On the Lower Bound on the Length of Relaxed Locally Decodable Codes

Oded Goldreich[*]
Department of Computer Science
Weizmann Institute of Science, Rehovot, Israel.

May 3, 2023

Abstract

We revisit the known proof of the lower bound on the length of relaxed locally decodable codes, providing an arguably simpler exposition that yields a slightly better lower bound for the non-adaptive case and a weaker bound in the general case.

Recall that a locally decodable code is an error correcting code that allows for the recovery of any desired bit in the message based on a constant number of randomly selected bits in the possibly corrupted codeword. The relaxed version requires correct recovery only in case of actual codewords, while requiring that for strings that are (only) close to the code, with high probability, the local decoder outputs either the correct value or a special failure symbol (but not a wrong value).

The lower bounds we prove are $n \geq k^{1+\Omega(1/q^2)}$ for the non-adaptive case and $n \geq k^{1+\Omega(1/q^3)}$ for the general case, where k denotes the message length, n denotes the length of the codewords, and q denotes the (constant) number of queries.

Contents

[*] Partially supported by the Israel Science Foundation (grant No. 1041/18) and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702).

1 Introduction

A locally decodable code (LDC) is a (binary) error correcting code that allows for the recovery of any desired bit in the message based on a constant number of (randomly selected) bits in the possibly corrupted codeword.

Locally decodable codes, or rather family of such codes, have several parameters: The length of the message, denoted k, the length of codewords, denoted n (and viewed as a function of k), the number of queries, denoted q, and the tolerated corruption rate, denoted δ. We shall view $q \in \mathbb{N}$ and $\delta > 0$ as fixed constants, whereas k and $n = n(k)$ are viewed as varying parameters. (This regime is fundamentally different from the one in [13] (and subsequent works), where $n = O(k)$ and q is allowed to be a function of k.)

The conjecture that locally decodable codes require large length (e.g., n must be super-polynomial in k)[1], which was supported by the super-linear lower bound (i.e., $n = \Omega(k^{q/(q-1)})$) of [11], led [3] to suggest a relaxed notion of LDCs. In this relaxation, hereafter referred to as *relaxed LDCs*, the decoder is allowed to announce failure and two conditions are made:

1. When given access to a valid codeword, the local decoder recovers the desired bit with probability at least 2/3.

2. When given access to a string that is δ-close to a valid codeword (i.e., the relative Hamming distance between the string and the codeword is at most δ), with probability at least 2/3, the local decoder does not err; that is, with probability at least 2/3, it outputs either the desired bit or a special failure symbol.

As shown in [3, Sec. 4.2.1], such relaxed LDCs can be transformed to ones in which $1 - O(\Delta(w))$ of the bits of the message can be correctly recovered (with probability at least 2/3) when the local decoder is given access to a string w that is $\Delta(w)$-close to the code. (In the case of non-adaptive decoders, the resulting relaxed LDCs maintain the parameters of the original ones up to a constant factor.)

More importantly, as shown in [3, Sec. 4.2.2], relaxed LDCs of polynomial length exist. Specifically, for every sufficiently large constant q, one can obtain a q-query relaxed LDC of length $n = k^{1+O(1/\sqrt{q})}$. This result was later improved (in [2]) to $n = k^{1+O(1/q)}$. While these relaxed LDCs are much shorter than the best known (super-polynomial length) constructions of LDCs (of [16, 5]), these relaxed LDCs are not shorter than the known lower bound for LDCs, which are $n = \Omega(k/\log k)^{1+\frac{1}{\lceil q/2 \rceil - 1}}$ (cf. [12], improving over [11]). Hence, the known results fail to separate relaxed LDCs from LDCs.

The hope of separating relaxed LDCs from LDCs turned out to be hard to materialize. A formidable difficulty has recently emerged from the work of [9, 4], which shows that q-query relaxed LDC must have length $n = k^{1+\Omega(1/q \log q)^2}$. Hence, the hope that a separation can be obtained by a construction of a relaxed LDC of length $n = \widetilde{O}(k)$ was shuttered, and the ballpark for improved constructions of relaxed LDCs is (conceptionally) smaller. Still, a q-query relaxed LDC of length $n = k^{1+o(1/q)}$ would yield the desired separation.

In any case, the main purpose of this work is to provide a simple exposition of the result of [9, 4], and actually improve it a bit in the non-adaptive case. That is, we prove the following

[1] It was even conjectured that $n > \exp(k^{\Omega(1)})$, but this conjecture was refuted in [16, 5].

Theorem 1.1 (main result): *Suppose that $C : \{0,1\}^k \to \{0,1\}^n$ is a q-query relaxed LDC. Then, $n \geq k^{1+\Omega(1/q^3)}$. Furthermore, if the relaxed decoder is non-adaptive, then $n \geq k^{1+\Omega(1/q^2)}$ holds.*

The key ideas that underlie our proof are rooted in [9, 4], but the low level details are quite different. Specifically, we do not analyze the set system of q-tuples with the aim of finding "relaxed sunflowers" (a.k.a daisies), but rather use a rather trivial 3-way partition of the queries (to heavy, intermediate, and light types). This allows us to avoid the error reduction, which is used in [9, 4], and obtain a slightly better lower bound in the non-adaptive case (i.e., $n \geq k^{1+\Omega(1/q)^2}$ rather than $n \geq k^{1+\Omega(1/q \log q)^2}$).

The pivot of the analysis is the global decoding procedure, used by [9], for the case of non-adaptive one-sided error local decoders. Our presentation starts with some simplifying assumptions, and gradually builds the argument for relaxed-LDCs that have a non-adaptive one-sided error local decoder (see Section 2). Later, the argument is extended to the two-sided error case, but still assuming non-adaptivity (see Section 3). These proofs are relatively simple, and we dare say that we believe that their exposition is very intuitive and appealing.

Unfortunately, the foregoing cannot be said about the proof of the adaptive case. The starting point is a rather natural adjustment of the global decoding procedure to the adaptive case (presented in Section 4.1). Actually, the latter presentation avoid a crucial detail, which turns out to be difficult to complete. This difficulty is addressed in Section 4.2, alas at the cost of a weaker bound (cf. the gap between the bounds stated in Theorem 1.1).

1.1 Preview of the case of non-adaptive one-sided error relaxed LDC

As stated above, the pivot of the analysis is a *global* decoding procedure, which views each bit of the (uncorrupted) n-bit codeword with probability p. This global decoder invokes executions of the q-query local decoder, in the hope of recovering the bits of the message. If this decoder is able to recover all k bits of the message, then $n = \Omega(k/p)$ must hold, and if p is small (e.g., $p = k^{-\Omega(1)}$, then we get a meaningful lower bound on n.

In the non-adaptive case, the analysis of the global decoder just relies on the observation that if each bit in the codeword is seen with probability p, then each q-tuple is seen with probability p^q. If the local decoder is actually of a (non-relaxed) LDC, then, as shown in [11], each of its queries is almost uniformly distributed, and in such a case the probability that the global decoder fails to recover a specific bit of the k-long message based on the sampled bits in the n-long codeword is $(1-p^q)^{\Omega(n)}$, where the analysis uses $\Omega(n)$ disjoint q-tuples. Setting $p = (\Theta(\log k)/n)^{1/q}$, we get $(1-p^q)^{\Omega(n)} = o(1/k)$, which means that $p \cdot n$ bits allow for determining a k-bit string, which in turn gives $n = \Omega(k/p) = k^{1+\frac{1}{q-1}-o(1)}$.

Unfortunately, in general, *relaxed* LDCs may make queries that are far from being uniformly distributed (and, indeed, the best known relaxed LDCs do so). The actual difficulty is dealing with the few "heavy" queries that the relaxed LDCs may make; that is, for each $\ell \in [k]$, we consider the set of queries that the local decoder makes with high probability when asked to recover the ℓ^{th} bit of the message. For example, suppose that when asked to recover the ℓ^{th} bit, the local decoder always queries the ℓ^{th} bit of the codeword.

The solution is to guess the value of the heavy queries, and use the relaxed-LDC guarantee in order to argue that this will not foil the global decoder. The latter claim holds, because the number of heavy queries is small and their corruption cannot lead the local decoder to output a wrong bit value (except for with small probability). (This solution is taken from [9], where it is applied to the "kernels" of the "relaxed sunflowers"; here, we apply it to "heavy" queries.) Furthermore, if the non-heavy queries are actually light, then the analysis may use uses q-tuples that are disjoint on their light queries. Specifically, if each light query occurs with probability at most $1/m$, then $\Omega(m)$ such tuples exist.

Hence, for each $\ell \in [k]$, a crucial issue is to 3-partition the queries into *heavy*, *intermediate* and *light* types, with the intension of guessing the value of the heavy queries, hoping to sample a q-tuple consisting only of heavy and light queries, and ignoring the intermediate queries. The key observation is that most q-tuple of queries that correspond to executions of the local decoder (1) do not contain queries of intermediate weights, and, (2) are "good" with respect to a fixed guess of values for all the heavy queries. Specifically, for every guess for the value of the heavy queries, most of the foregoing q-tuples provide evidence against a wrong value of the ℓ^{th} bit of the message.

We mention that we can afford to ignore the intermediate queries by choosing a 3-way partition such that the intermediate queries carry a total probability mass of $O(1/q)$. This leads us to set $p = n^{-1/\Theta(q^2)}$, and obtain $n = \Omega(k/p) = k^{1+\frac{1}{O(q^2)}}$, where the same bound holds also in the two-sided case. (Handling the two-sided error case is slightly more complex, because in this case there may be errors also when we use the correct guess for the heavy queries; hence, we must rule by majority, while facing the fact that the different votes are not totally independent.)

As noted above, a weaker lower bound holds in the adaptive case, and the argument in that case is significantly more complicated. We mention that the lower bound of [9] referred only to non-adaptive one-sided error local decoders, and the extension to the general case was done in [4]. The special case is sufficiently interesting given that it covers all known constructions, let alone the fact that general q-query local decoders can be emulated by 2^q-query non-adaptive local decoders.

1.2 Actual definitions

For the sake of good order, we recall the standard definition of *relaxed* locally decodable codes (relaxed LDC (rLDC)), while viewing $\delta > 0$ and $q \in \mathbb{N}$ are fixed constants, whereas k and $n = n(k)$ are viewed as varying parameters.

Definition 1.2 (relaxed LDC): *We say that $C : \{0,1\}^k \to \{0,1\}^q$ is a q-query relaxed LDC (q-rLDC) if there exists a randomized oracle machine D, called a local decoder, that makes q queries such that the following two conditions hold.*

1. *For every $x \in \{0,1\}^k$ and $\ell \in [k]$,*

$$\Pr[D^{C(x)}(\ell) = x_\ell] \geq 2/3.$$

If $\Pr[D^{C(x)}(\ell) = x_\ell] = 1$ for every x and ℓ, then we say that the decoder has one-sided error. Otherwise, we say that it has two-sided error.

2. *For every $x \in \{0,1\}^k$, every $w \in \{0,1\}^n$ that is δ-close to $C(x)$, and every $\ell \in [k]$,*

$$\Pr[D^w(\ell) \in \{x_\ell, \perp\}] \geq 2/3$$

where $\perp \notin \{0,1\}$ is a special symbol and w is δ-close to y if $|\{i \in [n] : w_i \neq y_i\}| \leq \delta \cdot n$.

The fixed parameter δ is called the decoding distance of D.

Indeed, δ must be smaller than half the relative distance of C, and q is the query complexity of the local decoder. The (original) *non-relaxed* notion of a locally decidable code is obtained by requiring that D never outputs $\bot$. We may assume, without loss of generality, that the local decoder always makes q queries and that it never makes the same query twice.

We shall focus on *non-adaptive decoders*; that is, decoders that make queries that are determined solely by their input and random choices (independently of answers provided to prior queries). Such decoders, studied in Sections 2 and 3, consists of two modules: A querying module, denoted Q, that, on input $\ell \in [k]$, outputs a random q-subset of $[n]$, and a decoding module, denoted D', which on input $\ell \in [k]$ and $(i_1, ..., i_q) \in [n]^q$ and $(b_1, ..., b_q) \in \{0,1\}^q$, outputs a value in $\{0, 1, \bot\}$. In this case, $D^w(\ell)$ equals $D'(\ell, (i_1, ..., i_q), (w_{i_1}, ..., w_{i_q}))$, where $(i_1, ..., i_q) \leftarrow Q(\ell)$. We shall often view $Q(\ell)$ as a set rather than a sequence.

Additional notation. For $w \in \{0,1\}^n$ and $I \subseteq [n]$, we let w_I denote the restriction of w to the locations I; that is, if $I = \{i_1, ..., i_m\}$ such that $i_j < i_{j+1}$ for every $j \in [m-1]$, then $w_I = w_{i_1} w_{i_2} \cdots w_{i_m}$, where w_i is the i^{th} bit of w. Similarly, for $f : S \to \{0,1\}$ and $I \subseteq S$, we let $f(I)$ denote the restriction of f to location I; that is, $f(I) = (f(i_1), ..., f(i_m))$.

A convention. Whenever the query complexity (i.e., q) or functions of it, appears as a multiple of k (or n), we may hide it in the O-notation (or Ω-notation). We definitely do not do this otherwise; in these other cases the O-notation (or Ω-notation) hides only universal constants that are independent of q.

2 The case of non-adaptive one-sided error relaxed LDCs

As stated in the introduction, the analysis of non-adaptive local decoders is significantly simpler than the analysis of general (i.e., adaptive) ones. Nevertheless, this restricted setting allows for the introduction of the main ideas. Things are further simplified by restricting attention to one-sided error local decoders, which is done in the current section. The argument will be extended to the two-sided error case, but still assuming non-adaptivity, in Section 3. (The analysis of general (adaptive) local decoders is presented in Section 4.1.)

