therefore, all the $V_{i'}$'s for different i' must be disjoint. Thus we have the following inequality:

$$\frac{\gamma}{4}\left(\sum_{i' \le i} |U_{i'}|\right) \le \sum_{i' \le i} |V_{i'}| \le 2c^{j_i}.$$

Notice that $\sum_{k \in T} |G_{k,i}| = \sum_{i' \le i} |S_{i'} \cap T| = \sum_{i' \le i} |U_{i'}|$ and $\forall k \in T \subseteq S$, we have $|G_{k,i}| \ge \eta n'$. Thus

$$\sum_{i' \le i} |U_{i'}| \ge \eta n' |T| \ge \frac{\eta^2 \gamma c^{j_i}}{4} n'.$$

Combining the two inequalities, we get $n' \le 32/(\eta^2\gamma^2) = 2048/\gamma^4$. Since $n' \ge n/2$. This also implies that $n \le 2n' = 4096/\gamma^4 = O_{\delta,\varepsilon}(1)$. □

Next we prove a simple exponential lower bound for general 2-query insdel LDCs, i.e. Theorem 2. This should serve as a warm-up for the general $q \ge 3$ case.

Theorem 2. *For any $(2, \delta, \varepsilon)$ insdel LDC $C : \{0,1\}^n \to \{0,1\}^m$, we have $m = \exp(\Omega_{\delta,\varepsilon}(n))$.*

Proof. Recall that $t = \lceil \log_c(2m) \rceil$ and $P_j = [2m] \times [c^{j-1}, c^j)$ for $j \in [t]$. For $j \in [t]$ and $i \in [n]$, we define $\beta_{j,i} = \frac{|P_j \cap \mathsf{Good}_i|}{|P_j|}$. Since $|P_j| = 2m(c^j - c^{j-1}) \le 2mc^j$, by Claim 1 there is a constant $\gamma = \gamma(\delta, \varepsilon) < 1$ such that for any $i \in [n]$, there exists a $j \in [t]$ satisfying $\beta_{j,i} \ge \gamma$. By the Pigeonhole Principle, there exists a $j \in [t]$ such that $\beta_{j,i} \ge \gamma$ for at least n/t different i's. Fix this j to be j_0. We have

$$\sum_{i=1}^{n} \beta_{j_0,i} \ge \frac{\gamma n}{t}.$$

On the other hand, by Proposition 3 every pair (k, d) can belong to Good_i for at most $2/(1 - \mathcal{H}(1/2 + \varepsilon/4))$ different i's. Thus we have

$$\sum_{i=1}^{n} |P_{j_0} \cap \mathsf{Good}_i| \le \frac{2}{1 - \mathcal{H}(1/2 + \varepsilon/4)} \cdot |P_{j_0}|.$$

Altogether this yields

$$\frac{\gamma n}{t} \le \sum_{i=1}^{n} \beta_{j_0,i} = \sum_{i=1}^{n} \frac{|P_{j_0} \cap \mathsf{Good}_i|}{|P_{j_0}|} \le \frac{2}{1 - \mathcal{H}(1/2 + \varepsilon/4)}.$$

We have $n \le O_{\delta,\varepsilon}(t) = O_{\delta,\varepsilon}(\log m)$ and $m = \exp\left(\Omega_{\delta,\varepsilon}(n)\right)$. □

4 A More General Error Distribution

In this section we describe a general framework for designing error distributions, and instantiate it with two sets of parameters. The error distribution defined in this section will be used in the proof of Theorem 3. As before the error distribution is applied to the augmented codeword which is obtained by concatenating m bits to the end of the original codeword — the final codeword can be truncated back down to m bits after applying the random deletions below.

Given parameters $L \in \mathbb{N}$, $\mathbf{s} = (s_1, \ldots, s_L) \in [2m]^L$ and $\mathbf{h} = (h_1, \ldots, h_L) \in [0,1]^L$ such that

$$h \coloneqq \sum_{\ell=1}^{L} h_\ell \le \frac{1}{4},$$

we consider an error distribution $\mathcal{D}(L, \mathbf{s}, \mathbf{h})$ defined by the following process.

Description of the error distribution $\mathcal{D}(L, \mathbf{s}, \mathbf{h})$

Step 1 The first step introduces deletions through L layers. For the ℓ-th layer, we first divide $[2m]$ into $\lceil 2m/s_\ell \rceil$ consecutive blocks each of size s_ℓ, except for the last block which may have smaller size. For the b-th block in layer ℓ, we pick $q_{\ell,b} \in [0, h_\ell \delta]$ uniformly at random (independent of other blocks), and mark each bit in the block independently with probability $q_{\ell,b}$. Finally, we delete all bits which are marked at least once.

Step 2 Pick $\beta \in [0, \frac{1}{4}]$ uniformly at random and delete each bit independently with probability $\beta\delta$.

Step 3 Pick an integer $e_2 \in \left\{0, 1, \ldots, \left\lfloor \frac{\delta m}{4} \right\rfloor\right\}$ uniformly at random and delete the first e_2 bits.

By a union bound, after Step 1, each symbol is deleted with probability at most $h\delta$. We thus have the following proposition as an easy consequence of Hoeffding's inequality.

Proposition 5. *Let $D \subseteq [2m]$ be a set of deletions generated by $\mathcal{D}(L, \mathbf{s}, \mathbf{h})$. Then we have*

$$\Pr\left[\left|D \cap [m]\right| > \delta m\right] \le \exp\left(-\frac{\delta^2 m}{8}\right), \text{ and } \Pr\left[|D| > m\right] \le \exp\left(-(1-\delta)^2 m\right).$$

Proof. Let $D_2 \subseteq D$ be the subset of deletions introduced during Step 1 and Step 2. Since Step 3 introduces at most $\delta m/4$ deletions, it suffices to upper bound the probabilities of $\left|D_2 \cap [m]\right| > 3\delta m/4$ and $|D_2| > m$. Moreover, it suffices to prove the upper bounds after conditioned on an arbitrary set of deletion probabilities $q_{\ell,b} \in [0, h_\ell \delta]$ for each $\ell \in [L]$ and $b \le \left\lceil 2m/s_\ell \right\rceil$, and $\beta \in [0, 1/4]$.

Under the conditional distribution, each bit $j \in [2m]$ is deleted with probability at most $(h+\beta)\,\delta \le \delta/2$, and these deletions are independent of each other. The Hoeffding's inequality shows that

$$\Pr\left[\left|D_2 \cap [m]\right| > \left(\frac{\delta}{2} + \frac{\delta}{4}\right) m\right] \le \exp\left(-2\left(\frac{\delta}{4}\right)^2 m\right) = \exp\left(-\frac{\delta^2 m}{8}\right),$$
$$\Pr\left[|D_2| > m\right] = \Pr\left[|D_2| > \left(\frac{\delta}{2} + \frac{1-\delta}{2}\right) \cdot 2m\right] \le \exp\left(-(1-\delta)^2 m\right).$$

□

We fix an arbitrary query $Q = \left(k, d_1, \ldots, d_{q-1}\right)$ of the decoder, and let $\left(k', d'_1, \ldots, d'_{q-1}\right) \in [2m]^q$ be the random tuple that corresponds to Q under the error distribution $\mathcal{D}(L, \mathbf{s}, \mathbf{h})$ (see the discussion before Lemma 1). It should be clear that we always have $k' \ge k, d'_1 \ge d_1, \ldots, d'_{q-1} \ge d_{q-1}$.

Given the query Q, we can define for each $i \in [q-1]$ a subset $F_i \subseteq [L]$ of layers as

$$F_i = \left\{\ell \in [L] \colon h_\ell \ne 0 \text{ and } \frac{d_i}{4} \le s_\ell \le \frac{d_i}{2}\right\}.$$

The following lemma is a generalization of Lemma 3.

Lemma 4. *Suppose that $F_i \ne \varnothing$ for each $i = 2, 3, \ldots, q-1$. The following propositions hold.*

- *Let $c = 4\ln\left(q/\varepsilon\right)$. The distribution of $(k', d'_1, \ldots, d'_{q-1})$ is concentrated in the set $[2m] \times [d_1, cd_1] \times \cdots \times [d_{q-1}, cd_{q-1}]$ with probability $1 - \varepsilon$.*
- *For any $\left(\ell_2, \ldots, \ell_{q-1}\right) \in F_2 \times \cdots \times F_{q-1}$, any support of $\left(k', d'_1, \ldots, d'_{q-1}\right)$ has probability at most $\frac{(32/\delta)^q}{m d_1} \prod_{i=2}^{q-1} \frac{1}{h_{\ell_i} d_i}$.*

Proof. For convenience, let $k'_0 = k'$ and $k'_i = k'_0 + \sum_{j=1}^{i} d'_i$. Similar to the proof of Lemma 3, we will write $k' \mapsto k$ for the event "the k'-th bit is not deleted and has index k after the deletion process", and write $\left(k', d'_1, \ldots, d'_{q-1}\right) \mapsto \left(k, d_1, \ldots, d_{q-1}\right)$ for the event $\bigwedge_{i=0}^{q-1} \left(k'_i \mapsto k_i\right)$.

To prove the first item, we are going to condition on an set of deletion probabilities (i.e. $q_{\ell,b}$ for each block and β), and e_2 in Step 3. For each $i \in [q-1]$, we consider a random variable X_i denoting the number of deletions introduced to $I_i \coloneqq \left\{k'_{i-1}+1, \ldots, k'_i - 1\right\}$. It always holds that $0 \le X_i \le d'_i - 1$. Note that X_i does not depend on the deletions introduced in Step 3. Under the error distribution, each of these bits is deleted independently with probability at most $(h+\beta)\delta \le \delta/2$. Thus, following an analysis similar to the proof of Lemma 3, the choice of $c = 4\ln(q/\varepsilon)$ guarantees

$$\Pr[d'_i > cd_i] \le \frac{\varepsilon}{q-1}.$$

Taking a union bound shows that

$$\Pr\left[(k', d'_1, d'_2, \ldots, d'_{q-1}) \in [2m] \times [d_1, cd_1] \times \cdots \times [d_{q-1}, cd_{q-1}]\right] \ge 1 - \varepsilon.$$

Recall that this holds for any set of deletion probabilities, and thus the first item follows.

We now show the second item: for any $\left(\ell_1, \ldots, \ell_{q-1}\right) \in F_1 \times \cdots \times F_{q-1}$, we have

$$\Pr\left[\left(k', d'_1, \ldots, d'_{q-1}\right) \mapsto \left(k, d_1, \ldots, d_{q-1}\right)\right] \le \frac{\left(32/\delta\right)^q}{md_1} \cdot \prod_{i=2}^{q-1} \frac{1}{h_{\ell_i} d_i}.$$

Denote by $\Pr_{S_1,S_2}[\cdot]$ the error distribution before Step 3. Recall that for $i \in [q-1]$, X_i is the number of deletions introduced to the interval I_i, which is independent of Step 3. We first observe that Step 3 does not change the relative distances among the queried indices. Therefore we have

$$\begin{aligned}
&\Pr\left[\left(k', d'_1, \ldots, d'_{q-1}\right) \mapsto \left(k, d_1, \ldots, d_{q-1}\right)\right] \\
=&\frac{1}{\lfloor \delta m/4 \rfloor} \cdot \sum_{e_2=0}^{\lfloor \delta m/4 \rfloor} \Pr\left[\left(k', d'_1, \ldots, d'_{q-1}\right) \mapsto \left(k, d_1, \ldots, d_{q-1}\right) \;\middle|\; e_2\right] \\
=&\frac{1}{\lfloor \delta m/4 \rfloor} \cdot \sum_{e_2=0}^{\lfloor \delta m/4 \rfloor} \Pr_{S_1,S_2}\left[\left(k', d'_1, \ldots, d'_{q-1}\right) \mapsto \left(k + e_2, d_1, \ldots, d_{q-1}\right)\right] \\
\le&\frac{8}{\delta m} \cdot \Pr\left[\left(X_1 = d'_1 - d_1\right) \wedge \cdots \wedge \left(X_{q-1} = d'_{q-1} - d_{q-1}\right)\right].
\end{aligned}$$

In the rest of the proof we will think of the error distribution as comprised of only Step 1 and 2. The chain rule of conditional probability gives

$$\begin{aligned}
&\Pr\left[\left(X_1 = d'_1 - d_1\right) \wedge \cdots \wedge \left(X_{q-1} = d'_{q-1} - d_{q-1}\right)\right] \\
=&\Pr\left[X_1 = d'_1 - d_1\right] \cdot \prod_{i=2}^{q-1} \Pr\left[X_i = d'_i - d_i \mid X_1 = d'_1 - d_1, \ldots, X_{i-1} = d'_{i-1} - d_{i-1}\right].
\end{aligned}$$

We finish the proof with 2 claims.

Claim 6. $\Pr[X_1 = d'_1 - d_1] \le 16/(\delta d'_1)$.

Proof of the claim. We are going to condition on the deletion probabilities $q_{\ell,b}$ and prove the same bound for any $q_{\ell,b} \in [0, h_\ell\delta]$. This clearly implies the claim. Moreover, under this conditional distribution, the deletions of individual bits in Step 1 are mutually independent.

Write $X_1 = X_1^{(1)} + X_1^{(2)}$ where $X_1^{(i)}$ $(i = 1, 2)$ is the number of deletions occurred in I_1, introduced in Step i. Since Step 1 deletes each bit independently with probability at most $h\delta \le \delta/4$, Hoeffding's inequality shows that

$$\Pr\left[X_1^{(1)} \ge \frac{1}{2}d_1'\right] \le \exp\left(-\frac{d_1'}{2}\right) \le \frac{1}{d_1'},$$

where the last inequality holds as long as $d_1' \ge 1$. Also notice that given $X_1^{(1)}$, $X_1^{(2)}$ follows a compound distribution $B\left(d_1' - X_1^{(1)} - 1, \mathbf{U}[0, \delta/4]\right)$. Therefore

$$\begin{aligned}
\Pr\left[X_1 = d_1' - d_1\right] &= \mathbb{E}_{X_1^{(1)}}\left[\Pr\left[X_1^{(2)} = d_1' - d_1 - X_1^{(1)}\right] \;\middle|\; X_1^{(1)}\right] \\
&\le \mathbb{E}_{X_1^{(1)}}\left[\frac{4}{\delta} \cdot \frac{1}{d_1' - X_1^{(1)}}\right] \\
&\le \frac{4}{\delta} \cdot \frac{1}{d_1'/2} + \frac{4}{\delta} \cdot \Pr\left[X_1^{(1)} \ge \frac{1}{2}d_1'\right] \\
&\le \frac{16}{\delta} \cdot \frac{1}{d_1'}.
\end{aligned}$$

Here the first equality uses Lemma 2. □

Claim 7. $\forall 2 \le i \le q-1$, $\Pr\left[X_i = d_i' - d_i \mid X_1 = d_1' - d_1, \ldots, X_{i-1} = d_{i-1}' - d_{i-1}\right] \le 32/(\delta h_{\ell_i} d_i)$.

Proof of the claim. For the i-th term where $2 \le i \le q-1$, we recall that $\ell_i \in F_i$. Since all blocks in layer ℓ_i have size $s_{\ell_i} \le d_i/2 \le d_i'/2$ by the definition of F_i, there exists a block in layer ℓ_i which is completely contained in I_i. Suppose it is the b-th block and denote it by B_i. Note that we may also assume $|B_i| \ge d_i/4$ (if B_i is the last block and $|B_i| < d_i/4$, then the second last block is also contained in I_i and has size $s_{\ell_i} \ge d_i/4$).

Similar to the proof of the previous claim, we are going to condition on β and the deletion probabilities $q_{\ell,b'}$ for all $\ell \in [L]$ and $b' \le \lceil 2m/s_\ell \rceil$, except for $q_{\ell_i,b}$ which is the deletion probability of B_i. Proving the same bound under the conditional distribution will imply the claim.

Write $X_i = X_{i,B} + X_{i,B}' + X_{i,\varnothing}$ where $X_{i,B}$ is the number of deletions introduced to B_i by layer ℓ_i, $X_{i,B}'$ is the number of deletions introduced to B_i by other sources, and $X_{i,\varnothing}$ is the number of deletions introduced to $I_i \setminus B_i$.

A crucial observation is that given $X_{i,B}'$, $X_{i,B}$ is independent of the X_j's for $j \ne i$, and follows a compound distribution $B\left(|B_i| - X_{i,B}', \mathbf{U}[0, h_{\ell_i}\delta]\right)$. Similar to the analysis for $X_1^{(1)}$, since each bit is deleted independently with probability at most $(h+\beta)\delta \le \delta/2$ during Step 1 and 2, Hoeffding's inequality implies

$$\Pr\left[X_{i,B}' \ge \frac{3}{4}|B_i|\right] \le \exp\left(-\frac{|B_i|}{8}\right) \le \frac{4}{|B_i|},$$

where the last inequality holds as long as $|B_i| \geq 1$. Therefore we have

$$\begin{aligned}
&\Pr\left[X_i = d'_i - d_i \mid X_1 = d'_1 - d_1, \ldots, X_{i-1} = d'_{i-1} - d_{i-1}\right] \\
=&\mathbb{E}_{X'_{i,B}, X_{i,\varnothing}}\left[\Pr\left[X_i = d'_i - d_i \mid X_1 = d'_1 - d_1, \ldots, X_{i-1} = d'_{i-1} - d_{i-1}\right] \;\middle|\; X'_{i,B}, X_{i,\varnothing}\right] \\
=&\mathbb{E}_{X'_{i,B}, X_{i,\varnothing}}\left[\Pr\left[X_{i,B} = d'_i - d_i - X'_{i,B} - X_{i,\varnothing}\right] \;\middle|\; X'_{i,B}, X_{i,\varnothing}\right] \\
\leq&\mathbb{E}_{X'_{i,B}, X_{i,\varnothing}}\left[\frac{1}{h_{\ell_i}\delta} \cdot \frac{1}{|B_i| - X'_{i,B} + 1}\right] \\
\leq&\frac{1}{h_{\ell_i}\delta} \cdot \left(\frac{1}{|B_i|/4} + \frac{4}{|B_i|}\right) = \frac{8}{h_{\ell_i}\delta} \cdot \frac{1}{|B_i|} \leq \frac{32}{\delta} \cdot \frac{1}{h_{\ell_i} d_i}.
\end{aligned}$$

Here the first inequality is again due to Lemma 2. □

Putting everything together, we have shown that

$$\begin{aligned}
\Pr\left[\left(k', d'_1, \ldots, d'_{q-1}\right) \mapsto \left(k, d_1, \ldots, d_{q-1}\right)\right] &\leq \frac{8}{\delta m} \cdot \left(\frac{16}{\delta} \cdot \frac{1}{d'_1}\right) \cdot \prod_{i=2}^{q-1} \left(\frac{32}{\delta} \cdot \frac{1}{h_{\ell_i} d_i}\right) \\
&\leq \frac{(32/\delta)^q}{m d_1} \cdot \prod_{i=2}^{q-1} \frac{1}{h_{\ell_i} d_i}.
\end{aligned}$$

□

In the rest of the section, we instantiate $\mathcal{D}(L, \mathbf{s}, \mathbf{h})$ with two specific sets of parameters, which we now describe.

4.1 An error distribution independent of the code

We now define an error distribution $\mathcal{D}_{obl}$ which is completely independent of the coding scheme $(C\colon \{0,1\}^n \to \Sigma^m, \mathsf{Dec})$, message x and codeword $C(x)$. As such lower bounds obtained from $\mathcal{D}_{obl}$ will also apply in the private-key setting where the encoder and decoder share secret random coins.

We take $L_0 = \lceil \log(2m) \rceil \leq \log m + 2$, $\mathbf{s}_0 = (s_1, \ldots, s_{L_0})$ and $\mathbf{h}_0 = (h_1, \ldots, h_{L_0})$ where

$$\forall \ell \in [L_0], \quad s_\ell = 2^\ell, \quad h_\ell = \frac{1}{4L_0}.$$

Let $\mathcal{D}_{obl} = \mathcal{D}(L_0, \mathbf{s}_0, \mathbf{h}_0)$. Note that $\mathcal{D}_{obl}$ is oblivious to the encoding/decoding scheme.