We make the simplifying assumption that Q has randomness complexity $O(1) + \log_2 n$ (see [9, Clm. 4.3]). Equivalently, on input $\ell \in [k]$, the querying module determines an n'-long sequence of (not necessarily distinct) q-subsets, where $n' = O(n)$, and output one of them selected uniformly at random. (Actually, the assumption $n' = O(n)$ is never used; yet, it is likely to help some readers.)

Warm-up: The case of LDCs. In the case of (non-relaxed) LDCs, it is known (see [11]) that the queries generated by the decoder are "smooth" in the sense that, for each $\ell \in [k]$, each specific query $i \in [n]$ is made with probability $O(1/n)$; that is, $\Pr[Q(\ell) \ni i] = O(1/n)$. It follows that each possible query i appears in $O(1)$ of the q-subsets that the query module generates on input ℓ. This fact, which does not hold for related LDCs that are not LDCs, will be used in the analysis of the following simplified construction of a global decoder.

Loosely speaking, this global decoder is specified by a parameter $p \in [0,1]$; it queries each bit of the given codeword $C(x)$ with probability p, with the aim of reconstructing the entire codeword (equivalently, the encoded message x). If the decoder manages to recover x (say, with probability at least 2/3), then it must hold that $p \cdot n = \Omega(k)$. The global decoder acts in the straightforward manner detailed next.

Construction 2.1 (the global decoder, with parameter p, take 1): *On input $C(x) \in \{0,1\}^n$, the global decoder proceeds as follows.*

1. *It selects a random sample S such that each $i \in [n]$ is included in S with probability p, independently of all other choices, and obtains $C(x)_i$ for each $i \in S$ by querying $C(x)$.*
2. *For each $\ell \in [k]$, the global decoder tries to retrieve x_ℓ as follows.*
   - *Let $(Q_1, ..., Q_{n'})$ be the sequence of q-subsets that $Q(\ell)$ uses; that is, $Q(\ell)$ selects $r \in [n']$ uniformly at random and outputs Q_r.*
   - *If there exists $r \in [n']$ such that $Q_r \subseteq S$, then x_ℓ is set according to $D'(\ell, Q_r, C(x)_{Q_r})$. Otherwise, the decoder fails.*

If all x_ℓ's were retieved, then the global decoder outputs $x_1 x_2 \cdots x_k$.

Turning to the analysis, we observe that the global decoder never retrieves a wrong answer, because the local decoder never errs on codewords (i.e., it has one-sided error). Furthermore, for each $\ell \in [k]$, the probability that the decoder fails to retrieve x_ℓ is given by $\Pr_S[(\forall r \in [n'])\ Q_r \not\subseteq S]$. Using the foregoing smoothness condition (i.e., $\Pr_{r\in[n']}[Q_r \ni i] = O(1/n)$ for every $i \in [n]$), it follows that each Q_r intersects $O(1)$ other subsets. Hence, $(Q_1, ..., Q_{n'})$ contains a sub-collection of $\Omega(n)$ pairwise disjoint Q_r's; that is, a set $I \subset [n']$ of size $\Omega(n)$ such that for every $r \neq s \in I$ it holds that $Q_r \cap Q_s = \emptyset$. It follows that

$$\begin{aligned}
\Pr_S[(\forall r \in [n'])\ Q_r \not\subseteq S] &\leq \Pr_S[(\forall r \in I)\ Q_r \not\subseteq S] \\
&= \prod_{r\in I} \Pr_S[Q_r \not\subseteq S] \\
&= (1 - \Pr_S[\{1, ..., q\} \subseteq S])^{|I|} \\
&= (1 - p^q)^{\Omega(n)}.
\end{aligned}$$

Now, using a sufficiently large $p = O((\log k)/n)^{1/q}$, we get

$$\begin{aligned}
(1 - p^q)^{\Omega(n)} &= \left(1 - \frac{O(\log k)}{n}\right)^{\Omega(n)} \\
&= o(1/k).
\end{aligned}$$

Hence, the global decoder retrieves x with probability $1 - o(1)$. On the other hand, using $p \cdot n \geq k$, we get $n \geq k/p = k \cdot (n/O(\log k))^{1/q}$, and $n \geq k^{\frac{q}{q-1} - o(1)}$ follows. Thus, we actually established

Theorem 2.2 (lower bound for one-sided error non-adaptive (non-relaxed) LDC): *Suppose that $C : \{0,1\}^k \to \{0,1\}^n$ is a q-query LDC in which the local decoder uses non-adaptive queries and has one-sided error. Then, $n > k^{1+\frac{1}{q-1}-o(1)}$.*

Another warm-up: A special case of rLDCs. As stated upfront, in general, relaxed LDCs are not smooth, and so the analysis presented above does not hold for them. For example, the simplest non-trivial rLDC known uses $n = k^{2+o(1)}$, and, on input $\ell \in [k]$, it always queries a fixed location, denoted i_ℓ, in the codeword (see [3, Sec. 4.2.2]). However, each of the other $q-1$ queries that this local decoder makes is uniformly distributed in some set of size at least $m \overset{\text{def}}{=} n/k$. Clearly, in this case we cannot expect the global decoder to sample the fixed location $i_\ell \in [n]$. Instead, we may let it try *both possible values* for location i_ℓ, and using $p \gg (1/m)^{1/(q-1)}$ we can reasonably hope that the sample covers one of the residual $(q-1)$-subset, and (unreasonably) hope that this prevents a wrong setting of x_ℓ. This would indeed be the case if the local decoder were error-free on non-codewords, but error-freeness is only guaranteed for codewords. Nevertheless, using all residual $(q-1)$-subsets that are covered by the sample, we set the value of x_ℓ only if all these subsets support the same bit value (when combined with the guess value of location i_ℓ). Otherwise, we try the other alternative for the value of location i_ℓ.

More generally, let us consider a local decoder that, on input $\ell \in [k]$, makes q' queries that are confined to a subset H_ℓ (of "heavy" queries) and makes $q'' \overset{\text{def}}{=} q - q'$ queries that are each uniformly distributed in some subset of size at least $m_\ell \gg |H_\ell|$. Actually, it suffices to assume that each of the other queries hit each possible location with probability $O(1/m_\ell)$; that is, for each $i \notin H_\ell$, it holds that $\Pr[Q(\ell) \ni i] = O(1/m_\ell)$. In this case, we use the following global decoder.

Construction 2.3 (the global decoder, with parameter p, take 2): *On input $C(x) \in \{0,1\}^n$, the global decoder proceeds as follows.*

1. *As in Construction 2.1, the global decoder selects a random sample S such that each $i \in [n]$ is included in S with probability p, independently of all other choices, and obtains $C(x)_i$ for each $i \in S$ by querying $C(x)$.*

2. *For each $\ell \in [k]$, the global decoder tries to retrieve x_ℓ as follows.*

   - *Let $(Q_1, ..., Q_{n'})$ be the sequence of q-subsets that $Q(\ell)$ uses, and let H_ℓ be determined accordingly to $Q(\ell)$.*

     (The specific determination of H_ℓ is left open at this point; we shall eventually use Claim 2.6 for this.)

   - *Let*
     $$R_\ell = R_\ell(S) \overset{\text{def}}{=} \{r \in [n'] : (Q_r \setminus H_\ell) \subseteq S\} \tag{1}$$
     denote the collection of revealed *residual subsets. If $R_\ell = \emptyset$, then the decoder fails. Otherwise, the decoder tries all possible $\alpha : H_\ell \to \{0,1\}$.*[2]

     For each $\alpha : H_\ell \to \{0,1\}$, the global decoder tries to retrieve x_ℓ as follows.

     - *For each $r \in R_\ell$, we consider the value that is obtained from the local decoder when it makes queries Q_r and get answers according to $\alpha(Q_r \cap H_\ell))$ and $C(x)_{Q_r \setminus H_\ell}$; that is, the queries in H_ℓ are answered according to α, whereas the answers to the queries outside of H_ℓ were already obtained from $C(x)$* (since $(Q_r \setminus H_\ell) \subseteq S$).

[2] Here and in subsequent constructions, it seems more intuitive to try all possible $\alpha : (H_\ell \setminus S) \to \{0,1\}$, because the global decoder knows $C(x)_S$. Yet, the analysis is simpler when ignoring this fact (and not using $C(x)_{S \cap H_\ell}$).

- *If the obtained values are all identical to some bit, then x_ℓ is set accordingly; that is, x_ℓ is set to the value $b \in \{0,1\}$ if and only if $b = D'(\ell, Q_r, C(x)_{Q_r \setminus H_\ell}, \alpha(Q_r \cap H_\ell))$ for every $r \in R_\ell$. Otherwise* (i.e., there is no consensus to a value that is in $\{0,1\}$), *we continue to the next α.*

(Note that, when $R_\ell \neq \emptyset$, some iteration will set x_ℓ; specifically, when $\alpha(H_\ell) = C(x)_{H_\ell}$, the bit x_ℓ is set (correctly)).

If all x_ℓ's were retieved, then the global decoder outputs $x_1 x_2 \cdots x_k$.

Turning to the analysis, in addition to upper-bounding the probability of failure to retrieve some x_ℓ (i.e., the case of $R_\ell = \emptyset$), which is done analogously to the first warm-up (except that here $|I| = \Omega(m_\ell)$), we also need to upper-bound the probability of setting x_ℓ wrongly. Specifically, we will show that the probability that a specific choice of $\alpha : H_\ell \to \{0,1\}$ yields a wrong value for x_ℓ is $o(2^{-|H_\ell|}/k)$. This will follow by a suitable setting of p in the next claim; that is, using a sufficiently large $p = O((|H_\ell| + \log k)/m_\ell)^{1/q''}$ will do.

Lemma 2.4 (upper bound on the error probability of the global decoder of Construction 2.3): *Suppose that for every $\ell \in [k]$ it holds that $|H_\ell| = o(n)$, and that for each $i \notin H_\ell$, it holds that* $\Pr[Q(\ell) \ni i] \leq 1/m_\ell$. *Then, for every $\ell \in [k]$ and $x \in \{0,1\}^k$, and for every $\alpha : H_\ell \to \{0,1\}$, the probability that the corresponding iteration of Construction 2.3 sets x_ℓ wrongly is at most* $\exp(-\Omega(p^{q''} \cdot m_\ell))$.

Hence, the probability that Construction 2.3 is wrong (on some $\ell \in [k]$ due to some α) is at most $\sum_{\ell \in [k]} 2^{|H_\ell|} \cdot \exp(-\Omega(p^{q''} \cdot m_\ell))$. Using a sufficiently large $p = O((|H_\ell| + \log k)/m_\ell)^{1/q''}$, the error probability is upper-bounded by $\sum_{\ell \in [k]} 2^{|H_\ell|} \cdot \exp(-(|H_\ell| + \log k)) = o(1)$, which yields a lower bound of $n \geq k/p = \min_{\ell \in [k]}\{\Omega(m_\ell/(|H_\ell| + \log k))^{1/q''}\} \cdot k$. So we get a meaningful lower bound whenever $m_\ell \geq (|H_\ell| + 1) \cdot k^{\Omega(1)}$ for every $\ell \in [k]$.

Proof: Fixing $x \in \{0,1\}^k$ and $\ell \in [k]$, our aim is to prove that, for every $\alpha : H_\ell$, it holds that

$$\Pr_S[(\forall r \in R_\ell(S))\, D'(\ell, Q_r, C(x)_{Q_r \setminus H_\ell}, \alpha(Q_r \cap H_\ell)) = 1 - x_\ell] \;\leq\; \exp(-\Omega(p^{q''} \cdot m_\ell)).$$

Fixing $\alpha : H_\ell \to \{0,1\}$, we say that $r \in [n']$ is good (i.e., good for α) if the value obtained from the local decoder when it makes queries Q_r and get answers according to $C(x)_{Q_r \setminus H_\ell}$ and $\alpha(Q_r \cap H_\ell)$) is either x_ℓ or $\perp$; that is, r is good if $D'(\ell, Q_r, C(x)_{Q_r \setminus H_\ell}, \alpha(Q_r \cap H_\ell)) \in \{x_\ell, \perp\}$. Note that, for a good r, if $Q_r \setminus H_\ell$ is covered by the sample S (i.e., $r \in R_\ell(S)$), then this prevents a consensus for the wrong value $1 - x_\ell$, which means that the corresponding iteration does not set the ℓ^{th} bit of x to a wrong value. Hence, we need to upper-bound the probability that no good r satisfies $(Q_r \setminus H_\ell) \subseteq S$ (equiv., no good r is in $R_\ell(S)$).