Clearly $h = 1/4$ for $\mathcal{D}_{obl}$. Consider an arbitrary query $(k, d_1, \ldots, d_{q-1})$. For each $i \in [q-1]$, we let $\ell_i = \lceil \log_2 d_i \rceil - 2$. Since $\log_2 d_i - 2 \leq \lceil \log_2 d_i \rceil - 2 \leq \log_2 d_i - 1$, we have

$$s_{\ell_i} = 2^{\ell_i} \geq 2^{\log_2 d_i - 2} \geq \frac{d_i}{4}, \text{ and } s_{\ell_i} \leq 2^{\log_2 d_i - 1} \leq \frac{d_i}{2},$$

which means $\ell_i \in F_i$. The corresponding $h_{\ell_i} = 1/(4L_0) \geq 1/(4(\log m + 2))$. Therefore we obtain the following corollary to Lemma 4.

Corollary 4. *Let* $\left(k', d'_1, \ldots, d'_{q-1}\right)$ *be the random tuple which corresponds to the query* $\left(k, d_1, \ldots, d_{q-1}\right)$ *under error distribution* $\mathcal{D}_{obl}$*. Then any support of* $\left(k', d'_1, \ldots, d'_{q-1}\right)$ *has probability at most*

$$\frac{(32/\delta)^q}{md_1} \cdot \prod_{i=2}^{q-1} \frac{4(\log m + 2)}{d_i}.$$

4.2 An adversarial error distribution for $q \geq 3$

We now define a non-oblivious error distribution $\mathcal{D}_{adv,i}$ which may depend on the decoder Dec. Analyzing $\mathcal{D}_{adv,i}$ allows us to derive tighter lower bounds on the codeword length m for a Insdel LDC with query complexity q. However, because the distribution is not oblivious the stronger lower bounds derived from $\mathcal{D}_{adv,i}$ no longer apply in the private-key setting.

Fix $i \in [n]$. Let $\left(K, D_1, D_2, \ldots, D_{q-1}\right)$ be the random variable that corresponds to queries of $\mathsf{Dec}\,(\cdot, m, i)$. For $1 \leq \tau \leq \lceil \log(2m) \rceil$, let $p_{\tau,i}$ be the probability that $2^{\tau-1} \leq D_2 < 2^{\tau}$. Thus, $p_{\tau,i}$ is the probability that the decoder for the i-th bit ($\mathsf{Dec}(\cdot, m, i)$) queries a tuple $(k, d_1, \ldots, d_{q-1})$ such that $2^{\tau-1} \leq d_2 < 2^{\tau}$. We have $\sum_{\tau=1}^{\lceil \log(2m) \rceil} p_{\tau,i} = 1$.

We take $L = 2L_0$ where $L_0 = \lceil \log(2m) \rceil$. The vectors $\mathbf{s} = (s_1, \ldots, s_L)$ and $\mathbf{h} = (h_1, \ldots, h_L)$ are defined as follows.

- $\forall\ 1 \leq \ell \leq L_0$, $s_\ell = 2^\ell$, and $h_\ell = 1/(8L_0)$.
- $\forall\ 1 \leq \tau \leq L_0$, $s_{L_0+\tau} = 2^{\tau-2}$, and $h_{d+L_0} = p_{\tau,i}/8$.

We define the adversary error distribution depending on $\mathsf{Dec}(\cdot, m, i)$ as $\mathcal{D}_{adv,i} \coloneqq \mathcal{D}(L, \mathbf{s}, \mathbf{h})$. For this error distribution we also have

$$h = \sum_{\ell=1}^{L_0} h_\ell + \sum_{\tau=1}^{L_0} h_{\tau+L_0} = L_0 \cdot \frac{1}{8L_0} + \frac{1}{8} \cdot \sum_{\tau=1}^{L_0} p_{\tau,i} = \frac{1}{4}.$$

Let $\left(k, d_1, d_2, \ldots, d_{q-1}\right)$ be an arbitrary query in the support of $\mathsf{Dec}(\cdot, m, i)$ and $1 \leq \tau_0 \leq t$ be the integer such that $2^{\tau_0-1} \leq d_2 < 2^{\tau_0}$. We set $\ell_2 = L_0 + \tau_0$. Since $s_{\ell_2} = 2^{\tau_0-2}$, we have $d_2/4 \leq s_{\ell_2} \leq d_2/2$. Thus, $\ell_2 \in F_2$ with $h_{\ell_2} = p_{\tau_0,i}/8$.

For $3 \leq j \leq q-1$, we set $\ell_j = \lceil \log d_j \rceil - 2 \in F_j$.Thus, $h_{\ell_j} = 1/(8L_0) \geq 1/(8(\log m + 2))$ for $3 \leq j \leq q-1$. We have the following corollary to Lemma 4.

Corollary 5. *Let* $\left(k', d'_1, \ldots, d'_{q-1}\right)$ *be the random tuple that corresponds to the query* $\left(k, d_1, \ldots, d_{q-1}\right)$ *under error distribution* $\mathcal{D}_{adv,i}$*. Let* τ_0 *be the integer such that* $2^{\tau_0-1} \leq d_2 < 2^{\tau_0}$*. Then any support of* $\left(k', d'_1, \ldots, d'_{q-1}\right)$ *has probability at most*

$$\frac{(32/\delta)^q}{m} \cdot \frac{8^{q-2}(\log m + 2)^{q-3}}{p_{\tau_0,i}} \cdot \prod_{\ell=1}^{q-1} \frac{1}{d_\ell}.$$

5 Lower Bounds For Private-key Insdel LDCs

We will prove the second part of Theorem 3 in this section, since the proof is simpler. The error distribution is going to be $\mathcal{D}_{obl}$ defined in Section 4.1. Because $\mathcal{D}_{obl}$ is independent of the coding scheme, message and codeword the lower bounds apply in the private-key setting. Of course the lower bounds still apply for general LDCs. However, we can derive tighter lower bounds for general LDCs using a different error distribution which may depend on the local decoder Dec — see Section 6.

Let $c = 4\ln(q/\varepsilon) \geq 2$ be the constant from Lemma 4. For $j_1, j_2, \ldots, j_{q-1} \in [t]$ where $t = \lceil \log_c(2m) \rceil \leq \log m + 2$, denote

$$P_{j_1,\ldots,j_{q-1}} = [2m] \times [c^{j_1-1}, c^{j_1}) \times \cdots \times [c^{j_{q-1}-1}, c^{j_{q-1}}).$$

Claim 8. *Let $\gamma = \varepsilon/\left(256c^2/\delta\right)^q$. For any $i \in [n]$, there exist $j_1, \cdots, j_{q-1} \in [t]$ such that*

$$\left|P_{j_1,\ldots,j_{q-1}} \cap \mathsf{Good}_i\right| \geq \frac{\gamma\left|P_{j_1,\ldots,j_{q-1}}\right|}{(\log m + 2)^{q-2}}.$$

Proof. Fix any $i \in [n]$. Let $D \subseteq [2m]$ be a random set of deletions generated by $\mathcal{D}_{obl}$. Let $\mathcal{E}$ be the event that $\left|D \cap [m]\right| \leq \delta m$ and $|D| \leq m$. By Proposition 5 and a union bound, $\mathcal{E}$ happens with probability at least $1 - \exp\left(-\delta^2 m/8\right) - \exp\left(-(1-\delta)^2 m\right) \geq 1 - \varepsilon/4$ for large enough n (and thus large enough m). Therefore by Lemma 1, we have

$$\begin{aligned}
\Pr\left[\mathsf{Dec}(\cdot, m, i) \text{ hits } \mathsf{Good}_i\right] &\geq \Pr\left[\mathsf{Dec}(\cdot, m, i) \text{ hits } \mathsf{Good}_i \mid \mathcal{E}\right] \cdot \Pr\left[\mathcal{E}\right] \\
&\geq \frac{3\varepsilon}{2} \cdot \left(1 - \frac{\varepsilon}{4}\right) \\
&\geq \frac{5\varepsilon}{4}.
\end{aligned}$$

Here in the case of $|D| > m$ we simply assume that $\mathsf{Dec}(\cdot, m, i)$ never hits Good_i. By the first item of Lemma 4 and a union bound, for at least one query $\left(k, d_1, \ldots, d_{q-1}\right)$ we have

$$\Pr\left[\left(k', d'_1, \ldots, d'_{q-1}\right) \in \left([2m] \times [d_1, cd_1) \times \cdots \times [d_{q-1}, cd_{q-1})\right) \cap \mathsf{Good}_i\right] \geq \frac{\varepsilon}{4},$$

where $\left(k', d'_1, \ldots, d'_{q-1}\right)$ corresponds to $\left(k, d_1, \ldots, d_{q-1}\right)$ under $\mathcal{D}_{obl}$. By Corollary 4, we have that

$$\left|\left([2m] \times [d_1, cd_1) \times \cdots \times [d_{q-1}, cd_{q-1})\right) \cap \mathsf{Good}_i\right| \geq \frac{\varepsilon m d_1 \cdots d_{q-1}}{4\left(32/\delta\right)^q \cdot \left(4\left(\log m + 2\right)\right)^{q-2}}.$$

Take $j_1, \cdots, j_{q-1} \in [t]$ such that $c^{j_\ell - 1} \leq d_\ell < c^{j_\ell}$ for all $1 \leq \ell \leq q-1$. Note that for each $\ell \leq q-1$, $[d_\ell, cd_\ell] \subseteq [c^{j_\ell - 1}, c^{j_\ell + 1}) = [c^{j_\ell - 1}, c^{j_\ell}) \cup [c^{j_\ell}, c^{j_\ell + 1})$. This implies

$$[2m] \times [d_1, cd_1] \times \cdots \times [d_{q-1}, cd_{q-1}] \subseteq \bigcup_{\forall 1 \leq \ell \leq q-1, j'_\ell \in \{j_\ell, j_\ell + 1\}} P_{j'_1,\ldots,j'_{q-1}}.$$

Therefore for some $j'_1, \ldots, j'_{q-1} \in [t]$ we have

$$\begin{aligned}
\left|P_{j'_1,\ldots,j'_{q-1}} \cap \mathsf{Good}_i\right| &\geq \frac{1}{2^{q-1}} \cdot \left|\left([2m] \times [d_1, cd_1] \times \cdots \times [d_{q-1}, cd_{q-1}]\right) \cap \mathsf{Good}_i\right| \\
&\geq \frac{1}{2^{q-1}} \cdot \frac{\varepsilon m d_1 \cdots d_{q-1}}{4\left(32/\delta\right)^q \cdot \left(4\left(\log m + 2\right)\right)^{q-2}} \\
&\geq \frac{\varepsilon}{(256c^2/\delta)^q} \cdot \frac{\left|P_{j'_1,\ldots,j'_{q-1}}\right|}{(\log m + 2)^{q-2}} \\
&= \frac{\gamma\left|P_{j'_1,\ldots,j'_{q-1}}\right|}{(\log m + 2)^{q-2}}.
\end{aligned}$$

Here the last inequality is because $\left|P_{j'_1,\ldots,j'_{q-1}}\right| \leq c^{2q} \cdot 2md_1d_2 \cdots d_{q-1}$. □

For each $i \in [n]$, we fix a tuple $\mathbf{J}_i = \left(j_1, \ldots, j_{q-1}\right) \in [t]^{q-1}$ such that

$$\left|P_{\mathbf{J}_i} \cap \mathsf{Good}_i\right| \geq \frac{\gamma\left|P_{\mathbf{J}_i}\right|}{(\log m + 2)^{q-2}}.$$

Such a $\mathbf{J}_i$ exists as guaranteed by Claim 8. Given a tuple $\mathbf{J} = \left(j_1, \ldots, j_{q-1}\right) \in [t]^{q-1}$, define

$$G_{\mathbf{J}} = \left\{i \in [n] \colon \mathbf{J} = \mathbf{J}_i\right\}.$$

Claim 9. $\forall \mathbf{J} \in [t]^{q-1}, |G_{\mathbf{J}}| \leq q\left(\log m + 2\right)^{q-2} / \left(\gamma\left(1 - \mathcal{H}(1/2 + \varepsilon/4)\right)\right)$.

Proof. By counting the number of pairs $(Q, i) \in P_{\mathbf{J}} \times G_{\mathbf{J}}$ such that $Q \in \mathsf{Good}_i$ in two ways, we have

$$\sum_{Q \in P_{\mathbf{J}}} \left|H_Q \cap G_{\mathbf{J}}\right| = \sum_{i \in G_{\mathbf{J}}} \left|P_{\mathbf{J}} \cap \mathsf{Good}_i\right|.$$

On the one hand, by Proposition 3 we have

$$\sum_{Q \in P_{\mathbf{J}}} \left|H_Q \cap G_{\mathbf{J}}\right| \leq \sum_{Q \in P_{\mathbf{J}}} \left|H_Q\right| \leq \frac{q|P_{\mathbf{J}}|}{1 - \mathcal{H}(1/2 + \varepsilon/4)}.$$

On the other hand, by definition of $G_{\mathbf{J}}$ we have

$$\sum_{i \in G_{\mathbf{J}}} \left|P_{\mathbf{J}} \cap \mathsf{Good}_i\right| \geq |G_{\mathbf{J}}| \cdot \frac{\gamma|P_{\mathbf{J}}|}{(\log m + 2)^{q-2}}.$$

Rearranging gives the claim. □

Now we are ready to prove the second part of Theorem 3. Of course the lower bound also applies in settings where the encoding/decoding scheme do not share secret random coins, but in these settings we can establish an even stronger bound by modifying $\mathcal{D}_{obl}$ to depend on the specific encoding/decoding scheme.

Theorem 3. *For any non-adaptive (q, δ, ε) insdel LDC $C\colon \{0,1\}^n \to \{0,1\}^m$ with $q \geq 3$, we have the following bounds.*

- *For arbitrary adversarial channels,*

$$m = \begin{cases} \exp(\Omega_{\delta,\varepsilon}(\sqrt{n})) \text{ for } q = 3\text{; and} \\ \exp\left(\Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)} \cdot \left(\varepsilon^3 n\right)^{1/(2q-4)}\right)\right) \text{ for } q \geq 4. \end{cases}$$

- *For the private-key setting,*

$$m = \exp\left(\Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)} \cdot \left(\varepsilon^3 n\right)^{1/(2q-3)}\right)\right).$$

Proof of the second part. Note that $\cup_{\mathbf{J}\in[t]^{q-1}} G_{\mathbf{J}} = [n]$. Therefore by Claim 9 and substituting $\gamma = \left(256c^2/\delta\right)^q$ we have

$$n \leq \sum_{\mathbf{J}\in[t]^{q-1}} |G_{\mathbf{J}}| \leq t^{q-1} \cdot \frac{q\,(\log m + 2)^{q-2}}{\gamma\left(1 - \mathcal{H}(1/2 + \varepsilon/4)\right)} \leq \frac{24}{(\ln 2)^2} \cdot \frac{1}{\varepsilon^3} \cdot \left(\frac{512c^2}{\delta}\right)^q \cdot (\log m + 2)^{2q-3}$$

where in the last inequality we used Proposition 1, $q \leq 2q - 3$ for $q \geq 3$ and $q \leq 2^q$ for $q \geq 1$. Substituting $c = 4\ln(q/\varepsilon)$ and taking C to be a large enough constant, we can write

$$n \leq \frac{1}{\varepsilon^3} \cdot \left(\frac{C}{\delta} \cdot \ln^2\left(\frac{q}{\varepsilon}\right) \cdot \log m\right)^{2q-3}.$$

Solving for m gives

$$m \geq \exp\left(\Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)} \cdot \left(\varepsilon^3 n\right)^{1/(2q-3)}\right)\right).$$

Finally, we observe that $\mathcal{D}_{obl}$ is oblivious to the encoding/decoding scheme and the specific codeword. Thus, the lower bound still applies even if the encoder/decoder share secret random coins. □

6 Stronger Lower Bounds For Insdel LDCs

In this section, we prove the first part of Theorem 3. We assume the error distribution is $\mathcal{D}_{adv,i}$ introduced in section 4.2. Following the notation from section 4.2 and section 5, let $p_{\tau,i}$ be the probability that $\mathsf{Dec}(\cdot, m, i)$ queries a tuple $(k, d_1, \ldots, d_{q-1})$ such that $2^{\tau-1} \leq d_2 < 2^\tau$. We have $\sum_{\tau=1}^{\lceil \log(2m) \rceil} p_{\tau,i} = 1$. Take $\eta = (256/\delta)^q$, $c = 4\ln(q/\varepsilon) \geq 2$ and denote $t = \lceil \log_c(2m) \rceil$. For $j_1, j_2, \ldots, j_{q-1} \in [t]$, denote

$$P_{j_1,\ldots,j_{q-1}} = [2m] \times [c^{j_1-1}, c^{j_1}) \times \cdots \times [c^{j_{q-1}-1}, c^{j_{q-1}}).$$

Let $I_\tau = \{P_{j_1,\ldots,j_{q-1}} : 2^{\tau-1} \le c^{j_2} \le c^2 2^\tau\}$ be a set of subcubes. We define

$$\beta_{\tau,i} = \max_{P_{\mathbf{J}} \in I_\tau} \frac{|P_{\mathbf{J}} \cap \mathsf{Good}_i|}{|P_{\mathbf{J}}|}.$$

Thus, $\beta_{\tau,i}$ is the maximum fraction of good points in any subcube $P_{\mathbf{J}}$ in the set I_τ.

Claim 10. *For any $i \in [n]$, we have*

$$\sum_{\tau=1}^{t} \beta_{\tau,i} \ge \frac{\varepsilon}{8\eta(2c^2)^{q-1}(\log m + 2)^{q-3}}.$$

Proof. We fix $i \in [n]$. Let $Q = (k, d_1, \ldots, d_{q-1})$ be an arbitrary query in the support of $\mathsf{Dec}(\cdot, m, i)$, and let $Q' = (k', d'_1, \ldots, d'_{q-1})$ be the random tuple corresponding to $(k, d_1, \ldots, d_{q-1})$ under error distribution $\mathcal{D}_{adv,i}$.

For $1 \le \ell \le q-1$, we let j'_ℓ be the integer such that $c^{j'_\ell - 1} \le d_\ell < c^{j'_\ell}$. We have $[d_\ell, cd_\ell] \subseteq [c^{j'_\ell - 1}, c^{j'_\ell}) \cup [c^{j'_\ell}, c^{j'_\ell + 1})$. Let U_Q be a set of 2^{q-1} tuples $(j_1, \ldots, j_{q-1})$ such that $j_\ell \in \{j'_\ell, j'_\ell + 1\}$ for all $\ell \in [q-1]$ (if $j'_\ell = t$, fix $j_\ell = j'_\ell$). By Lemma 4, with probability at least $1 - \varepsilon$, we have

$$(k', d'_1, \ldots, d'_{q-1}) \in \bigcup_{\mathbf{J} \in U} P_{\mathbf{J}}.$$

Denote this event by $\mathcal{E}$. We now give an upper bound of the probability that Q' hits Good_i in terms of the $\beta_{\tau,i}$'s. Let $1 \le \tau_Q \le \lceil \log(2m) \rceil$ be the integer such that $2^{\tau_Q - 1} \le d_2 < 2^{\tau_Q}$. Notice that $2^{\tau_Q - 1} \le d_2 < c^{j'_2} < c^{j'_2 + 1}$ and $c^{j'_2 + 1} \le c^2 d_2 < c^2 2^{\tau_Q}$. We have $2^{\tau_Q - 1} \le c^{j'_2} < c^{j'_1 + 1} \le c^2 2^{\tau_Q}$. Thus for any $\mathbf{J} \in U_Q$, we have $P_{\mathbf{J}} \in I_{\tau_Q}$. By our definition of $\beta_{\tau_Q,i}$, we have

$$\beta_{\tau_Q,i} \ge \frac{|P_{\mathbf{J}} \cap \mathsf{Good}_i|}{|P_{\mathbf{J}}|}.$$

By Corollary 5, any support of $(k', d'_1, \cdots, d'_{q-1})$ has probability at most

$$\frac{\eta(\log m + 2)^{q-3}}{m d_1 \ldots d_{q-1} p_{\tau_Q,i}}$$

for $\eta = \big(256/\delta\big)^q$.