Letting G denote the set of good r's and using $|H_\ell| \leq \delta \cdot n$, it follows that $|G| \geq 2n'/3$, because $\Pr[D^w(\ell) \in \{x_\ell, \perp\}] \geq 2/3$ must hold for w such that $w_i = \alpha(i)$ if $i \in H_\ell$ and $w_i = C(x)_i$ otherwise (since w is δ-close to $C(x)$). Analogously to the first warm-up, we wish to identify a set of $\Omega(m_\ell)$ choices r in G such that the corresponding Q'_r's are pairwise disjoint, where $Q'_r \stackrel{\text{def}}{=} Q_r \setminus H_\ell$. Such an $\Omega(m_\ell)$-subset of G, denoted G', exists because for every $s \in [n] \setminus H_\ell$ it holds that

$$\Pr_{r \in G}[Q'_r \ni s] \;\leq\; \frac{3}{2} \cdot \Pr_{r \in [n']}[Q'_r \ni s] \;=\; O(1/m_\ell),$$

which implies that for every $s \in G$ it holds that $\Pr_{r \in G}[Q'_r \cap Q'_s \neq \emptyset] = O(1/m_\ell)$, which in turn yields a greedy procedure for finding G' (cf. Lemma 3.1).[3] Using G', we upper-bound the probability that no good choice is covered (i.e., $G \cap R_\ell(S) = \emptyset$) as follows

$$\begin{aligned}
\Pr_S[G \cap R_\ell(S) = \emptyset] &= \Pr_S[(\forall r \in G)\ Q'_r \not\subseteq S] \\
&\leq \Pr_S[(\forall r \in G')\ Q'_r \not\subseteq S] \\
&= \prod_{r \in G'} \Pr_S[Q'_r \not\subseteq S] \\
&= \left(1 - p^{q''}\right)^{|G'|} \\
&= \left(1 - p^{q''}\right)^{\Omega(m_\ell)}.
\end{aligned}$$

The claim follows. ■

Digest and beyond: The analysis of Construction 2.3 is pivoted at hypothesis that $m_\ell/|H_\ell|$ is large, where H_ℓ is the set of heavy queries and $O(1/m_\ell)$ is an upper bound on the probability mass assigned to each non-heavy query (i.e., queries not in H_ℓ). Furthermore, the analysis (captured by Lemma 2.4) holds also if q'' is only an upper-bound on the number of non-heavy queries (i.e., on $\max_i\{|Q_i \setminus H_\ell|\}$). Hence, we get the following

Theorem 2.5 (the lower bound, special case): *Let Q denote the querying module of a q-query local decoder for* (the rLDC) $C : \{0,1\}^k \to \{0,1\}^n$, *and $\epsilon > 0$ be arbitrary. Suppose that, for every $\ell \in [k]$, there exists a set H_ℓ of size $o(n)$ such that, for every $i \in [n] \setminus H_\ell$, it holds that $\Pr[Q(\ell) \ni i] \leq \epsilon/(|H_\ell|+1)$. Then, $n = \Omega(\epsilon^{-1}/\log k)^{1/q} \cdot k$.*

Proof: Using a sufficiently large $p = O(\epsilon \cdot \log k)^{1/q}$ and applying Lemma 2.4 (with $q'' = q$ and $m_\ell = (|H_\ell|+1)/\epsilon$), we conclude that the probability that Construction 2.3 fails to recover x (when given orcale access to $C(x)$) is at most

$$\sum_{\ell \in [k]} 2^{|H_\ell|} \cdot \exp(-\Omega(p^q \cdot (|H_\ell|+1)/\epsilon)) \ \leq \ \sum_{\ell \in [k]} 2^{|H_\ell|} \cdot \exp(-((|H_\ell|+1) \cdot \log k)) \ = \ o(1).$$

Hence, we get a lower bound of $n = \Omega(k/p) = k/O(\epsilon \cdot \log k)^{1/q}$. (Specifically, one may consider fixing a set S of size approximately $p \cdot n$ such that the global decoder is successful in rerieving a $1 - o(1)$ fraction of the x's.) ■

In light of Theorem 2.5, we are faced with a challenge of finding, for each $\ell \in [k]$, a (heavy) set H_ℓ such that $\max_{i \in [n] \setminus H_\ell}\{\Pr[Q^{\mathrm{R}}(\ell) = i]\} \ll 1/|H_\ell|$, where $Q^{\mathrm{R}}(\ell)$ denotes a query selected uniformly at random in the q-subset $Q(\ell)$. Unfortunately, this phrasing of the problem yields very poor results (e.g., consider the case that, for every $i \in [n]$, it holds that $\Pr[Q^{\mathrm{R}}(\ell) = i] = \Theta(1/i \log n)$).[4] Fortunately, we can relax the foregoing requirement and allow for ignoring a set of queries of total

[3] The argument here is much simpler: Each $r \in G$ that we include in G', rules out at most $t \stackrel{\text{def}}{=} O(|G|/m_\ell)$ other elements of G, which implies that we can get $|G'| \geq |G|/(t+1)$.

[4] In this case, for every non-empty set $H \subset [n]$, it holds that $\max_{i \in [n] \setminus H}\{\Pr[Q^{\mathrm{R}}(\ell) = i]\} = \Omega(1/|H| \log n)$.

probability mass $1/10q$ (or so). Specifically, we aim at identifying a 3-way partition (H_ℓ, M_ℓ, L_ℓ) of $[n]$ such that $|H_\ell| = o(n)$, $\Pr[Q^{\mathtt{R}}(\ell) \in M_\ell] \leq 1/10q$, and $\max_{i \in L_\ell}\{\Pr[Q^{\mathtt{R}}(\ell) = i]\} \ll 1/|H_\ell$. Indeed, the set L_ℓ consists of light queries, which we shall use, whereas the set M_ℓ is a "middle" set of queries (with intermediate probability weight), which we shall ignore.

Claim 2.6 (creating a gap between heavy and light elements): *Let ζ be a random variable that is distributed over $[n]$, and $c \in \mathbb{N}$. Then, there exists a 3-way partition, (H, M, L), of $[n]$ such that* (i) $|H| < n^{(c-1)/c}$, (ii) $\Pr[\zeta \in M] \leq 1/(c-2)$, *and* (iii) *for every $z \in L$ it holds that $\Pr[\zeta = z] \leq n^{-1/c}/(|H|+1)$.*

Note that Claim 2.6 does not even guarantee that $L \neq \emptyset$, but in our application of Claim 2.6 (see Corollary 2.7) it be the case that $|L| = \Omega(|H| + |M|)$ because $Q^{\mathtt{R}}(\ell)$ has support size $\Omega(n)$ (cf. the proof of Lemma 2.4).

Proof: For every $j \in [c-1]$, let

$$B_j \stackrel{\rm def}{=} \left\{ z \in [n] : \Pr[\zeta = z] \in (n^{-j/c}, n^{-(j-1)/c}] \right\} \tag{2}$$

and note that $|B_j| < n^{j/c}$. Let $B_c \stackrel{\rm def}{=} \{z \in [n] : \Pr[\zeta = z] \leq n^{-(c-1)/c}\}$. Using an arbitrary $t \in \{2, ..., c-1\}$ such that $\Pr[\zeta \in B_t] \leq 1/(c-2)$, we let $H \stackrel{\rm def}{=} \bigcup_{j \leq t-1} B_j$ and $L \stackrel{\rm def}{=} \bigcup_{j \geq t+1} B_j$. Then, $|H| < n^{(t-1)/c}$, whereas for every $z \in L$ it holds that $\Pr[\zeta = z] \leq n^{-t/c}$. Letting $M = B_t$, the claim follows. ■

Corollary 2.7 (the actual lower bound for one-sided error local decoders): *Suppose that $C : \{0,1\}^k \to \{0,1\}^n$ is a q-query relaxed LDC in which the local decoder uses non-adaptive queries and has one-sided error. Then, $n > k^{1+\frac{1}{10(q+1)^2}}$.*

Proof: Let Q be the querying module of the relaxed local decoder D. Using Claim 2.6 with $\zeta = Q^{\mathtt{R}}(\ell)$ and $c = 10q+2$, for each $\ell \in [k]$, we obtain (H_ℓ, M_ℓ, L_ℓ) such that (1) $|H_\ell| = o(n)$, (2) $\Pr[Q(\ell) \cap M_\ell \neq \emptyset] \leq 0.1$, and (3) for every $i \in L_\ell$ it holds that $\Pr[Q(\ell) \ni i] = O(n^{-1/(10q+2)}/(|H_\ell| + 1))$. (Indeed, we used $\Pr[Q(\ell) \cap M_\ell \neq \emptyset] \leq q \cdot \Pr[Q^{\mathtt{R}}(\ell) \in M_\ell]$.)

We stress that the 3-way partition (H_ℓ, M_ℓ, L_ℓ) can be determined by the global decoder of Construction 2.3, based on $Q(\ell)$. (It is instructive to restate (2) as $\Pr[(Q(\ell) \setminus H_\ell) \not\subseteq L_\ell] \leq 0.1$.) Recall that, for a fixed $\ell \in [k]$, we denote the possible outcomes of $Q(\ell)$ by $Q_1, ..., Q_{n'}$. Hence, $\Pr_{r \in [n']}[(Q_r \setminus H_\ell) \not\subseteq L_\ell] \leq 0.1$.

We shall use the same definition of a good choice as in the proof of Lemma 2.4. Recall that, for fixed $\ell \in [k]$ and $x \in \{0,1\}^k$, when considering any $\alpha : H_\ell \to \{0,1\}$, a choice $r \in [n']$ is defined as good (i.e., good for α) if selecting the corresponding query-set Q_r leads the local decoder to output either x_ℓ or $\perp$. (Recall that the fraction of good choices is at least 2/3).[5]

[5] Recall that this is proved by considering $w \in \{0,1\}^n$ that is δ-close to $C(x)$ (i.e., $w_i = \alpha(i)$ if $i \in H_\ell$ and $w_i = C(x)_i$ otherwise), and observing that the local decoder D must satisfy $\Pr[D^w(\ell) \in \{x_\ell, \perp\}] \geq 2/3$, which means that

$$\Pr_{r \in [n'}[D'(\ell, Q_r, C(x)_{Q_r \setminus H_\ell}, \alpha(Q_r \cap H_\ell)) \in \{x_\ell, \perp\}] \geq 2/3.$$

We comment that, by the same reasoning, the set of good r's must make queries that cover more than $\delta \cdot n$ elements of $[n]$, which implies that $|M_\ell \cup L_\ell| \geq \delta \cdot n - o(n)$. Furthermore, the same holds also for the set of "effectively" good r's (defined next), which implies that $|L_\ell| \geq \delta \cdot n - o(n)$.

We proceed as in the proof of Theorem 2.5, except that we adapt the rest of the proof of Lemma 2.4 so to account for the fact that we can use a good choice r only if $Q_r \setminus H_\ell$ is a subset of L_ℓ (where, unlike in the original proof, it is not guaranteed that $L_\ell = [n] \setminus H_\ell$). This means that the fraction of "effectively" good r's (i.e., good r's such that $Q_r \setminus H_\ell \subseteq L_\ell$) may decrease from 2/3 to $\frac{2}{3} - 0.1 > 0.55$, which is immaterial for the rest of the argument (i.e., we just consider the set of *effectively good* choices rather than the set of good choices considered originally). Note that we are invoking Theorem 2.5 with $\epsilon = O(n^{-1/(10q+2)})$, and obtain the lower bound

$$n \;=\; \Omega(\epsilon^{-1}/\log k)^{1/q} \cdot k \;=\; \Omega(k^{1/(10q+2)}/\log k)^{1/q} \cdot k \;>\; k^{1/(10q^2+2q)} \cdot k^{1-o(1)}.$$

The claim follows. ■

Digest. The crucial fact used in the proof of Corollary 2.7 is that the set of effectively good choices has cardinality $\Omega(n')$. Actually, as in the proof of Theorem 2.5, it is immaterial that this set holds a clear majority (be it 55% or 66%) among all n' choices. In both cases, the foregoing fact is combined with the third hypothesis (i.e., $\Pr[Q(\ell) \ni i] = O(n^{-1/(10q+2)}/(|H_\ell|+1))$ for every $i \in L_\ell$) to yield a subset G' (of the effectively good choices) that is "hit" by the sample S (in the sense that there exists $r \in G'$ such that $Q'_r \subseteq S$) with probability at least $1 - o(2^{-|H_\ell|}/k)$. Observe that this means that we can afford to set the constant c in Claim 2.6 to $2q+2$, obtain $\Pr[Q(\ell) \cap M_\ell \neq \emptyset] \leq 0.5$ (rather than $\Pr[Q(\ell) \cap M_\ell \neq \emptyset] \leq 0.1$), and derive a lower bound of $n > k^{1+\frac{1}{2(q+1)^2}}$. The setting of $c = 10q+2$, which yields a clear majority of effectively good choices, was used in order to streamline with the the next section, where we analyze local decoders with two-sided error.

3 The case of non-adaptive two-sided error relaxed LDCs

Recall that a two-sided error relaxed LDC is the general case defined in Definition 1.2; that is, we are only guaranteed that, for every $\ell \in [k]$ and $x \in \{0,1\}^k$, it holds that $\Pr[D^{C(x)}(\ell) = x_\ell] \geq 2/3$ (rather than $\Pr[D^{C(x)}(\ell) = x_\ell] = 1$). In contrast, so far, our global decoders were based on the hypothesis that $\Pr[D^{C(x)}(\ell) = x_\ell] = 1$. Specifically, in these global decoders, each bit of x is set to a bit value only if *all* potential values retrieved for it (by different invocations of D') were in *consensus*.

It is tempting to revise these constructions so as to require only a majority towards this bit value (among all retrieved values), but the problem is that the retrieval attempts are not totally independent. This fact is relevant because our analysis was based on identifying a large enough set of independent trials and using this set in our probabilistic analysis. This was adequate in the context of the hitting problem that arises from the analysis of the one-sided error local decoder; but it is not adequate in the context of the approximation problem that arises from the analysis of the two-sided error local decoders. Instead, we use the following generic fact, which is of independent interest.

Lemma 3.1 (implicit in the proof of [1, Thm. 1]): *Let V be the vertex set of a graph with maximal degree at most d, and suppose that $(\zeta_v)_{v \in V}$ is a sequence of random variables assigned values in $[0,1]$ such that for every independent set S in the graph it holds that the random variables in the* *subsequence $(\zeta_v)_{v\in S}$ are totally independent. Then, for any $\gamma > 0$, it holds that*

$$\Pr\left[\left|\frac{1}{|V|}\cdot\sum_{v\in V}\zeta_v - \mu\right| \geq \gamma\right] \;=\; (d+1)\cdot\exp(-\Omega(\gamma^2\cdot|V|/(d+1)))$$

where $\mu \stackrel{\rm def}{=} \sum_{v\in V}{\rm E}[\zeta_v]/|V|$. In particular, if $\gamma \geq B \stackrel{\rm def}{=} O(\sqrt{(d\log d)/|V|})$, then the upper bound simplifies to $\exp(-\Omega(\gamma^2\cdot|V|/(d+1)))$.

The foregoing bound should be contrasted with a bound of $\exp(-\Omega(\gamma^2\cdot|V|))$ that holds in the case that the random variables $(\zeta_v)_{v\in V}$ are totally independent (equiv., the graph has no edges). Indeed, in the general case (of $d \geq 1$), we lose a factor of $1/(d+1)$ in the exponent, but this is unavoidable (e.g., consider a graph that consists of $|V|/(d+1)$ isolated $(d+1)$-cliques). The original proof of Lemma 3.1, which is reproduced below, uses a combinatorial theorem of [10]. A alternative proof that avoids the use of [10] was suggested to us by Noga Alon and is presented in the appendix.