The size of any subcube $P_{j_1,\ldots,j_{q-1}}$ is bounded by $2mc^{j_1} \cdots c^{j_{q-1}}$. Since for $\mathbf{J} \in U_Q$ we have $c^{j_\ell} \le c^{j'_\ell + 1} \le c^2 d_\ell$ for any $1 \le \ell \le q-1$, we have $|P_{\mathbf{J}}| \le (c^2)^{q-1} \cdot 2m d_1 \cdots d_{q-1}$. Thus the probability that $(k', d'_1, \ldots, d'_{q-1})$ hits Good_i can be bounded by

$$\begin{aligned}
&\Pr\left[(k', d'_1, \ldots, d'_{q-1}) \text{ hits } \mathsf{Good}_i\right] \\
\le &\Pr\left[\overline{\mathcal{E}}\right] + \Pr\left[(k', d'_1, \ldots, d'_{q-1}) \in \mathsf{Good}_i \cap \bigcup_{\mathbf{J} \in U_Q} P_{\mathbf{J}}\right] \\
\le &\varepsilon + \frac{\eta(\log m + 2)^{q-3}}{m d_1 \cdots d_{q-1} p_{\tau_Q,i}} \cdot \beta_{\tau_Q,i} \cdot \sum_{\mathbf{J} \in U_Q} |P_{\mathbf{J}}| \\
\le &\varepsilon + 2\beta_{\tau_Q,i} \cdot \frac{(2c^2)^{q-1}\eta(\log m + 2)^{q-3}}{p_{\tau_Q,i}}.
\end{aligned}$$

Denote by $\mu_i(\cdot)$ the probability distribution of the queries of $\mathsf{Dec}(\cdot, m, i)$. For the probability that $\mathsf{Dec}(\cdot, m, i)$ hits Good_i, we have

$$\begin{aligned}
&\Pr\left[\mathsf{Dec}(\cdot, m, i) \text{ hits } \mathsf{Good}_i\right] \\
=&\sum_{Q\in\binom{[2m]}{q}} \mu_i(Q)\cdot \Pr\left[\mathsf{Dec}(\cdot, m, i) \text{ hits } \mathsf{Good}_i \mid \mathsf{Dec}(\cdot, m, i) \text{ queries } Q\right] \\
\le&\sum_{Q\in\binom{[2m]}{q}} \mu_i(Q)\cdot\left(\varepsilon + 2\beta_{\tau_Q,i}\cdot\frac{(2c^2)^{q-1}\eta(\log m+2)^{q-3}}{p_{\tau_Q,i}}\right) \\
=&\sum_{Q\in\binom{[2m]}{q}} \mu_i(Q)\varepsilon + \sum_{Q\in\binom{[2m]}{q}} \mu_i(Q)\cdot 2\beta_{\tau_Q,i}\cdot\frac{(2c^2)^{q-1}\eta(\log m+2)^{q-3}}{p_{\tau_Q,i}} \\
=&\varepsilon + (2c^2)^{q-1}\eta(\log m+2)^{q-3}\cdot\sum_{\tau=1}^{\lceil\log(2m)\rceil} 2\beta_{\tau,i}\cdot\frac{1}{p_{\tau,i}}\cdot\sum_{Q:\ \tau_Q=\tau}\mu_i(Q) \\
\le&\varepsilon + 2(2c^2)^{q-1}\eta(\log m+2)^{q-3}\sum_{\tau=1}^{\lceil\log(2m)\rceil}\beta_{\tau,i}.
\end{aligned}$$

The last equality is due to the fact that for any $1\le\tau\le\lceil\log(2m)\rceil$,

$$p_{\tau,i} = \sum_{Q:\ \tau_Q=\tau}\mu_i(Q).$$

Similar to the argument used in the proof of Claim 8, by Proposition 5 and Lemma 1, for large enough n (and thus m) the probability that $\mathsf{Dec}(\cdot, m, i)$ hits Good_i is at least $3\varepsilon/2 - \varepsilon/4 = 5\varepsilon/4$. Thus

$$2(2c^2)^{q-1}\eta(\log m+2)^{q-3}\sum_{\tau=1}^{\lceil\log(2m)\rceil}\beta_{\tau,i}\ge\varepsilon/4.$$

□

Claim 11. $\sum_{i=1}^{n}\sum_{\tau=1}^{\lceil\log(2m)\rceil}\beta_{\tau,i}\le 3q\log c\cdot t^{q-1}/\left(1-\mathcal{H}(1/2+\varepsilon/4)\right)$.

Proof. By Proposition 3, for any tuple $Q\in\binom{[m]}{k}$, Q is in Good_i for at most $\frac{q}{1-\mathcal{H}(1/2+\varepsilon/4)}$ different i's. Thus for any subcube $P_{\mathbf{J}}$, we have

$$\sum_{i=1}^{n}|P_{\mathbf{J}}\cap\mathsf{Good}_i|\le\frac{q}{1-\mathcal{H}(1/2+\varepsilon/4)}|P_{\mathbf{J}}|.$$

Meanwhile, by the definition of $\beta_{\tau,i}$, we have

$$\beta_{\tau,i} = \max_{P_{\mathbf{J}}\in I_\tau}\frac{|P_{\mathbf{J}}\cap\mathsf{Good}_i|}{|P_{\mathbf{J}}|}\le\sum_{P_{\mathbf{J}}\in I_\tau}\frac{|P_{\mathbf{J}}\cap\mathsf{Good}_i|}{|P_{\mathbf{J}}|}.$$

Combining the above two inequalities, we have

$$\sum_{i=1}^{n}\beta_{\tau,i} \le \sum_{i=1}^{n}\sum_{P_{\mathbf{J}}\in I_\tau}\frac{|P_{\mathbf{J}}\cap \mathsf{Good}_i|}{|P_{\mathbf{J}}|} \le \frac{q}{1-\mathcal{H}(1/2+\varepsilon/4)}|I_\tau|\,.$$

By the definition of I_τ, each subcube $P_{\mathbf{J}}$ belongs to at most $\lceil \log 2c^2\rceil \le 3\log c$ consecutive I_τ's. Notice that the total number of subcubes is bounded by t^{q-1}. By counting the number of subcubes, we have

$$\sum_{\tau=1}^{\lceil\log(2m)\rceil}|I_\tau| \le 3\log c\cdot t^{q-1}.$$

Thus,

$$\sum_{i=1}^{n}\sum_{\tau=1}^{\lceil\log(2m)\rceil}\beta_{\tau,i} \le \frac{q}{1-\mathcal{H}(1/2+\varepsilon/4)}\cdot\sum_{\tau=1}^{t}|I_\tau| \le \frac{q\cdot 3\log c\cdot t^{q-1}}{1-\mathcal{H}(1/2+\varepsilon/4)}.$$

□

Now we are ready to prove the first part of Theorem 3.

Theorem 3. *For any non-adaptive (q,δ,ε) insdel LDC $C\colon \{0,1\}^n \to \{0,1\}^m$ with $q\ge 3$, we have the following bounds.*

- *For arbitrary adversarial channels,*

$$m = \begin{cases} \exp(\Omega_{\delta,\varepsilon}(\sqrt{n})) \text{ for } q=3\text{; and} \\ \exp\left(\Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)}\cdot\left(\varepsilon^3 n\right)^{1/(2q-4)}\right)\right) \text{ for } q\ge 4. \end{cases}$$

- *For the private-key setting,*

$$m = \exp\left(\Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)}\cdot\left(\varepsilon^3 n\right)^{1/(2q-3)}\right)\right).$$

Proof of the first part. By Claim 10, we have

$$\sum_{i=1}^{n}\sum_{\tau=1}^{\lceil\log(2m)\rceil}\beta_{\tau,i} \ge \frac{n\varepsilon}{8\eta(2c^2)^{q-1}(\log m+2)^{q-3}}.$$

Combined with Claim 11, we have

$$\frac{n\varepsilon}{8\eta(2c^2)^{q-1}(\log m+2)^{q-3}} \le \frac{q\cdot 3\log c\cdot t^{q-1}}{1-\mathcal{H}(1/2+\varepsilon/4)}.$$

Plugging in $\eta=(256/\delta)^q$ and $c=4\ln(q/\varepsilon)$, and noticing that $t\le\log m+2$, $q\le 2^q$, $3\log c\le c^2$, for some large enough constant C we have

$$n \le \frac{1}{\varepsilon(1-\mathcal{H}(1/2+\varepsilon/4))}\cdot\left(\frac{C\ln^2(q/\varepsilon)}{\delta}\right)^q(\log m+2)^{2q-4}.$$

By Proposition 1, we have $1 - \mathcal{H}(1/2 + \varepsilon/4) = \Omega(\varepsilon^2)$. We can rewrite the above inequality as

$$m = \exp\left(\Omega\left(\left(\frac{\delta}{\ln^2(q/\varepsilon)}\right)^{\frac{q}{2q-4}} \cdot \left(\varepsilon^3 n\right)^{\frac{1}{2q-4}}\right)\right).$$

Thus, for $q = 3$, we have $m = \exp(\Omega_{\delta,\varepsilon}(\sqrt{n}))$. For $q \geq 4$, we have $\frac{q}{2q-4} \leq 1$ and $\left(\frac{\delta}{\ln^2(q/\varepsilon)}\right)^{\frac{q}{2q-4}} = \Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)}\right)$. We can write

$$m = \exp\left(\Omega\left(\frac{\delta}{\ln^2(q/\varepsilon)} \cdot \left(\varepsilon^3 n\right)^{\frac{1}{2q-4}}\right)\right).$$

□

References

[ALRW17] 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.

[BB21] Alexander R Block and Jeremiah Blocki. Private and resource-bounded locally decodable codes for insertions and deletions. In *IEEE International Symposium on Information Theory, ISIT*, page (to appear), 2021.

[BBG+20] 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.

[BCG20] 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.

[BDSS16] 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.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC*, pages 21–31, 1991.

[BG17] 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.

[BGGZ19] Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. In *ISIT*, pages 2414–2418. IEEE, 2019.

[BGH+06] 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).

[BGT16] Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query lccs over large alphabet. *arXiv preprint arXiv:1611.06980*, 2016.

[BGZ18] Joshua Brakensiek, Venkatesan Guruswami, and Samuel Zbarsky. Efficient low-redundancy codes for correcting multiple deletions. *IEEE Trans. Inf. Theory*, 64(5):3403–3410, 2018.