Proof: The starting point is the aforementioned theorem (of [10]) that asserts that the vertices of such a graph can be partitioned into $(d+1)$ independent sets, denoted $S_1, ..., S_{d+1}$, such that $\lfloor|V|/(d+1)\rfloor \leq |S_i| \leq \lceil|V|/(d+1)\rceil$ for every $i \in [d+1]$. Letting $\sigma_i \stackrel{\rm def}{=} \sum_{v\in S_i}{\rm E}[\zeta_v]$, we observe that

$$\begin{aligned}
\Pr\left[\left|\sum_{v\in V}\zeta_v - \mu\cdot|V|\right| \geq \gamma\cdot|V|\right] &\leq \sum_{i\in[d+1]}\Pr\left[\left|\sum_{v\in S_i}\zeta_v - \sigma_i\right| \geq \gamma\cdot|S_i|\right] \\
&\leq \sum_{i\in[d+1]}\exp(-\Omega(\gamma^2\cdot|S_i|)) \\
&\leq (d+1)\cdot\exp(-\Omega(\gamma^2\cdot|V|/(d+1)))
\end{aligned}$$

and the claim follows. ■

The majority-based decision rule. We now spell out the new (majority-based) decision rule, which replaces the consensus decision rule used in Construction 2.3. Recall that, fixing $\ell \in [k]$, we considered the set $R_\ell \stackrel{\rm def}{=} \{r\in[n'] : (Q_r \setminus H_\ell) \subseteq S\}$, where the Q_r's are the q-subsets used by $Q(\ell)$ and S is the sample used in Step 1 of the global decoder. Actually, observing that we did not gain from M_ℓ anyhow, we define the set of admissible choices

$$A_\ell \stackrel{\rm def}{=} \{r\in[n'] : (Q_r \setminus H_\ell) = (Q_r \cap L_\ell)\} \;=\; \{r\in[n'] : (Q_r \cap M_\ell) = \emptyset\}, \tag{3}$$

while noting that $|A_\ell| \geq 0.9\cdot n'$, and redefine R_ℓ as follows

$$R_\ell = R_\ell(S) \stackrel{\rm def}{=} \{r\in A_\ell : (Q_r \cap L_\ell) \subseteq S\}. \tag{4}$$

Another difficulty is that different choices $r \in A_\ell$ are not necessarily included in $R_\ell(S)$ with the same probability, because $\Pr_S[r \in R_\ell(S)] = p^{|Q_r\cap L_\ell|}$ (for each $r \in A_\ell$). Hence, we include each $r \in R_\ell(S)$ in the "redefined R_ℓ", denoted R'_ℓ, with probability $p^{|Q_r\cap H_\ell|}$. With this modification in place, each $r \in A_\ell$ is included in R'_ℓ with probability p^q.

Construction 3.2 (majority rule for the global decoder, when using a two-sided error local decoder): *Fixing an assignment $\alpha : H_\ell \to \{0,1\}$, for each $r \in R'_\ell = R'_\ell(S)$, let $v_r \in \{0,1,\perp\}$ denote the value that is obtained from the local decoder when it makes queries Q_r and get answers according to $C(x)_{Q_r \cap L_\ell}$ and $\alpha(Q_r \cap H_\ell)$); that is, the queries in H_ℓ are answered according to α, whereas the queries in L_ℓ were obtained from the oracle* (by the hypothesis $(Q_r \cap L_\ell) \subseteq S$). *Then, if a strict majority of the v_r's equal a bit $b \in \{0,1\}$* (i.e., $|\{r \in R'_\ell : v_r = b\}| > |R'_\ell|/2$),[6] *then we set $x_\ell \leftarrow b$. Otherwise, we continue to the next α.*

Indeed, $v_r = D'(\ell, Q_r, C(x)_{Q_r \cap L_\ell}, \alpha(Q_r \cap H_\ell))$ denotes the vote of $r \in R'_\ell$ based on $C(x)_{Q_r \cap L_\ell}$ and $\alpha(Q_r \cap H_\ell)$. Note that, unlike in Construction 2.3, the global decoder may fail (rather than err) also when $R'_\ell \neq \emptyset$; this happens when no bit enjoyed a strict majority in any of the possible $\alpha : H_\ell \to \{0,1\}$. But as shown next, this bad event occurs with small probability.

Theorem 3.3 (lower bound for two-sided error local decoders): *Suppose that $C : \{0,1\}^k \to \{0,1\}^n$ is a two-sided error q-query relaxed LDC in which the local decoder uses non-adaptive queries. Then, $n > k^{1+\frac{1}{20(q+1)^2}}$.*

Proof: We shall use the global decoder of Construction 2.3, while using Construction 3.2 for handling each candidate $\alpha : H_\ell \to \{0,1\}$ (instead of using the consensus rule outlined in Step 2 of Construction 2.3). Recall that Construction 3.2 refers to the definition of R'_ℓ, which builds upon the definitions of A_ℓ and R_ℓ (see Eq. (3) and Eq. (4), resp.).[7] Fixing $x \in \{0,1\}^k$ and $\ell \in [k]$, our aim is to prove that the foregoing global decoder recovers x_ℓ correctly with probability $1 - o(1/k)$.

Let Q be the querying module of the relaxed LDC. As in the proof of Corollary 2.7, using Claim 2.6 (with $c = 10q + 2$ and $\epsilon = O(n^{-1/c})$), we obtain a 3-way partition of $[n]$, denoted (H_ℓ, M_ℓ, L_ℓ), such that (1) $|H_\ell| = O(n^{1-(1/(10q+2))})$, (2) $\Pr[(Q(\ell) \setminus H_\ell) \not\subseteq L_\ell] \leq 0.1$, and (3) for every $i \in L_\ell$ it holds that $\Pr[Q(\ell) \ni i] \leq \epsilon/(|H_\ell|+1)$, where $\epsilon \overset{\text{def}}{=} O(n^{-1/(10q+2)})$.

Consider the graph defined on the vertex set $V_\ell \overset{\text{def}}{=} A_\ell = \{r \in [n'] : Q_r \subseteq H_\ell \cup L_\ell\}$, where r and s are connected if $Q'_r \cap Q'_s \neq \emptyset$, where $Q'_v = Q_v \cap L_\ell$. (Note that if $Q_r \subseteq H_\ell$ (equiv., $Q'_r = \emptyset$), then r is an isolated vertex in this graph and that r is included in R'_ℓ with probability p^q independently of any other event.) Recalling that $|V_\ell| \geq 0.9 \cdot n'$, note that vertices in the graph have degree at most $d \overset{\text{def}}{=} (\epsilon/(|H_\ell|+1)) \cdot n'$. Applying Lemma 3.1 to this graph, while letting $\zeta_r(S) = 1$ if $r \in R'_\ell(S)$ and $\zeta_r(S) = 0$ otherwise (and noting that $\mu = p^q$ and using $\gamma = 0.01 \cdot p^q$), we conclude that $|R'_\ell| = (1 \pm 0.01) \cdot p^q \cdot |V_\ell|$ holds with probability at least

$$\begin{aligned} 1 - \exp(-\Omega(\gamma^2 \cdot |V_\ell|/d)) &= 1 - \exp\left(-\Omega\left((0.01 \cdot p^q)^2 \cdot \frac{0.9 \cdot n'}{((\epsilon/(|H_\ell|+1)) \cdot n'}\right)\right) \\ &= 1 - \exp\left(-\Omega\left(p^{2q} \cdot \frac{|H_\ell|+1}{\epsilon}\right)\right) \\ &= 1 - o(1/k) \end{aligned}$$

where the last transition uses $|H_\ell| \geq 0$ and $\Omega(p^{2q}/\epsilon) > 2 \ln k$. This (as well as using the simpler bound of Lemma 3.1) requires using a sufficiently large $p = O(\epsilon \cdot \log k)^{1/2q}$ (rather than $p = O(\epsilon \cdot \log k)^{1/q}$ as in the proof of Theorem 2.5).

[6] Indeed, if $R'_\ell = \emptyset$, then the condition does not hold.

[7] Recall that Eq. (4) replaces Eq. (1).

Next, we adapt the analysis of the error probability (i.e., proof of Lemma 2.4) in order to show that, when given $\ell \in [k]$ and oracle access to $C(x)$, it holds that

1. the global decoder is unlikely to retrieve a wrong value for x_ℓ (i.e., it rarely errs);
2. the global decoder is likely to retrieve the correct value for x_ℓ (i.e., it rarely fails).

(Indeed, Condition 1 corresponds to Lemma 2.4, whereas Condition 2 corresponds to the analysis of the first warm-up.) Towards proving that the foregoing conditions hold (w.h.p.), for every $\alpha : H_\ell \to \{0,1\}$, we define $r \in V_\ell$ as *good* (good for α) if the following holds:

The case of $\alpha(H_\ell) \neq C(x)_{H_\ell}$ (as in the proof of Lemma 2.4)**:** In this case, the choice $r \in V_\ell$ is called good if the value obtained from the local decoder when it makes queries Q_r and get answers according to $C(x)_{Q_r \setminus H_\ell}$ and $\alpha(Q_r \cap H_\ell)$ is either x_ℓ or $\perp$; that is, r is good if $D'(\ell, Q_r, C(x)_{Q_r \setminus H_\ell}, \alpha(Q_r \cap H_\ell)) \in \{x_\ell, \perp\}$.

The case of $\alpha(H_\ell) = C(x)_{H_\ell}$ (implicit in the analysis of the first warm-up)**:** In this case, the choice $r \in V_\ell$ is called good if the value obtained from the local decoder when it makes queries Q_r and get answers according to $C(x)_{Q_r \setminus H_\ell}$ and $\alpha(Q_r \cap H_\ell))$ equals x_ℓ.

Our aim is to show that, with probability $1 - o(2^{|H_\ell|}/k)$, a strict majority of the choices $r \in R'_\ell$ are good. This will imply that both aforementioned conditions hold, where the first case implies Condition 1 (since each good r votes *against* $1 - x_\ell$) and the second case implies Condition 2 (since each good r votes *in favor* of x_ℓ).

Letting G denote the set of good r's, recall that in either case it holds that $|G| \geq \frac{2}{3} \cdot n'$. Using $|V_\ell| \geq 0.9 \cdot n'$, it follows that $|G \cap V_\ell| > 0.55 \cdot n' \geq 0.55 \cdot |V_\ell|$. Applying Lemma 3.1 to the foregoing graph, while letting $\zeta_r(S) = 1$ if $r \in R'_\ell(S) \cap G$ and $\zeta_r(S) = 0$ otherwise[8], we conclude that

$$\begin{aligned}
\Pr_S\left[|R'_\ell(S) \cap G| \geq 0.54 \cdot p^q \cdot |V_\ell|\right] &\geq \Pr_S\left[|R'_\ell(S) \cap G| = (p^q \cdot |G \cap V_\ell| \pm 0.01 \cdot p^q \cdot |V_\ell|)\right] \\
&\geq 1 - \exp(-\Omega((0.01 \cdot p^q)^2 \cdot |V_\ell|/d)) \\
&= 1 - \exp(-\Omega(p^{2q} \cdot (|H_\ell| + 1)/\epsilon)) \\
&\geq 1 - o(2^{-|H_\ell|}/k),
\end{aligned}$$

where the last inequality uses a sufficiently large $p = O(\epsilon \cdot \log k)^{1/2q}$. Recalling that $|R'_\ell| = (1 \pm 0.01) \cdot p^q \cdot |V_\ell|$ (w.h.p.), it follow that in this (highly likely) case it holds that $|R'_\ell(S) \cap G| > \frac{0.54}{1.01} \cdot |R'_\ell| > |R'_\ell|/2$. Using a union bound over all $\alpha : H_\ell \to \{0,1\}$, we conclude that the global decoder correctly retrieves x_ℓ with probability $1 - o(1/k)$.

As in Section 2, it follows that $n = \Omega(k/p)$. Using $p = O(\epsilon \cdot \log k)^{1/2q}$ and recalling that $\epsilon = O(n^{-1/(10q+2)})$, implies that $n > k^{1/(20q^2+4q)} \cdot k^{1-o(1)}$. ■

Digest. Moving from the one-sided error case to the general (two-sided error) case amounts to replacing the consensus decision rule used in Construction 2.3 by the majority-based decision rule used in Construction 3.2. The analysis of the corresponding global decoder requires coping with the fact that the relevant votes are sampled at random but are not totally independent. The crucial observation here is that the graph of dependencies has relatively low degree and that votes that

[8] Alternatively, we can use the subgraph induced by G and define the random variables as before (i.e., $\zeta_r(S) = 1$ if $r \in R'_\ell(S)$ and $\zeta_r(S) = 0$ otherwise)

correspond to an independent set in the graph are totally independent in the statistical sense. The analysis is completed by employing a generic result that refers to such (partially independent) random variables (captured by Lemma 3.1).

4 The general case (adaptive two-sided error relaxed LDCs)

Turning to the general case (of an adaptive two-sided error local decoder), we warn that the analysis of this case is far more complex than the analysis of the non-adaptive cases, which was presented in the previous sections. We will start by presenting the basic idea that underlies the adjustment of the strategy presented in the previous sections to the current case of adaptive local decoders. This idea will yield a general outline for an adequate global decoder, which is presented in Section 4.1, but this outline faces difficulties that need to be resolved. The latter task is undertaken in Section 4.2, which is the most challenging part of this work.