[BK95] Manuel Blum and Sampath Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, 1995.

[BKZ20] Jeremiah Blocki, Shubhang Kulkarni, and Samson Zhou. On Locally Decodable Codes in Resource Bounded Channels. 163:16:1–16:23, 2020.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993.

[BRdW08] 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.

[BSB+21] 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.

[CGdW13] Victor Chen, Elena Grigorescu, and Ronald de Wolf. Error-correcting data structures. *SIAM J. Comput.*, 42(1):84–111, 2013.

[CGHL21] 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.

[CHL+19] 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.

[CJLW18] 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.

[CJLW19] 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.

[CKGS98] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[CL21] Kuan Cheng and Xin Li. Efficient document exchange and error correcting codes with asymmetric information. In *SODA*, pages 2424–2443. SIAM, 2021.

[CLZ20] Kuan Cheng, Xin Li, and Yu Zheng. Locally decodable codes with randomized encoding. *CoRR*, abs/2001.03692, 2020.

[DGL04] Yan Ding, Parikshit Gopalan, and Richard Lipton. Error correction against computationally bounded adversaries. Manuscript, 2004.

[DGY11] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM J. Comput.*, 40(4):1154–1178, 2011.

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007.

[DSW17] 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.

[Dud78] Richard M Dudley. Central limit theorems for empirical measures. *The Annals of Probability*, pages 899–929, 1978.

[Dvi10] Zeev Dvir. On matrix rigidity and locally self-correctable codes. In *Computational Complexity Conference*, pages 291–298. IEEE Computer Society, 2010.

[Efr12] Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM J. Comput.*, 41(6):1694–1703, 2012.

[Gas04] William I. Gasarch. A survey on private information retrieval (column: Computational complexity). *Bulletin of the EATCS*, 82:72–107, 2004.

[GHS20] Venkatesan Guruswami, Bernhard Haeupler, and Amirbehshad Shahrasbi. Optimally resilient codes for list-decoding from insertions and deletions. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *STOC*, pages 524–537. ACM, 2020.

[GKST06] 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.

[GL18] Venkatesan Guruswami and Ray Li. Coding against deletions in oblivious and online models. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 625–643. SIAM, 2018.

[GL19] Venkatesan Guruswami and Ray Li. Polynomial time decodable codes for the binary deletion channel. *IEEE Trans. Inf. Theory*, 65(4):2171–2178, 2019.

[GL21] Tom Gur and Oded Lachish. On the power of relaxed local decoding algorithms. *SIAM J. Comput.*, 50(2):788–813, 2021.

[GM12] 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.

[GRR18] Tom Gur, Govind Ramnarayan, and Ron D. Rothblum. Relaxed locally correctable codes. In *ITCS*, pages 27:1–27:11, 2018.

[GS16] Venkatesan Guruswami and Adam Smith. Optimal rate code constructions for computationally simple channels. *J. ACM*, 63(4):35:1–35:37, September 2016.

[GW17] Venkatesan Guruswami and Carol Wang. Deletion codes in the high-noise and high-rate regimes. *IEEE Transactions on Information Theory*, 63(4):1961–1970, 2017.

[Hae19] Bernhard Haeupler. Optimal document exchange and new codes for insertions and deletions. In David Zuckerman, editor, *FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 334–347, 2019.

[HO08] Brett Hemenway and Rafail Ostrovsky. Public-key locally-decodable codes. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Proceedings*, pages 126–143, 2008.

[HOSW11] Brett Hemenway, Rafail Ostrovsky, Martin J. Strauss, and Mary Wootters. Public key locally decodable codes with short keys. In *14th International Workshop, APPROX, and 15th International Workshop, RANDOM, Proceedings*, pages 605–615, 2011.

[HOW15] Brett Hemenway, Rafail Ostrovsky, and Mary Wootters. Local correctability of expander codes. *Inf. Comput.*, 243:178–190, 2015.

[HRS19] Bernhard Haeupler, Aviad Rubinstein, and Amirbehshad Shahrasbi. Near-linear time insertion-deletion codes and $(1+\epsilon)$-approximating edit distance via indexing. In Moses Charikar and Edith Cohen, editors, *STOC*, pages 697–708. ACM, 2019.

[HS17] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: codes for insertions and deletions approaching the singleton bound. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *STOC*, pages 33–46. ACM, 2017.

[HS18] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: explicit constructions, local decoding, and applications. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *STOC*, pages 841–854. ACM, 2018.

[HS21] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings and codes for insertions and deletions – a survey, 2021.

[HSS18] Bernhard Haeupler, Amirbehshad Shahrasbi, and Madhu Sudan. Synchronization strings: List decoding for insertions and deletions. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP*, volume 107 of *LIPIcs*, pages 76:1–76:14, 2018.

[KdW04] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. Syst. Sci.*, 69(3):395–420, 2004.

[Kiw05] Expected length of the longest common subsequence for large alphabets. *Advances in Mathematics*, 197(2):480–498, 2005.

[KMRS17] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *J. ACM*, 64(2):11:1–11:42, 2017.

[KS16] Swastik Kopparty and Shubhangi Saraf. Guest column: Local testing and decoding of high-rate error-correcting codes. *SIGACT News*, 47(3):46–66, 2016.

[KSY14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5):28:1–28:20, 2014.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC*, pages 80–86, 2000.

[KV10] Tali Kaufman and Michael Viderman. Locally testable vs. locally decodable codes. In *APPROX-RANDOM*, volume 6302 of *Lecture Notes in Computer Science*, pages 670–682. Springer, 2010.

[Lev66] Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[Lip94] Richard J. Lipton. A new approach to information theory. In *STACS*, pages 699–708, 1994.

[LT13] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

[LTX19] Shu Liu, Ivan Tjuawinata, and Chaoping Xing. On list decoding of insertion and deletion errors. *CoRR*, abs/1906.09705, 2019.

[MBT10] 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.

[Mit08] Michael Mitzenmacher. A survey of results for deletion channels and related synchronization channels. volume 6, pages 1–3, 07 2008.

[MPSW05] 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.

[OPC15] 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.

[OPS07] Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private locally decodable codes. In *ICALP*, pages 387–398, 2007.

[Slo02] N.J.A. Sloane. On single-deletion-correcting codes. *arXiv: Combinatorics*, 2002.

[SS16] 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.

[STV99] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma (abstract). In *CCC*, page 4, 1999.

[SZ99] L. J. Schulman and D. Zuckerman. Asymptotically good codes correcting insertions, deletions, and transpositions. *IEEE Transactions on Information Theory*, 45(7):2552–2557, 1999.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *CoRR*, cs.CC/0409044, 2004.

[WdW05] 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.

[Woo07] David P. Woodruff. New lower bounds for general locally decodable codes. Technical report, Weizmann Institute of Science, Israel, 2007.

[Woo12] 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.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1):1:1–1:16, 2008.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.

[YGM17] S. M. Hossein Tabatabaei Yazdi, Ryan Gabrys, and Olgica Milenkovic. Portable and error-free dna-based data storage. *Scientific Reports*, 7:2045–2322, 2017.

A A Note on the Definition of Insdel LDCs

Recall the definition of Insdel LDCs from Definition 1.

Definition 1. *[Insdel Locally Decodable Codes (Insdel LDCs)] Fix an integer q and constants $\delta \in [0,1]$, $\varepsilon \in (0, \frac{1}{2}]$. We say $C\colon \{0,1\}^n \to \Sigma^m$ is a (q,δ,ε)-*locally decodable insdel code *if there exists a probabilistic algorithm* Dec *such that:*

- *For every $x \in \{0,1\}^n$ and $y \in \Sigma^{m'}$ such that $\mathsf{ED}\left(C(x), y\right) \leq \delta \cdot 2m$, and for every $i \in [n]$, we have*

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

where the probability is taken over the randomness of Dec*, and $\mathsf{ED}\left(C(x), y\right)$ denotes the minimum number of insertions/deletions necessary to transform $C(x)$ into y.*

- *In every invocation,* Dec *reads at most* q *symbols of* y*. We say that* Dec *is* non-adaptive *if the distribution of queries of* $\mathsf{Dec}(y, m', i)$ *is independent of* y*.*

In our definition of Insdel LDCs we assume that the decoder Dec is directly given m', the length of the corrupted codeword y. Arguably it may be more reasonable to require Dec to recover x_i without a priori knowledge of the length m'. This question does not arise in the definition of Hamming LDCs as the length of the corrupted codeword is fixed. If we do not give the Insdel decoder the length m' then Dec may query for an out of range index $j > m'$ and we would need to define how such queries are handled e.g., if Dec queries for $y[j]$ for $j > m'$ we might return $\perp$ to indicate that the query is out of range. In this case the decoder could always recover m' after $O(\log m)$ queries by using binary search to find the maximum j such that $y[j] \neq \perp$.

We stress that giving the local decoder access to m' can only help the decoder. If the information is not helpful the decoder can always chose to ignore the extra information m'. Since our focus is on proving lower bounds we chose to give Dec access to the length m' which only makes the lower bounds stronger.

Another modification of Definition 1 might allow for the Insdel codewords $C(x)$ to have variable length i.e., $C : \Sigma^n \to \Sigma^{\leq m}$. Now if we require that this insdel distance between $C(x)$ and the corrupted codeword y is at most $2\delta|C(x)|$ and if we additionally give Dec access to the length $m' = |y|$ of of the corrupted codeword then the encoding algorithm can "cheat" and use codeword length to encode x. For example, when $\Sigma = \{0, 1\}$ and $\delta < 1/6$ we could define a $(q = 0, \delta, \epsilon = \frac{1}{2})$-insdel LDC $C : \{0,1\}^n \to \{0,1\}^{\leq m}$ with $m = 2^{2^n}$ as follows: define a bijective mapping $\mathtt{Int} : \{0,1\}^n \to \{0, \ldots, 2^n - 1\}$ in the natural way and then set $C(x) = 1^{2^{\mathtt{Int}(x)+1}}$ i.e., 1 repeated $2^{\mathtt{Int}(x)+1}$ times. If $\delta < 1/6$ then $m' = |y|$ must lie in the range

$$\frac{2}{3}2^{\mathtt{Int}(x)+1} < m' < \frac{4}{3}2^{\mathtt{Int}(x)+1} = \frac{2}{3}2^{\mathtt{Int}(x)+2}$$

since we require that the insdel distance between $C(x)$ and y is at most $2\delta|C(x)| < 2^{\mathtt{Int}(x)+1}/3$. This allows the local insdel decoder to recover the entire message x (and any particular bit x_i) from m' without any queries to the corrupted codeword y. In particular, the decoder could find the unique integer k such that $\frac{2}{3}2^{2^{k+1}} < m' < \frac{2}{3}2^{k+2}$ and then recover $x = \mathtt{Int}^{-1}(k)$.