4.1 The basic strategy and its difficulties

Recall that we associated the set of possible random choices of the local decoder with $[n']$. In the case of a *non-adaptive* local decoder, for a fixed $\ell \in [k]$, each choice $r \in [n']$ corresponds to to a q-subset of $[n]$, which was denoted Q_r; that is, Q_r denotes the set of q queries made by the local decoder (on input ℓ and) when using randomness r. In contrast, in the case of an *adaptive* local decoder, each $r \in [n']$ corresponds to a decision tree of depth q with variables in $[n]$. Denoting this tree by $T_r = T_{\ell,r}$, we view it as a function from the previous answers to the next query; that is, for every $j \in [q]$ and $\overline{a} \in \{0,1\}^{j-1}$, we let $T_r(\overline{a})$ denote the j^{th} query made after obtaining the $j-1$ prior answers represented in $\overline{a}$. (Hence, $T_r : \left(\bigcup_{j\in[q]}\{0,1\}^{j-1}\right) \to [n]$; whereas $T_r : \{0,1\}^q \to \{0,1\}$ represents the final decision of the local decoder as a function of the q answers.)[9]

Our analysis of the *non-adaptive* case has referred to the distribution of queries made by the local decoder; that is, the distribution of a random element of the q-subset Q_r, when r is selected uniformly in $[n']$. Moving to the *adaptive* case, we note that, while the distribution of the first query is oblivious of the n-bit oracle, the distribution of later queries does depend on this oracle. The problem is that, for a generic $r \in [n']$, we (or rather the global decoder) may not know the answers to all prior queries made by T_r. Fortunately, we do know these answers if these queries were either included in the sample S or were deemed heavy (and guessed by the global decoder). Let us detail this idea when referring to the second query.

When considering the distribution of the second query, we shall restrict ourselves to the set of choices, denoted $R^{(1)}$, for which we have obtained the answer to the first query; that is, $r \in R^{(1)}$ if the answer for the first query (i.e., to the query $T_r(\lambda)$), is available to us either because this query is heavy (and so we use a guess for its answer) or because it was included in the sample (i.e., S). (Indeed, the 3-way partition of $[n]$ into heavy, intermediate and light first-queries is defined based on the distribution of the first query (only), in a way that is analogous to Claim 2.6.) Hence, when considering the distribution of the second query, we shall consider only choices in $R^{(1)}$ and define a 3-way partition of the set of second-queries, analogously. Specifically, this distribution is defined by selecting r uniformly at random in $R^{(1)}$ and considering the value of $T_r(v_{T_r(\lambda)})$, where $v_i \in \{0,1\}$ is obtained from a guess (of the global decoder) if $i \in [n]$ is a heavy first-query and from $C(x)_S$

[9] In our description, we shall only refer to the value of T_r on $\bigcup_{j\in[q]}\{0,1\}^{j-1}$.

otherwise (i.e., when $i \in S$). We stress that, for $r \in R^{(1)}$, we do know the answer to its first query by definition of $R^{(1)}$.

Note that, for each assignment to the heavy first-queries, we may obtain a different distribution of the second query, and hence a different 3-way partition of second-query set and a different set of choices $R^{(2)} \subseteq R^{(1)}$ for which we know the answers to the first two queries. The set $R^{(2)}$ will then be used to determine the corresponding distribution of the third query, and so on. Indeed, one may expect that $|R^{(1)}| \approx p \cdot n'$ and similarly $|R^{(2)}| \approx p \cdot |R^{(1)}|$. We proceed analogously till we reach $R^{(q)}$, at which point we proceeds analogously to Construction 3.2. For simplicity, we consider a fresh sample for each of the q iterations (equiv., the q queries of the local decoder).

In light of the foregoing, the global decoder is described as an iterative (or rather recursive) procedure. For each $j \in [q]$, in the $j^{\rm th}$ iteration, the global decoder picks a random sample, denoted $S^{(j)}$, where each element is included with probability p. Using an adequate 3-way partition of the relevant $j^{\rm th}$-queries, it determines $R^{(j)}$ and explores all possible assignments to the heavy $j^{\rm th}$-queries, making a recursive call for each such assignment (when $j < q$). For simplicity, we describe the iteration (or recursion) as treating a single value of $\ell \in [k]$, but we stress that *the same sample $S^{(j)}$ is used for all values of ℓ.*

In the following recursive procedure, the parameter $j \in [q]$ represents the depth of the recursion, and we shall invoke it with $j = 1$. In addition to its main input, which is an n-bit long codeword, and the parameters p and ℓ, the recursive procedure gets auxiliary inputs $R, \overline{H}$ and α such that $R \subseteq [n']$ is a set of surviving indices, $\overline{H} = (H^{(1)}, ..., H^{(j-1)})$ is a $(j-1)$-long sequence of subsets (of $[n]$), which is sometimes viewed as their union (e.g., $|\overline{H}|$ means $|\bigcup_{j' \in [j-1]} H^{(j')}|$), and $\alpha : \overline{H} \to \{0,1\}$, which is actually $\alpha : (\bigcup_{j' \in [j-1]} H^{(j')}) \to \{0,1\}$. A key issue, which is intensionally left open (for now), is the determination of the 3-way partition in Step 2.

The following description refers to a local decoder D for the code $C : \{0,1\}^k \to \{0,1\}^n$. Recall that T_r denotes the decision tree used by $D(\ell)$ to determine its queries, when using randomness $r \in [n']$. The first iteration (i.e., $j = 1$) is invoked with $\overline{H} = \emptyset$ and $R = R_\ell^{(0)} = [n']$.

Construction 4.1 (the $j^{\rm th}$ iteration of the global decoder, with parameters $p \in (0,1]$ and $\ell \in [k]$): *On main input $C(x) \in \{0,1\}^n$ and auxiliary inputs that include a set of surviving choices $R \subseteq [n']$, a $(j-1)$-long sequence of prior heavy queries $\overline{H} = (H^{(1)}, ..., H^{(j-1)})$, and a corresponding assignment $\alpha : \overline{H} \to \{0,1\}$, the current iteration proceeds as follows.*

1. Obtaining a sample: *The global decoder selects a random sample $S^{(j)}$ such that each $i \in [n]$ is included in $S^{(j)}$ with probability p, independently of all other choices, and obtains $C(x)_i$ for each $i \in S^{(j)}$ by querying $C(x)$.*

2. Finding an adequate 3-way partition: *For each $r \in R$, let $q_{r,j}^{\alpha,C(x)}$ denote the $j^{\rm th}$ query of T_r when the $j-1$ first queries are answered according to α if the query is heavy in the relevant iteration and according to $C(x)$ otherwise; that is, $q_{r,j}^{\alpha,C(x)} = T_r(a_1, ..., a_{j-1})$, where, for every $j' \in [j-1]$, it holds that $a_{j'} = \alpha(q_{r,j'}^{\alpha,C(x)})$ if $q_{r,j'}^{\alpha,C(x)} \in \overline{H}^{(j')} \stackrel{\rm def}{=} (H^{(1)}, ..., H^{(j')})$ and $a_{j'} = C(x)_{q_{r,j'}^{\alpha,C(x)}}$ otherwise* (i.e., if $q_{r,j'}^{\alpha,C(x)} \notin \overline{H}^{(j')}$).

   Let ζ be a random variable representing the distribution of $q_{r,j}^{\alpha,C(x)}$ when r is selected uniformly at random in R. Applying a variant of Claim 2.6 to ζ, we obtain a number $s_\ell^{(j)}$ and a 3-way partition $(H_\ell^{(j)}, M_\ell^{(j)}, L_\ell^{(j)})$ of $[n]$ such that

(1) $|H_\ell^{(j)}| < s_\ell^{(j)}$, (2) $\Pr[\zeta \in M_\ell^{(j)}] \leq 1/10q$, *and* (3) *for every* $z \in L_\ell^{(j)}$ *it holds that* $\Pr[\zeta = z] \leq \epsilon / s_\ell^{(j)}$.

(We note that letting $\epsilon = n^{-1/c}$ and applying Claim 2.6 itself (or rather its proof), with $c = 10q^2 + 2q$, would have yielded the foregoing 3-way partition with $s_\ell^{(j)} = n^{(t-1)/c}$ for some $t \in \{2, ..., c-1\}$.)[10]

Let $\overline{H}' \stackrel{\rm def}{=} (H^{(1)}, ..., H^{(j-1)}, H_\ell^{(j)})$, *while recalling that* $\overline{H} \stackrel{\rm def}{=} (H^{(1)}, ..., H^{(j-1)})$.

3. Sieving the set of choices: *We construct the random set* $R_\ell^{(j)}$ *as follows.*

   - *Each* $r \in R$ *such that* $q_{r,j}^{\alpha, C(x)} \in [n] \setminus \overline{H}'$ (equiv., $q_{r,j}^{\alpha, C(x)} \in (M_\ell^{(j)} \cup L_\ell^{(j)}) \setminus \overline{H}$) *is included in* $R_\ell^{(j)}$ *if and only if* $q_{r,j}^{\alpha, C(x)} \in S^{(j)}$.
   - *Each* $r \in R$ *such that* $q_{r,j}^{\alpha, C(x)} \in \overline{H}'$ (equiv., $q_{r,j}^{\alpha, C(x)} \in H_\ell^{(j)} \cup \overline{H}$) *is included in* $R_\ell^{(j)}$ *with probability* p*, independently of all other choices.*

   (Indeed, the distinction between $M_\ell^{(j)}$ and $L_\ell^{(j)}$ plays no role in the procedure, but it will be used in the analysis.)[11]

4. Branching (for $j \in [q-1]$) and deciding (for $j = q$): *If* $R_\ell^{(j)} = \emptyset$*, then the current iteration returns failure if* $j > 1$ *and the entire decoding fails otherwise* (i.e., if $j = 1$). *Otherwise* (i.e., $R_\ell^{(j)} \neq \emptyset$)*, the decoder tries all possible* $\alpha^{(j)} : H_\ell^{(j)} \to \{0,1\}$ *and treat these trials as follows.*

   Case $j \in [q-1]$**:** *For each* $\alpha^{(j)} : H_\ell^{(j)} \to \{0,1\}$*, we make a recursive call* (to level $j+1$) *with auxiliary inputs* $R_\ell^{(j)}$, $\overline{H}'$*, and* $\alpha' : \overline{H}' \to \{0,1\}$ *such that* $\alpha'(i) = \alpha(i)$ *if* $i \in \overline{H}$ *and* $\alpha'(i) = \alpha^{(j)}(i)$ *otherwise.*[12]

   If some recursive call returns a binary value, then the current iteration returns this value. Otherwise (i.e., all recursive calls failed)*, then the current iteration returns failure if* $j > 1$ *and the entire decoding fails otherwise* (i.e., if $j = 1$).

   Case $j = q$**:** *As in the case of* $j \in [q-1]$*, for each* $\alpha^{(q)} : H_\ell^{(q)} \to \{0,1\}$*, we let* $\alpha' : \overline{H}' \to \{0,1\}$ *such that* $\alpha'(i) = \alpha(i)$ *if* $i \in \overline{H}$ *and* $\alpha'(i) = \alpha^{(q)}(i)$ *otherwise. Now, we try to retrieve the value of* x_ℓ *as follows.*

   - *For each* $r \in R_\ell^{(q)}$*, we consider the value, denoted* $v_r \in \{0, 1, \perp\}$*, that is obtained from the local decoder* D *when it makes queries* $q_{r,1}^{\alpha', C(x)}, ..., q_{r,q}^{\alpha', C(x)}$ *and gets answers according to* $C(x)$ *and* α'*; that is, the* ${j'}^{\rm th}$ *query is answered according to* α' *if it was heavy* (in the relevant j' iteration)[13]*, and is answered according to* $C(x)$ *otherwise* (while relying on the hypothesis in the latter case the query was in $S^{(j')}$).

[10] For reasons that will become clear later, we do not commit to making this choice.

[11] In addition, as stated in Footnote 2, we make the unnatural choice of treating elements of $\overline{H}' \cap S^{(j)}$ as other elements of $\overline{H}'$, while ignoring the fact that we know $C(x)_{S^{(j)}}$. This unnatural choice facilitates the analysis.

[12] Recall that $\overline{H} = (H^{(1)}, ..., H^{(j-1)})$ and $\overline{H}' = (H^{(1)}, ..., H^{(j-1)}, H_\ell^{(j)})$.

[13] Specifically, for $j' \in [q-1]$ it means that the query is in $\overline{H}^{(j')} = (H^{(1)}, ..., H^{(j')})$, whereas for $j' = q$ it means that the query is in $\overline{H}' = \overline{H}^{(q-1)} \cup H_\ell^{(q)}$.

- *If a strict majority of the v_r's equal a bit $b \in \{0,1\}$* (i.e., $|\{r \in R_\ell^{(q)} : v_r = b\}| > |R_\ell^{(q)}|/2$), *then the recursive call returns b. Otherwise, we continue to the next $\alpha^{(q)}$.*

If all attempts to retrieve x_ℓ failed (i.e., for each $\alpha^{(q)} : H_\ell^{(q)} \to \{0,1\}$ there was no strict majority that supports a value in $\{0,1\}$), *then the current iteration returns failure.*

Construction 4.1 describes an (j^{th} level) iteration of the global decoder with parameter $\ell \in [k]$ and auxiliary inputs (i.e., $R, \overline{H}$ and α) obtained from the previous (calling) iteration. Hence, the various objects it constructs (i.e., the 3-way partition $(H_\ell^{(j)}, M_\ell^{(j)}, L_\ell^{(j)})$, the sets $\overline{H}'$ and $R_\ell^{(j)}$, and the augmented assignment $\alpha' : \overline{H}' \to \{0,1\}$) depend on these auxiliary inputs, although this dependence was omitted from the notation. In contrast, we stress again, that *the samples $\overline{S} = (S^{(1)}, ..., S^{(q)})$ are oblivious of these auxiliary inputs as well as of the parameter ℓ*; that is, we shall invoke the first iteration (i.e., $j = 1$), for each possible value of $\ell \in [k]$, but all these invocations will use the same q-sequence of samples $\overline{S}$, and all invocations of iteration $j \in \{2, ..., q\}$ will use the same $S^{(j)}$ (regardless of $R, \overline{H}$ and α).