Arguably the above construction "cheats" by encoding the message x in the length of the codeword (unary) and allowing the decoder to directly learn the length of the (corrupted) codeword. If we do not allow the decoder to directly learn the length m' of the corrupted message then there are no known constructions of $(q = 2, \delta, \epsilon)$-insdel LDCs for any constants $\delta, \epsilon > 0$ — for any information rate n/m.

B Definition of Private-Key (Insdel) LDCs

In the private-key setting the encoder $C(x; R)$ and decoder $\mathsf{Dec}(y, m', i; R)$ are given access to shared (secret) set of random coins R which is not given to the channel $\mathcal{A}$. Fixing x and R we say that a corrupted codeword y ϵ-fools the decoder if there exists an index $i \leq n$ such that $\Pr[\mathsf{Dec}(y, m', i; R) = x_i] < \frac{1}{2} + \epsilon$. In the classical setting (no shared randomness) we require that no corrupted codeword y with $\mathsf{ED}\left(C(x; R), y\right)$ has the property that it ϵ-fools the decoder. In the private-key setting we relax this requirement and allow that ϵ-fooling codewords y exist so long as the probability the channel $\mathcal{A}$ outputs such a codeword is negligible. Formally, let

$\texttt{FoolED}(\mathcal{A}, x, R) = 1$ (resp. $\texttt{FoolHamm}(\mathcal{A}, x, R) = 1$) denote the event that the corrupted codeword $y = \mathcal{A}(x, C(x; R))$ output by the channel ϵ-fools the decoder and $\mathsf{ED}\left(C(x; R), y\right) \leq 2\delta m$ (resp. $\texttt{Hamm}(C(x; R), y) \leq \delta m$). Crucially, the random coins R are not known to the channel otherwise $\mathcal{A}$ could do a brute-force search to find such an ϵ-fooling string.

We say that the pair (C, Dec) is a (q, δ, ϵ)-private-key LDC for Insdel (resp. Hamming) errors if there is a negligible function $\mu(\cdot)$ such that for all channels $\mathcal{A}$ and messages $x \in \Sigma^n$ we have $\Pr_R[\texttt{FoolED}(\mathcal{A}, x, R)] \leq \mu(n)$ (resp. $\Pr_R[\texttt{FoolHamm}(\mathcal{A}, x, R)] \leq \mu(n)$).

Prior Private-Key Constructions [OPS07] gives a (q, δ, ϵ)-private-key Hamming LDC with query complexity $q = \log^2 n$ and $\epsilon > \frac{1}{2} - \mu(n)$ for a negligible function $\mu(n)$ i.e., except with negligible probability $\mu(n)$ the channel outputs a codeword y such that $\forall i \leq n$ we have $\Pr[\mathsf{Dec}(y, m', i; R) = x_i] \geq 1 - \mu(n)$. [BB21] gives a (q, δ, ϵ)-private-key Insdel LDC with query complexity $q = \log^c n$ and $\epsilon > \frac{1}{2} - \mu(n)$ for a negligible function $\mu(n)$. The result is obtained by applying the Hamming to Insdel compiler of [BBG+20] to [OPS07].

$q = 1$ query private-key Hamming LDC We can also extend ideas from [OPS07] to obtain a $(q = 1, \delta, \epsilon)$-private-key Hamming LDC for suitable constants $\delta + \epsilon < \frac{1}{2}$. The length of the codeword is just $m = n \log^2 n$. In specific we define $C(x; R = (\pi, \texttt{OTP})) = \pi\left(x^t \oplus \texttt{OTP}\right)$. Here, $\texttt{OTP} \in \{0,1\}^{tn}$ is a uniformly random tn-bit string used as a one-time-pad and $x^t \in \{0,1\}^{tn}$ denotes the string x concatenated to itself t times i.e., $x^1 \doteq x$ and $x^{i+1} \doteq x^i \circ x$. Note that for any x the string $\texttt{OTP} \oplus x^t$ is distributed uniformly at random. The random permutation $\pi : [tn] \to [tn]$ randomly shuffles the bits of $z = x^t \oplus \texttt{OTP}$ e.g., if $z = z[1] \circ \ldots \circ z[tn]$ then $\pi(z) = z[\pi(1)] \circ \ldots \circ z[\pi(tn)]$.

Fixing π and an index $i \leq n$ we can define $S_{\pi,i} \doteq \{\pi(i), \ldots, \pi(i + tn)\}$ to be the set of indices of the codeword $C(x; \pi, \texttt{OPT})$ which correspond to $x[i]$ i.e., such that $C(x; \pi, \texttt{OPT})[j] \oplus \texttt{OTP}[j] = x[i]$. The decoder $\mathsf{Dec}(y, m', i; R = (\pi, \texttt{OTP}))$ will randomly pick $j \in S_{\pi,i}$ and return $y[j] \oplus \texttt{OTP}[j]$ as our guess for $x[i]$. Now a corrupted codeword y ϵ-fools the decoder if and only if for some $i \leq n$ at least $\left(\frac{1}{2} - \epsilon\right)$-fraction of the bits in $S_{\pi,i}$ were flipped i.e., $\left|\left\{j \in S_{\pi,i} : y[j] = C(x; \pi, \texttt{OPT})[j]\right\}\right| \leq t\left(\frac{1}{2} + \epsilon\right)$.

Because the channel $\mathcal{A}$ does not have π or $\texttt{OTP}$ and can flip at most δm bits in the codeword the expected number of bit flips in $S_{\pi,i}$ is just $\delta t < \left(\frac{1}{2} - \epsilon\right) t$. Applying concentration bounds and union bounding we have

$$\Pr[\exists i.\ \left|\left\{j \in S_{\pi,i} : y[j] = C(x; \pi, \texttt{OPT})[j]\right\}\right| \leq t\left(\frac{1}{2} + \epsilon\right)] \leq \mu(n)$$

for a negligible function $\mu(n)$ whenever we set $t = \log^2 n$. Here, the randomness is taken over the selection of π and $\texttt{OTP}$.

C A Note on Hadamard Codes and Type 1 Errors

In this section, we consider the following error pattern corresponding to error type 1 mentioned in the technique overview i.e., pick arbitrary $e \leq \delta m/2$ and delete the first e bits from the codeword and append e arbitrary bits at the end of the codeword. In particular, if y is a codeword with length m then the error pattern D_e deletes the first e bits of the codeword y and then insert e arbitrary bits to the end. We denote the the corrupted codeword we obtained after this error pattern by $D_e(y)$.

This section shows that a variant of the Hadamard code allows the local decoder to recover from type 1 errors using just $q = 2$ queries.

We prove the following theorem.

Theorem 6. *There exist explicit* $(2, \delta, 1/2 - \delta - 2^{-t})$ *insdel LDCs* $C\colon \{0,1\}^n \to \{0,1\}^m$ *with* $m = 2^{nt}$*, which corrects error Type* ***1****.*

Let Had_n be the Hadamard encoding for message length n. Given a string $\mathbf{x} = x_0 x_2 \dots x_{n-1} \in \{0,1\}^n$, we view the codeword $\mathsf{Had}_n(\mathbf{x})$ as a function $f_{\mathbf{x}}\colon [2^n - 1] \to \{0,1\}^n$, where

$$\forall a \in [2^n - 1] \text{ with binary representation } a = \sum_{j=0}^{n-1} a_j \cdot 2^j, \quad f_{\mathbf{x}}(a) = \bigoplus_{j=0}^{n-1} a_j x_j.$$

Encoder Let $t \in \mathbb{N}$ be a parameter. For a message $\mathbf{x} = x_0 x_1 \cdots x_{n-1} \in \{0,1\}^n$, let $\mathbf{x}^{(t)} \coloneqq (x_0 0^{t-1})(x_1 0^{t-1}) \dots (x_{n-1} 0^{t-1})$, i.e. the string $\mathbf{x}$ with $t-1$ zeros following each bit. The encoder function is given by

$$\mathsf{Enc}(\mathbf{x}) = \mathsf{Had}_{tn}(\mathbf{x}^{(t)}).$$

Therefore the codeword length is $m = 2^{tn}$. Note that x_i has index ti in $\mathbf{x}^{(t)}$, we can recover $x_i = f_{\mathbf{x}^{(t)}}(2^{ti})$.

Decoder Consider the following decoder Dec. Let $f\colon [2^{tn} - 1] \to \{0,1\}$ be the received string. To decode x_i, Dec picks a random $a \in [2^{tn} - 1]$ such that $a + 2^{ti} \le 2^{tn} - 1$ and outputs $f(a) \oplus f(a + 2^{ti})$.

Analysis For integers $a, b \in \mathbb{N}$ with binary representations a_j and b_j we write $a \le_2 b$ if $a_j \le b_j$ for all j. We say $a \le 2^{tn} - 2^{ti} - 1$ is *bad* for i if $2^{ti}(2^t - 1) \le_2 a$, i.e. $a_j = 1$ for all $ti \le j \le t(i+1) - 1$. Otherwise we say a is *good* for i.

The success rate of the decoder is implied by the following two claims.

Claim 12. *If* a *is good for* i*, then for any* $\mathbf{x} \in \{0,1\}^n$*,* $f_{\mathbf{x}^{(t)}}(a) \oplus f_{\mathbf{x}^{(t)}}(a + 2^{ti}) = x_i$*.*

Proof. Let a_j and a'_j be the binary representations of a and $a + 2^{ti}$, respectively. Note that $a_j = a'_j$ for $j < ti$, and $a_{ti} \ne a'_{ti}$. If a is good for i, we must also have $a_j = a'_j$ for $j \ge t(i+1)$, since otherwise there exists $j_0 \ge t(i+1)$ such that $a'_{j_0} = 1$ and $a_{j_0} = 0$, and then

$$2^{ti} = (a + 2^{ti}) - a = \sum_{j=ti}^{j_0} (a'_j - a_j) 2^j > 2^{j_0} - \sum_{j=ti}^{j_0 - 1} 2^j = 2^{ti},$$

which is a contradiction. The inequality is strict because $a_j = 0$ for at least one $ti \le j \le j_0 - 1$.