As stated upfront, a key details that was avoided in Construction 4.1 is the determination of the 3-way partition $(H_\ell^{(j)}, M_\ell^{(j)}, L_\ell^{(j)})$ in Step 2. For starters, note that proceeding analogously to the proof of Theorem 3.3 calls for setting $p = \Theta(\epsilon \log k)^{1/2}$, whereas Claim 2.6 yields $\epsilon = n^{-1/c}$. Furthermore, when aiming at $\Pr[\zeta \in M_\ell^{(j)}] \leq 1/10q$, the proof of Claim 2.6 allows for using any $s_\ell^{(j)} \in \{n^{(t-1)/c} : t \in T\}$ such that T excludes at most $10q$ elements of $\{2, ..., c-1\}$. Intuitively, it seems that we merely need to avoid $10q$ elements per each recursion level $j \in [q]$, which is easy to do when using $c = \Omega(q^2)$.

The difficulty is that it is not clear which elements should be excluded at each level. Specifically, unlike in proof of Theorem 3.3, here we need to deal with extremely many different distributions (arising from the different branches at each recursion level, except for the first level (of $j = 1$)). In particular, for each $j \in [q]$, we wish to consider all possible values of the auxilary inputs at level j, whereas their number may be $j^n \cdot \prod_{j' \in [j-1]} \binom{n}{s_\ell^{(j')}} \cdot 2^{s_\ell^{(j')}}$.

Furthermore, for each $j \in [q]$, we wish to apply a union bound over all $\prod_{j' \in [q]} 2^{|H_\ell^{(j')}|}$ leaves (of the recursion tree), rather than over the $2^{|H_\ell^{(j)}|}$ current branches, let alone that actually the $H_\ell^{(j)}$'s may be different in different invocations of level $j > 1$. Ignoring the latter issue, we stress that, at level j, we actually need to consider $\prod_{j' \in [j]} 2^{|H_\ell^{(j')}|}$ different branches, but for each of them we need to consider all leaves in the corresponding sub-tree of the reursion. In other words, the choice of $S^{(j)}$ should be "good" with respect to all leaves in the recursion sub-tree. We stress that while the union bound is over at least $\prod_{j \in [q]} 2^{s_\ell^{(j)}}$ events, the probability bound in iteration j is only $\exp(-\Omega(n^{-1/c} \cdot s_\ell^{(j)}))$. Thus, we should use approximately equal $s_\ell^{(j)}$'s for all j's.

4.2 The actual implementation

Our solution to the foregoing difficulties is to try all possible $t \in \{2, ..., c-1\}$, and set $s_\ell^{(j)} \leftarrow n^{(t-1)/c}$, for all $j \in [q]$, accordingly. Indeed, a specific value of t may be *inadequate* for the current (branch of an) iteration; that is, it may not hold that $\Pr[\zeta \in M_\ell^{(j)}] \leq 1/10q$, where ζ is as defined in Step 2 and $M_\ell^{(j)} \stackrel{\text{def}}{=} \{z \in [n] : \Pr[\zeta = z] \in (m^{-t/c}, n^{-(t-1)/c}]\}$. But this inadequate choice is detectable, and in such a case we can just abort this iteration while returning failure. The crucial observation is

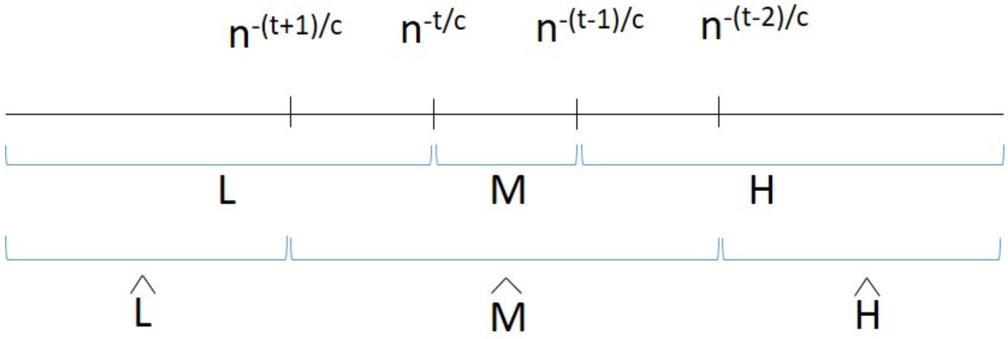

Figure 1: An illustration of the 3-way partition (H, M, L) and the auxiliary set $\widehat{M}$. (Also shown are $\widehat{H} = H \setminus \widehat{M}$ and $\widehat{L} = L \setminus \widehat{M}$.)

that we *need to avoid such an inadequate choice* (of t), made globally and upfront, *only on the* (recursion) *path that corresponds to the correct values of the* $\alpha^{(j)}$*'s* (i.e., $\alpha^{(j)}(H_\ell^{(j)}) = C(x)_{H_\ell^{(j)}}$. Intuitively, since each $j \in [q]$ rules out at most $10q$ choices of t (for the single correct path), using $c \geq 10q^2 + 3$ will do. Hence, invoking the (first iteration of the) global decoder (i.e., $j = 1$) at most $c - 2$ times, each time with a different value of $t \in \{2, ..., c - 1\}$, and using the bit value provided by the first non-failing invocation will yield x_ℓ (as desired).

4.2.1 Preliminaries

Unfortunately, implementing the foregoing idea is not straightforward. The main problem is that the path that corresponds to the correct values of the $\alpha^{(j)}$'s is not well-defined, because this path is actually a random variable (since it depends on the random choice of $S^{(1)}, ..., S^{(q)}$, which in turn determine $M_\ell^{(1)}, ..., M_\ell^{(q)}$). Furthermore, for $j > 1$, the total probability mass assigned to $M_\ell^{(j)}$ (along this path) is a random variable. Intuitively, this should not affect the question of which t's are adequate too much, but the issue is making this intuition precise.

Of special concern is the case that a large amount of the probability mass (of the j^{th} query, for several j's) resides on integer powers of $n^{-1/c}$. In such a case, this mass may appear as larger than a specific power in one execution, but smaller in another. There are several ways of coping with this type of problem. In the current context, it makes sense to rule out as inadequate values of t for which the foregoing situation occurs. This calls for increasing c by a factor of three, and using t only if the triple $(t - 1, t, t + 1)$ contains no inadequate value. Towards this goal, we prove the following, where in first reading one may consider the setting $c' \approx c/3$ and $c'' = 1$.

Claim 4.2 (a variation on Claim 2.6 (see illustration in Figure 1)): *Let ζ be a random variable that is distributed over $[n]$, and $c, c', c'' \in \mathbb{N}$ such that $c > c'' + 3$. Suppose that t is selected uniformly in $\{c''+1, ..., c-2\}$. Then, with probability at least $1 - \frac{3 \cdot c' - 1}{c - c'' - 2}$ there exist a 3-way partition, (H, M, L), of $[n]$ such that*
`(i)` *$H = \{z \in [n] : \Pr[\zeta = z] > n^{-(t-1)/c}\}$, which implies $|H| < n^{(t-1)/c}$.*
`(ii)` *$\Pr[\zeta \in \widehat{M}] \leq 1/c'$, where $\widehat{M} \overset{\text{def}}{=} \{z \in [n] : \Pr[\zeta = z] \in (n^{-(t+1)/c}, n^{-(t-2)/c}]\} \supseteq M$, and*
`(iii)` *$L = \{z \in [n] : \Pr[\zeta = z] \leq n^{-t/c}\}$.*

(Indeed, $M = \{z \in [n] : \Pr[\zeta = z] \in (n^{-t/c}, n^{-(t-1)/c}]\}$.)

Note that a version of Claim 2.6 follows as a special case by letting $c' = (c-3)/3$ and $c'' = 1$ (and using $n^{-t/c} \le n^{-1/c}/(|H|+1)$).[14] On the other hand, for our application we shall use $c' = 10q$ and $c = 3 \cdot (c'+1) \cdot q + c'' = \Theta(q^2) = \Theta(c')^2$. Observing that

$$q \cdot \frac{3 \cdot c' - 1}{c - c'' - 2} \;=\; q \cdot \frac{3 \cdot c' - 1}{3 \cdot (c'+1) \cdot q - 2} \;<\; \frac{c'}{c' + 1 - (1/q)}$$

it follows that there exists a value $t \in \{c''+1, ..., c-2\}$ that fits q different random variables (which correspond to the q iterations of the global decoder). Note that in this case it holds that

$$\epsilon \stackrel{\rm def}{=} |H| \cdot \max_{z \in L} \{\Pr[\zeta = z]\} \;<\; n^{(t-1)/c} \cdot n^{-t/c} \;=\; n^{-1/c} \tag{5}$$

Recall that, in our application (which intends to follow the structure of the proof of Theorem 3.3), we shall use $p = \Omega(\epsilon \log k)^{1/2}$, and so we seek to have ϵ as small as possible. The reason for using a large c'' in our application will become clear later (i.e, in the proof of Claim 4.4.2).[15]

Proof: Defining the $B_{j'}$'s exactly as in the proof of Claim 2.6 (i.e., essentially, $z \in B_{j'}$ if and only if $\Pr[\zeta = z] \in (n^{-j'/c}, n^{-(j'-1)/c}]$), we observe that the number of t's that violate

$$\Pr\left[\zeta \in \bigcup_{j' \in \{t-1, t, t+1\}} B_{j'}\right] \le 1/c' \tag{6}$$

is at most $3 \cdot c' - 1$. Hence, with probability at least $1 - \frac{3c'-1}{c-c''-2}$ over the choice of $t \in \{c''+1, ..., c-2\}$, Eq. (6) holds. Letting $H \stackrel{\rm def}{=} \bigcup_{j' \le t-1} B_{j'}$ and $L \stackrel{\rm def}{=} \bigcup_{j' \ge t+1} B_{j'}$, and using $\widehat{M} = \bigcup_{j' \in \{t-1,t,t+1\}} B_{j'}$, the claim follows. ■

An ideally adequate pivot. As hinted upfront, the threshold value t, hereafter called the pivot, will be selected in a hope of fitting the 3-way partitions determined along the recursion path that corresponds to the correct $\alpha^{(j)}$'s. A value that is likely to do so is one for which Claim 4.2 holds when considering the q distributions that correspond to the distribution of the q queries in execution of the local decoder itself, on fixed input $\ell \in [k]$ and $C(x) \in \{0,1\}^n$. Before proceeding, let us highlight the choice of the constant c that we shall use from this point on. We use

$$c = 3 \cdot (10q + 2) \cdot q \tag{7}$$

which corresponds to $c' = 10q$ and $c'' = 3 \cdot q$.

Definition 4.3 (ideally adequate pivot): *Fixing $(\ell, x) \in [k] \times \{0,1\}^k$ and $t \in \{3q+1, ..., c-2\}$, for each $j \in [q]$, we denote by $\underline{\zeta}^{(j)}$ the distribution of the $j^{\rm th}$ query of the local decoder, on input ℓ when given oracle access to $C(x)$. We say that the pivot t is* ideally adequate *if for every $j \in [q]$ it holds that $\Pr[\underline{\zeta}^{(j)} \in \underline{\widehat{M}}^{(j)}] \le 1/10q$, where $\underline{\widehat{M}}^{(j)} \stackrel{\rm def}{=} \{z \in [n] : \Pr[\underline{\zeta}^{(j)} = z] \in (n^{-(t+1)/c}, n^{-(t-2)/c}]\}$.*

[14] In this case, we get $\Pr[\zeta \in M] \le 3/(c-3)$ instead of $\Pr[\zeta \in M] \le 1/(c-2)$. Actually, we can get $\Pr[\zeta \in M] \le 1/(c-3)$ by using $M = \{z \in [n] : \Pr[\zeta = z] \in (n^{-t'/c}, n^{-(t'-1)/c}]\}$ for some $t' \in \{t-1, t, t+1\}$, and redefine H and L accordingly.

[15] We hint that $t \ge c''+1$ implies that each query in the corresponding set M is supported by at most $n^{-(t-1)} \cdot n' \le n^{-c''/c} \cdot n'$ random choices in $[n']$. The latter fact will be used in the proof of Claim 4.4.2.

By Claim 4.2, there exists a value $t \in \{3q+1, ..., c-2\}$ that is ideally adequate. We shall prove that, with very high probability, an ideally adequate pivot does fit the 3-way partitions determined along the recursion path that corresponds to the correct $\alpha^{(j)}$'s. Unfortunately, we are able to prove this result only for $p = \Omega(n^{-1/cq})$ (rather than for any p or just for $p = \Omega(n^{-1/c})$).

Note that all objects that directly refer to the foregoing ideal experiment with the original local decoder appear with an underline (e.g., see $\underline{\zeta}^{(j)}$ and $\widehat{\underline{M}}^{(j)}$ in Definition 4.3). In contrast, notations that directly refer to the global decoder (i.e., Construction 4.1 and its variants) appear without an underline (e.g., see $\zeta^{(j)}$ and $(H^{(j)}, M^{(j)}, L^{(j)})$ next).