Finally, note that $x_j^{(t)} = 0$ for $ti < j < t(i+1)$. We thus have

$$f_{\mathbf{x}^{(t)}}(a) \oplus f_{\mathbf{x}^{(t)}}(a + 2^{ti}) = \bigoplus_{j=ti}^{t(i+1)-1} a_j x_j^{(t)} \oplus \bigoplus_{j=ti}^{t(i+1)-1} a'_j x_j^{(t)} = a_{ti} x_{ti}^{(t)} \oplus a'_{ti} x_{ti}^{(t)} = \left(a_{ti} \oplus a'_{ti}\right) x_i = x_i.$$

□

Claim 13. *For any* $0 \le i \le n - 1$*,* $\Pr_{a \in [2^{tn} - 2^{ti} - 1]}[a \text{ is good for } i] \ge 1 - 2^{-t}$*.*

Proof. Note that for $a \geq 2^{t(i+1)}$, $2^{ti}(2^t-1) \leq_2 a$ if and only if $2^{ti}(2^t-1) \leq_2 a - 2^{t(i+1)}$. Therefore we only need to show $\Pr_{a\in[2^{t(i+1)}-1]}[a \text{ is good for } i] \geq 1-2^{-t}$. Clearly a is good for i when $a < 2^{ti}(2^t-1)$, and hence

$$\Pr_{a\in[2^{tn}-2^{ti}-1]}[a \text{ is good for } i] \geq \frac{2^{ti}(2^t-1)}{2^{t(i+1)}} = \frac{2^t-1}{2^t} = 1-2^{-t}.$$

□

If the distances among the indices are preserved, each deletion at the beginning of the codeword reduces the number of good indices by at most 1. So after δm deletions we are left with at least

$$(1-2^{-t})(2^{tn}-2^{ti}) - \delta\cdot 2^{tn} = (1-\delta)2^{tn} - 2^{t(n-1)} - 2^{ti} \geq (1-\delta-2^{-t+1})2^{tn}$$

good a's for i. Therefore it is a $(2, \delta, 1/2-\delta-2^{-t+1})$ LDC for such errors.

D Proof of Theorem 4

The following theorem says that from any linear (resp. affine) insdel LCC, we can obtain a linear (resp. affine) insdel LDC that has the same parameter. The theorem is essentially Lemma 2.3 from [Yek12].

Theorem 7. *Let $\mathbb{F}$ be a finite field. Suppose $C \subseteq \mathbb{F}^m$ is a linear (resp. affine) (q,δ,ε)-insdel LCC, then there exists a linear (resp. affine) (q,δ,ε)-insdel LDC C' encoding messages of length $\mathsf{dim}(C)$ to codewords of length m. The same holds for the insdel LCCs and insdel LDCs in the private-key setting.*

Proof of Theorem 7. For any linear code $C \subseteq \mathbb{F}^m$, it encodes a message $\mathbf{x} \in \mathbb{F}^n$ to a codeword $\mathbf{y} \in \mathbb{F}^m$ through encoding function $\mathbf{y} = \mathbf{x}\cdot G$ with generating matrix $G \in \mathbb{F}^{n\times m}$. Let $I \subseteq [m]$ be a set of $\mathsf{dim}(C)$ information coordinates of C (i.e. a set of coordinates whose value uniquely determines an element in C). For $\mathbf{y} \in C$, let $\mathbf{y}|_I \in \mathbb{F}^n$ be the restriction of $\mathbf{y}$ to the coordinates in I. We can find another generator matrix G' such that for any message $\mathbf{x} \in \mathbb{F}^n$, $\mathbf{y} = \mathbf{x}\cdot G' \in C$ and $\mathbf{y}|_I = \mathbf{x}$. It is easy to verify that the locally correctability of C implies the locally decodability of C'.

If C is an affine code, the encoding function becomes $\mathbf{y} = \mathbf{x}\cdot G + \mathbf{b}$ for some $\mathbf{b} \in \mathbb{F}^m$. Again, let $I \subseteq [m]$ be a set of $\mathsf{dim}(C)$ information coordinates of C. Similarly, we can pick generator matrix G' such that for any message $\mathbf{x} \in \mathbb{F}^n$, $\mathbf{y} = \mathbf{x}\cdot G' + \mathbf{b} \in C$ and $\mathbf{y}|_I - \mathbf{b}|_I = \mathbf{x}$. The locally correctability of C implies the locally decodability of C'.

We note the above argument holds for insdel LCCs and insdel LDCs in the private-key setting. □

The proof of Theorem 4 uses the same reduction introduced by [BGT16]. We first introduce two lemmas. The following lemma says that insdel LCCs must have large Hamming distance.

Lemma 5. *If $C \in \Sigma^m$ is a (q,δ,ε)-LCC, then for any two codewords $c, c' \in C$, the Hamming distance between c and c' is larger than $2\delta m$.*

Proof. Assume there are two codewords $c, c' \in C$ such that $c \neq c'$ and the Hamming distance between c and c' is at most $2\delta m$. Then we can conclude that $\mathsf{ED}(c, c') \leq 4\delta m$. This is because we can transform c into c' with less then $2\delta m$ symbol substitutions. Meanwhile, each symbol substitution can be replaced by an insertion and a deletion. Thus it takes at most $4\delta m$ insertions/deletions to transform c into c'.

Since $\mathsf{ED}(c, c') \leq 4\delta m$, there must exists a $y \in \Sigma^{m'}$ such that $\mathsf{ED}(c, y) \leq 2\delta m$ and $\mathsf{ED}(c', y) \leq 2\delta m$. Let i be one of the positions that c and c' differs. By the definition of insdel LCCs, there is a probabilistic algorithm Dec such that $\Pr\left[\mathsf{Dec}(y, m', i) = c_i\right] \geq \frac{1}{2} + \varepsilon$ and $\Pr\left[\mathsf{Dec}(y, m', i) = c'_i\right] \geq \frac{1}{2} + \varepsilon$. This is impossible because $c_i \neq c'_i$. Thus, the Hamming distance between c and c' must be larger than $2\delta m$.

□

The reduction needs the following notion of VC-dimension.

Definition 6 (VC-dimension)**.** *Let $A \subseteq \{0,1\}^n$, then the VC-dimension of A, denoted by $\mathsf{vc}(A)$ is the cardinality of the largest set $I \subseteq [n]$ which is shattered by A. That is, the restriction of A to I, $A|_I = \{0,1\}^{|I|}$.*

The following lemma due to Dudley [Dud78] is the key to the reduction.

Lemma 6 (Theorem 14.12 in [LT13])**.** *Let $A \subseteq \{0,1\}^n$ such that for every distinct $x, y \in A$, $\|x - y\|_{\ell_2} \geq \varepsilon\sqrt{n}$. Then,*

$$\mathsf{vc}(A) = \Omega\left(\frac{\log(|A|)}{\log(2/\varepsilon)}\right).$$

Proof of Theorem 4. Assume $C : \{0,1\}^k \to \Sigma^m$ is a (q, δ, ε)-insdel LCC. Without loss of generality, we can assume $\Sigma = \{0,1\}^s$. Consider an error correcting code $C^0 : \{0,1\}^s \to \{0,1\}^t$ with constant distance δ_0, i.e. any two codewords in C^0 has Hamming distance $\delta_0 t$. Let $C^1 : \{0,1\}^k \to \{0,1\}^{mt}$ be the concatenation of code C and C^0. That is, every codeword in C^1 is obtained by first encoding the message with code C and then encoding each symbol in Σ with code C^0. By Lemma 5, any two codewords in C^1 must have Hamming distance at least $2\delta\delta_0 mt$. Thus, let $\varepsilon = \sqrt{2\delta\delta_0}$, for any $c, c' \in C^1$, we have

$$\|c - c'\|_{\ell_2} \geq \varepsilon\sqrt{mt}.$$

By Lemma 6, the VC-dimension of C^1 ($\mathsf{vc}(C^1)$) is

$$\Omega\left(\frac{\log(|C^1|)}{\log(2/\varepsilon)}\right) = \Omega\left(\frac{\log(|C|)}{\log(2/\varepsilon)}\right) = \Omega\left(\frac{k}{\log(1/\delta)}\right)$$

Let $k' = \mathsf{vc}(C^1)$. By the definition of VC-dimension, there exists a set $I \subseteq [mt]$ of size k' that shatters C^1, i.e. $C^1|_I = \{0,1\}^{|I|}$.

We can build a (q, δ, ε)-insdel LDC $C' : \{0,1\}^{k'} \to \Sigma^m$ as follows. For any message $x \in \{0,1\}^{k'}$, let $C'(x) = z \in C$ such that $C^0(z)|_I = x$. Here, by $C^0(z)$, we mean encoding each symbol of z with code C^0. By the property of set I, we know such a codeword $z \in C$ must exist. If there are more than one $z \in C$ satisfying $C^0(z)|_I = x$, we pick one of them arbitrarily. Assuming $I = \{t_1, t_2, \ldots, t_{k'}\}$, for any message $x \in \{0,1\}^{k'}$ and any $i \in [k']$, we have $x_i = \left(C^0(C'(x))\right)_{t_i}$.

We now show that C' is indeed a (q, δ, ε)-insdel LDC. To decode a message bit x_i, we only need to look at the block of $\left(\{0,1\}^t\right)^m$ that contains t_i, say it is the j-th block. Assume we are given a

$y \in \Sigma^*$ with $\mathsf{ED}\left(y, C'(x)\right) \leq 2\delta m$, and notice that $C'(x)$ is also a codeword of C. The decoding algorithm for C can recover $C'(x)_j$ with probability of at least $1/2 + \varepsilon$, using at most q queries to y. Applying C^0 to $C'(x)_j$ will give us x_i. Thus, x_i can be decoded with probability at least $1/2 + \varepsilon$ using at most q queries to y.

□