Lemma 4.4 (ideally adequate pivots are adequate for a sieving process akin to Construction 4.1): *For fixed $(\ell, x) \in [k] \times \{0,1\}^k$ and $t \in \{3q+1, ..., c-2\}$, starting with $R^{(0)} = [n']$, consider the following iterative* (sieving) *process that is akin to Construction 4.1. For $j = 1, ..., q$, the $j^{\rm th}$ iteration is as follows.*

- *Defining $\zeta^{(j)}$* (based on $R^{(j-1)}$ and $\alpha : (\bigcup_{j' \in [j-1]} H^{(j')}) \to \{0,1\}$) *as in Step 2, we obtain a 3-way partition $(H^{(j)}, M^{(j)}, L^{(j)})$ such that $M^{(j)} \stackrel{\rm def}{=} \{z \in [n] : \Pr[\zeta^{(j)} = z] \in (n^{-t/c}, n^{-(t-1)/c}]\}$, and $H^{(j)}$* (resp., $L^{(j)}$) *contains the heavier* (resp., lighter) *elements of $[n]$.*
- *Next, $R^{(j)}$ is selected at random* (based on $(H^{(j)}, M^{(j)}, L^{(j)})$) *as in Step 3, and α is extended to $H^{(j)}$ such that $\alpha(H^{(j)}) = C(x)_{H^{(j)}}$. Specifically, $R^{(j)}$ is constructed as follows.*[16]
    - *Each $i \in [n] \setminus (\bigcup_{j' \in [j]} H^{(j')})$ is selected* (i.e., placed in a set S) *with probability p, independently of all other choices, and all r's in $R^{(j-1)}$ such that $q_{r,j}^{\alpha, C(x)}$ was selected* (i.e., all elements of $\{r \in R^{(j-1)} : q_{r,j}^{\alpha, C(x)} \in S\}$) *are included in $R_\ell^{(j)}$.*
    - *Each $r \in R^{(j-1)}$ such that $q_{r,j}^{\alpha, C(x)} \in \overline{H}'$* (equiv., $q_{r,j}^{\alpha, C(x)} \in H_\ell^{(j)} \cup \overline{H}$) *is included in $R_\ell^{(j)}$ with probability p, independently of all other choices.*

Suppose that t is ideally adequate and $p > n^{-1/cq}$. Then, for each $j \in [q]$, with probability $1 - \exp(-\Omega(n^{q/c}))$ *over the choice of $R^{(1)}, ..., R^{(j-1)}$, it holds that* $\Pr[\zeta^{(j)} \in M^{(j)}] \leq 0.12/q$.

We stress that (unlike $\underline{\zeta}^{(j)}$ of Definition 4.3) only the random variable $\zeta^{(1)}$ represents a fixed distribution. In contrast, for $j > 1$, the notation $\zeta^{(j)}$ refers to a distribution that may vary based on the random choice of $R^{(j-1)}$, which in turn is based on $\zeta^{(j-1)}$. Hence, it would have been prudent to denote it by $\zeta^{(j)}(R^{(1)}, ..., R^{(j-1)})$, which we avoid for sake of friendliness. Likewise, unlike $\widehat{\underline{M}}^{(j)}$, for $j > 1$, the notation $M^{(j)}$ represents a distribution on sets that is determined by $\zeta^{(j-1)}$, and would have more prudently denoted $M^{(j)}(R^{(1)}, ..., R^{(j-1)})$.

Unfortunately, the proof of Lemma 4.4 is quite long and complex; the reader may consider skipping this proof and proceeding directly to Section 4.2.2, where a suitable revision of Construction 4.1 is analyzed. Of course, Lemma 4.4 is crucial to that analysis, but seeing the analysis first may provide motivation for reading the proof of Lemma 4.4.

Proof: The claim holds trivially for $j = 1$, because $\zeta^{(1)} \equiv \underline{\zeta}^{(1)}$ and $M^{(1)} = \underline{M}^{(1)}$, where $\underline{M}^{(j)} \stackrel{\rm def}{=} \{z \in [n] : \Pr[\underline{\zeta}^{(j)} = z] \in (n^{-t/c}, n^{-(t-1)/c}]\} \subseteq \widehat{\underline{M}}^{(j)}$ for every $j \in [q]$, whereas $\Pr[\underline{\zeta}^{(1)} \in \widehat{\underline{M}}^{(1)}] \leq 1/10q$

[16] Recall that $q_{r,j}^{\alpha, C(x)}$ denotes the $j^{\rm th}$ query made by the local decoder, on input ℓ and randomness r, when prior queries are answered according to $\alpha : (\bigcup_{j' \in [j-1]} H^{(j')}) \to \{0,1\}$ and $C(x)_{[n] \setminus \bigcup_{j' \in [j-1]} H^{(j')}}$.

(by the hypothesis that t is ideally adequate). However, when $j > 1$, neither of these equalities is guaranteed to hold; in particular, three things may go wrong.

1. The set $M^{(j)}$ may extend beyond $\underline{M}^{(j)}$ due to elements of $\widehat{\underline{M}}^{(j)} \setminus \underline{M}^{(j)}$ moving to $M^{(j)}$.

   Recall that $M^{(j)}$ depends on a random sieving process that may change the weight of possible queries based on the (random) choice of $R^{(j-1)}$. Indeed, an element $r \in \widehat{\underline{M}}^{(j)} \setminus \underline{M}^{(j)}$ having weight that is close to one of the thresholds of $\underline{M}^{(j)}$, is likely to move to $M^{(j)}$, and so increase the total weight of $M^{(j)}$.

   This is exactly the reason that we upper-bounded the total weight of $\widehat{\underline{M}}^{(j)}$ (rather than the weight of $\underline{M}^{(j)}$). Hence, this case is covered already.

2. The set $M^{(j)}$ may extend beyond $\underline{M}^{(j)}$ due to elements of $[n] \setminus \widehat{\underline{M}}^{(j)}$ moving to $M^{(j)}$.

   Intuitively, such a move is unlikely because it requires a significant change in the relative weight of individual elements. Specifically, each element of $[n] \setminus \widehat{\underline{M}}^{(j)}$ that is heavier (resp., lighter) than a generic element of $\widehat{\underline{M}}^{(j)}$ is at least $n^{1/c}$ times heavier (resp., lighter) than a generic element of $\underline{M}^{(j)}$.

   This case is most difficulty to handle. Actually, the case of the heavier elements is easy, because they are relatively few in number, and so even if they all move to $M^{(j)}$ then this will not increase its total weight by much. The real problem is with the lighter elements, and we resolve it by relying on the hypothesis that $p^q > n^{-1/c}$ (equiv., $p > n^{-1/cq}$). This is the only place in the proof that relies on this hypothesis, which in turn yields an inferior lower bound for the adaptive case. (Recall that $c = \Theta(q^2)$, and so $p > n^{-\Theta(1/q^3)}$.)

3. The total weight of $\widehat{\underline{M}}^{(j)}$ may increase when moving from $\underline{\zeta}^{(j)}$ to $\zeta^{(j)}$.

   Intuitively, a significant change is unlikely because $R^{(j-1)}$ provides a good sample of $[n']$. Note that we only need to establish that $R^{(j-1)}$ samples well few sets (i.e., the $\widehat{\underline{M}}^{(j)}$'s), and the difference between $L^{(j-1)}$ and $M^{(j-1)}$ does not matter here.

Concretely, for every $R^{(1)}, ..., R^{(j-1)}$, it holds that

$$\Pr\left[\zeta^{(j)} \in M^{(j)}\right] \leq \Pr\left[\zeta^{(j)} \in \widehat{\underline{M}}^{(j)}\right] + \Pr\left[\zeta^{(j)} \in \left(M^{(j)} \setminus \widehat{\underline{M}}^{(j)}\right)\right] \tag{8}$$

where this inequality materializes the treatment of the first phenomenon. Letting[17]

$$\begin{aligned} \underline{\widehat{H}}^{(j)} &\overset{\rm def}{=} \{z \in [n] : \Pr[\underline{\zeta}^{(j)} = z] > n^{-(t-2)/c}\} \\ \underline{\widehat{L}}^{(j)} &\overset{\rm def}{=} \{z \in [n] : \Pr[\underline{\zeta}^{(j)} = z] \leq n^{-(t+1)/c}\} \end{aligned}$$

(see illustration in Figure 1), and using $M^{(j)} \setminus \widehat{\underline{M}}^{(j)} = (M^{(j)} \cap \underline{\widehat{H}}^{(j)}) \cup (M^{(j)} \cap \underline{\widehat{L}}^{(j)})$, we rewrite the r.h.s. of Eq. (8) as

[17] We used the notation $\underline{\widehat{H}}^{(j)}$ and $\underline{\widehat{L}}^{(j)}$, because it is natural to define $\underline{H}^{(j)} = \{z \in [n] : \Pr[\underline{\zeta}^{(j)} = z] > n^{-(t-1)/c}\}$ and $\underline{L}^{(j)} = \{z \in [n] : \Pr[\underline{\zeta}^{(j)} = z] \leq n^{-t/c}\}$. In that case, it holds that $\underline{\widehat{H}}^{(j)} = \underline{H}^{(j)} \setminus \widehat{\underline{M}}^{(j)}$ and $\underline{\widehat{L}}^{(j)} = \underline{L}^{(j)} \setminus \widehat{\underline{M}}^{(j)}$.

$$\Pr\left[\zeta^{(j)}\in\widehat{\underline{M}}^{(j)}\right]+\Pr\left[\zeta^{(j)}\in\left(M^{(j)}\cap\widehat{\underline{H}}^{(j)}\right)\right]+\Pr\left[\zeta^{(j)}\in\left(M^{(j)}\cap\widehat{\underline{L}}^{(j)}\right)\right] \tag{9}$$

We first observe that, for every $R^{(1)},....,R^{(j-1)}$, it holds that

$$\begin{aligned}\Pr\left[\zeta^{(j)}\in\left(M^{(j)}\cap\widehat{\underline{H}}^{(j)}\right)\right] &= \sum_{i\in(M^{(j)}\cap\widehat{\underline{H}}^{(j)})}\Pr[\zeta^{(j)}=i] \\ &\leq |\widehat{\underline{H}}^{(j)}|\cdot\max_{i\in M^{(j)}}\left\{\Pr[\zeta^{(j)}=i]\right\} \\ &< n^{(t-2)/c}\cdot n^{-(t-1)/c} \\ &= n^{-1/c},\end{aligned}$$

where the second inequality is due to the definitions of $\widehat{\underline{H}}^{(j)}$ and $M^{(j)}$. Combining this with Eq. (8) and Eq. (9), we infer that, for every $R^{(1)},....,R^{(j-1)}$, it holds that

$$\Pr[\zeta^{(j)}\in M^{(j)}] \;\leq\; n^{-1/c}+\Pr\left[\zeta^{(j)}\in\widehat{\underline{M}}^{(j)}\right]+\Pr\left[\zeta^{(j)}\in\left(M^{(j)}\cap\widehat{\underline{L}}^{(j)}\right)\right] \tag{10}$$

We stress that Eq. (10) holds for every fixed sequence $R^{(1)},....,R^{(j-1)}$. In contrast, the next upper-bound holds only under certain conditions regarding the sequence $R^{(1)},....,R^{(j-1)}$, whereas we shall later show (in Claim 4.4.2) that these conditions hold with overwhelmingly high probability.

Claim 4.4.1 (conditional upper bound on Eq. (10)): *For every $j\in[q]$, suppose that the sequence $R^{(1)},....,R^{(j-1)}$ satisfies the following two conditions:*

1. $|R^{(j-1)}|>n^{-1/c}\cdot n'$;
2. *Letting* $B^{(j)}\stackrel{\rm def}{=}\left\{r\in[n']:q_{r,j}^{\alpha,C(x)}\in\widehat{\underline{M}}^{(j)}\right\}$, *it holds that*

$$\frac{|R^{(j-1)}\cap B^{(j)}|}{|R^{(j-1)}|}\;\leq\;\frac{|B^{(j)}|}{n'}+\frac{0.01}{q}$$

which means that $R^{(j-1)}$ does not over-represents the set $B^{(j)}$.

Then, $\Pr[\zeta^{(j)}\in M^{(j)}]\leq 0.12/q$.

Recalling that $R^{(0)}=[n]$ and $\mathrm{E}[|R^{(j-1))}|]=p^{j-1}\cdot|R^{(0)}|$, and relying on $p^{j-1}\gg n^{-1/q}$, it is reasonable to believe that Condition 1 holds for a random sequence (i.e., $\Pr_{R^{(1)},...,R^{(j-1)}}[|R^{(j-1)}|\leq n^{-1/c}\cdot n']=o(1/k)$). Indeed, this as well as an analogous statement regarding Condition 2 follow from a stronger concentration bound (regarding $|R^{(j-1)}\cdot T|$ for any fixed set T) that is proved in Claim 4.4.2. At this point, let us stress that $p^{j-1}\gg n^{-1/q}$ relies on the hypothesis $p>n^{-1/cq}$ and that this is the only place in our proof where the latter hypothesis is used.

Proof: Our starting point is Eq. (10). We first observe that under Condition 1, it holds that $M^{(j)}\cap\widehat{\underline{L}}^{(j)}$ is empty. This is the case because $i\in\widehat{\underline{L}}^{(j)}$ means that there are at most $n^{-(t+1)/c}\cdot n'$ choices that lead the local decoder to make i its $j^{\rm th}$ query, whereas $i\in M^{(j)}$ means that there must be more than $n^{-t/c} \cdot |R^{(j-1)}|$ such choices in $R^{(j-1)} \subseteq [n']$. Hence, $i \in M^{(j)} \cap \widehat{\underline{L}}^{(j)}$ implies $n^{-(t+1)/c} \cdot n' > n^{-t/c} \cdot |R^{(j-1)}|$, which implies $|R^{(j-1)}| < n^{-1/c} \cdot n'$, in contradiction to Condition 1. Consequently, we have

$$\Pr\left[\zeta^{(j)} \in \left(M^{(j)} \cap \widehat{\underline{L}}^{(j)}\right)\right] = 0. \tag{11}$$

Turning to $\Pr[\zeta^{(j)} \in \widehat{\underline{M}}^{(j)}]$, observe that it equals the fraction of $r \in R^{(j-1)}$ that leads the global decoder of Construction 4.1 to make a j^{th} query that resides in $\widehat{\underline{M}}^{(j)}$ (in the branch of the recursion that agrees with $C(x)$); that is,

$$\begin{aligned}\Pr\left[\zeta^{(j)} \in \widehat{\underline{M}}^{(j)}\right] &= \frac{|R^{(j-1)} \cap B^{(j)}|}{|R^{(j-1)}|} \\ &\leq \frac{|B^{(j)}|}{n'} + \frac{0.01}{q}\end{aligned}$$

where the inequality uses Condition 2. On the other hand, $\frac{|B^{(j)}|}{n'}$ equals $\Pr[\underline{\zeta}^{(j)} \in \widehat{\underline{M}}^{(j)}]$, which is upper-bounded by $1/10q$ (per the lemma's hypothesis). Combining $\Pr[\zeta^{(j)} \in \widehat{\underline{M}}^{(j)}] \leq 0.11/q$ with Eq. (11) and Eq. (10), while using $n^{-1/c} < 0.01/q$, the claim follows. ■

Claim 4.4.2 (on the sampling features of $R^{(j)}$): *For every $j \in [q]$ and every set $T \subseteq [n']$, it holds that*

$$\Pr\left[|T \cap R^{(j)}| = p^j \cdot |T| \pm \beta \cdot p^j \cdot n'\right] \;=\; 1 - \exp(-\Omega(\beta^2 \cdot p^{2j} \cdot n^{3q/c}))$$

where the probability is taken over the random process of selecting $R^{(1)}, ..., R^{(j)}$.

Using $p^{2q} \cdot n^{3q/c} \geq n^{q/c}$ (equiv., $p \geq n^{-1/c}$), we get a probability bound of $1 - \exp(-\Omega(\beta^2 \cdot n^{q/c}))$.

Proof: The claim is proved by induction on j, where the based case of $j = 0$ is trivial, and in the (j^{th}) induction step we shall rely on $|R^{(j-1)}| = (1 \pm 0.1) \cdot p^{j-1} \cdot n'$, which follows as a special case (with $T = [n']$ and $\beta = 0.1$). Specifically, the induction claim for step j is

$$\Pr\left[|T \cap R^{(j)}| = p^j \cdot |T| \pm \frac{j\beta}{q} \cdot p^j \cdot n'\right] \;=\; 1 - \exp(-\Omega(\beta^2 \cdot p^{2j} \cdot n^{3q/c})). \tag{12}$$

Towards applying Lemma 3.1, we consider a graph on the vertex set $R^{(j-1)}$ such that r and s are connected if and only if they lead to the same j^{th} query (i.e., $q_{r,j}^{\alpha,C(x)} = q_{s,j}^{\alpha,C(x)}$) and this query is not in $H^{(j)}$. Indeed, here we use the fact that each r that yields a heavy j^{th} query is included in $R^{(j)}$ at random *independently of any other choice.* The key observation is that the vertices in $M^{(j)} \cup L^{(j)}$, which are the only non-isolated vertices, have degree at most $n^{-(t-1)/c} \cdot n'$. Using $t \geq 3q+1$ (which follows by the setting of $c'' = 3q$)[18], it follows that this graph has at least $0.9 \cdot p^{j-1} \cdot n'$ vertices (i.e.,

[18] In contrast, a setting of $c'' = 1$, would have yielded maximal degree at most $n^{-1/c} \cdot n'$, which would have yielded a probability bound of $1 - \exp(-\Omega(\beta/q)^2 \cdot p^{j+1} \cdot n^{1/c})$. In such a case, a meaningful result would have required $p^{q+1} \cdot n^{1/c} = \omega(1)$ (equiv., $p = \omega(n^{-1/(q+1)c})$).

$|R^{(j-1)}| \geq 0.9 \cdot p^{j-1} \cdot n'$) and maximal degree at most $n^{-3q/c} \cdot n'$. Defining random variables that represents whether $r \in T$ and applying Lemma 3.1, we get

$$\begin{aligned}
\sigma_j &\stackrel{\rm def}{=} \Pr\left[|T \cap R^{(j)}| = p \cdot |T \cap R^{(j-1)}| \pm (\beta/q) \cdot p^j \cdot n'\right] \\
&\geq \Pr\left[|T \cap R^{(j)}| = p \cdot |T \cap R^{(j-1)}| \pm (\beta/q) \cdot p \cdot |R^{(j-1)}|/1.1\right] \\
&= 1 - \exp\left(-\Omega(\beta/q)^2 \cdot p^2 \cdot \frac{|R^{(j-1)}|}{n^{-3q/c} \cdot n'}\right) \\
&\geq 1 - \exp(-\Omega(\beta/q)^2 \cdot p^{j+1} \cdot n^{3q/c}),
\end{aligned}$$

where the first inequality uses $|R^{(j-1)}| \leq 1.1 \cdot p^{j-1} \cdot n'$ and the last inequality uses $|R^{(j-1)}| \geq 0.9 \cdot p^{j-1} \cdot n'$. Combining the bound for σ_j with the induction hypothesis (cf. Eq. (12)), we get

$$\begin{aligned}
&\Pr\left[|T \cap R^{(j)}| \neq \left(p^j \cdot |T| \pm \frac{j\beta}{q} \cdot p^j \cdot n'\right)\right] \\
&\leq \Pr\left[|T \cap R^{(j-1)}| \neq \left(p^{j-1} \cdot |T| \pm \frac{(j-1)\beta \cdot p}{q} \cdot p^{j-1} \cdot n'\right)\right] + (1 - \sigma_j) \\
&= \exp(-\Omega((\beta \cdot p)^2 \cdot p^{2(j-1)} \cdot n^{3q/c})) + \exp(-\Omega(\beta^2 \cdot p^{j+1} \cdot n^{3q/c}))
\end{aligned}$$

and the induction claim follows. ■

The conditions of Claim 4.4.1 hold w.v.h.p. over the random selection of $R^{(1)}, ..., R^{(j)}$. Using Claim 4.4.2 we show that the two conditions of Claim 4.4.1 hold, with overwhelmingly high probability, over the random process of generating $R^{(1)}, ..., R^{(j-1)}$. Starting with Condition 1, we apply Claim 4.4.2 with $T = [n']$ (and $\beta = 1$), while recalling that $p > n^{-1/cq}$ (equiv., $p^q > n^{-1/c}$). Hence, for every $j \geq 2$,

$$\begin{aligned}
\Pr\left[|R^{(j-1)}| \leq n^{-1/c} \cdot n'\right] &< \Pr\left[|R^{(j-1)}| \leq 2 \cdot p^{j-1} \cdot n'\right] \\
&= \exp(-\Omega(n^{q/c})).
\end{aligned}$$

Turning to Condition 2 (of Claim 4.4.1), using Claim 4.4.2 with $T = B^{(j)}$ (as well as with $T = [n']$) and $\beta = 0.001/q$,

$$\begin{aligned}
&\Pr\left[\frac{|R^{(j-1)} \cap B^{(j)}|}{|R^{(j-1)}|} > \frac{|B^{(j)}|}{n'} + \frac{0.01}{q}\right] \\
&= \Pr\left[|B^{(j)} \cap R^{(j-1)}| > p^{j-1} \cdot |B^{(j)}| + (0.001/q) \cdot p^{j-1} \cdot n'\right] \\
&\quad + \Pr\left[|R^{(j-1)}| < p^{j-1} \cdot n' - (0.001/q) \cdot p^{j-1} \cdot n'\right] \\
&= 2 \cdot \exp(-\Omega(n^{q/c})).
\end{aligned}$$

The lemma follows. ■

A useful corollary of Claim 4.4.2. In addition to Lemma 4.4, we shall also use the following result, which is a direct corollary of Claim 4.4.2.

Corollary 4.5 (concentration bound for the size of $R^{(j)}$): *For fixed $\ell \in [k]$ and $x \in \{0,1\}^k$, let $R^{(1)}, ..., R^{(q)}$ be generated as in Lemma 4.4. Then, for every $j \in [q]$, it holds that*

$$\Pr\left[|R^{(j)}| = (1 \pm \beta) \cdot p^j \cdot n'\right] \ = \ 1 - \exp(-\Omega(\beta^2 \cdot n^{q/c})),$$

provided that $p \geq n^{-1/c}$.

Hence, for $\beta < 0.01/q$, with probability $1 - \exp(-\Omega(\beta^2 \cdot n^{q/c}))$, it holds that $|R^{(j)}| = (1 \pm 3q \cdot \beta) \cdot p \cdot |R^{(j-1)}|$. (Actually, such a bound is established directly in the proof of Claim 4.4.2.)

4.2.2 The revised (pivot-based) global decoder

Claim 4.2 and Lemma 4.4 establish the viability of the idea of using an ideally adequate pivot in a modification of Construction 4.1. Specifically, Claim 4.2 implies that such a pivot exists (for any $\ell \in [k]$ and $x \in \{0,1\}^k$), whereas Lemma 4.4 asserts that this pivot will we adequate also for the actual execution (of the global decoder) *along the recursion path that fits $C(x)$*. Hence, we now turn to specify this modification of Construction 4.1, which consists of checking whether the (ideally adequate) pivot fits the actual distribution of the j^{th} query. That is, rather than finding a 3-way partition that fits this distribution (as done in Step 2 of Construction 4.1), we check whether the 3-way partition determined by the fixed pivot t fits this distribution.

Construction 4.6 (revised Step 2 of Construction 4.1): *For a fixed parameter $t \in \{3q+1, ..., c-2\}$, we proceed as follows, where ζ is defined as in the original step.*[19] *Let $(H_\ell^{(j)}, M_\ell^{(j)}, L_\ell^{(j)})$ be a 3-partition of $[n]$ such that $H_\ell^{(j)} = \{z \in [n] : \Pr[\zeta = z] > n^{-(t-1)/c}\}$ and $L_\ell^{(j)} = \{z \in [n] : \Pr[\zeta = z] \leq n^{-t/c}\}$. If $\Pr[\zeta \in M_\ell^{(j)}] \leq 0.12/q$ holds, then we proceed* (with this 3-way partition) *as before. Otherwise, we return failure if $j > 1$ and the entire decoding fails otherwise* (i.e., if $j = 1$)

Indeed, in terms of the original step, this corresponds to insisting on $s_\ell^{(j)} = n^{(t-1)/c}$ and $\epsilon = n^{-1/c}$, where $c = 3 \cdot (10q+2) \cdot q$. Actually, we shall also slightly modify Step 3 so that it halts in the rare case that $|R^{(j)}| \not\approx p^j \cdot n'$. Specifically, we shall use the following.

Construction 4.7 (revised Step 3 of Construction 4.1): *After generating $R^{(j)}$ as in the original step, we proceed only if $|R^{(j)}| = (1 \pm (0.001/q)) \cdot p \cdot |R|$. Otherwise, we return failure if $j > 1$ and the entire decoding fails otherwise* (i.e., if $j = 1$)

Recall that Corollary 4.5 implies that the foregoing failure will occur with negligible probability *along the recursion path* (of Construction 4.1) *that fits $C(x)$*. Using the revised global decoder, we prove the following.

[19] Recall that ζ represents the distribution of $q_{r,j}^{\alpha,C(x)}$ when r is uniformly distributed in R, where $q_{r,j}^{\alpha,C(x)}$ denotes the j^{th} query made by the local decoder, on input ℓ and randomness r, when prior queries are answered according to $\alpha : (\bigcup_{j' \in [j-1]} H^{(j')}) \to \{0,1\}$ and $C(x)_{[n] \setminus \bigcup_{j' \in [j-1]} H^{(j')}}$.

Theorem 4.8 (lower bound for general local decoders): *Suppose that $C : \{0,1\}^k \to \{0,1\}^n$ is a general q-query relaxed LDC. Then, $n > k^{1+\frac{1}{30\cdot(q+1)^3}}$.*

Proof: Here we use $p = n^{-1/cq} \cdot \log k < n^{-1/30(q+1)^3}$. For any fixed $\ell \in [k]$ and $x \in \{0,1\}^k$, we shall prove that invoking the first iteration (i.e., $j = 1$) of the *revised* Construction 4.1, on input $C(x)$, with parameters p and ℓ and the pivot t, yields x_ℓ with probability $1 - o(1/k)$ if t is ideally adequate, whereas a wrong value is output with probability $o(1/k)$ for any value of t. Indeed, in the other cases (i.e., when not outputing a bit value), the global decoder announces failure. By invoking this global decoder on each $t \in \{3q+1, ..., c-2\}$, we obtain a global decoder that, on any input $\ell \in [k]$ and $C(x) \in \{0,1\}^n$, outputs x_ℓ probability at least $1 - o(1/k)$. As in the proof of Theorem 3.3, using the latter global decoder, we can recover each x, with probability $1-o(1)$, when given a sample of density approximately $q \cdot p$ of the bits of $C(x)$, and it follows that $n = \Omega(k/qp)$. Observing that $p = o(n^{-1/30(q+1)^3})$, we obtain $n = \Omega(n/p) > k^{1+\frac{1}{30(q+1)^3}}$.

We stress that, in the foregoing as well as hereafter, the term revised Construction 4.1 means *the revision of Construction 4.1 obtained by replacing the original Steps 2 and 3 by Constructions 4.6 and 4.7, respectively.* Recall that the first revision introduces another parameter, denoted t, which is initially selected in $\{3q+1, ..., c-2\}$, where $c = 3 \cdot (10q+2) \cdot q$.

Fixing any x, ℓ and t, our analysis of the revised Construction 4.1 follows the strategy of the proof of Theorem 3.3, but is considerably more complex. In particular, our proof proceeds in q iterations that correspond to the q iterations of (the revised) Construction 4.1. Recalling that in each iteration (of this construction) the set of surviving choices is "sieved" (by a factor of p) and the assignment to heavy queries is extended, we mimic the same effect on the local decoder. This is done by introducing a notion of a *residual local decoder*, which is parameterized by $(j, R, \overline{H}, \alpha)$ that fit the parameter $j \in [q]$ and the auxiliary inputs $R \subseteq [n']$, $\overline{H} \subseteq [n]$, and $\alpha : \overline{H} \to \{0,1\}$ of Construction 4.1. For such $(j, R, \overline{H}, \alpha)$, we shall also consider the residual local decoders that are parameterized by $(j+1, R_\ell^{(j)}, \overline{H}', \alpha')$, where $R_\ell^{(j)}, \overline{H}'$ and $\alpha' : \overline{H}' \to \{0,1\}$ are derived as in Step 4 of Construction 4.1. Note that, for $j \in [q-1]$, the latter tuples appear in the corresponding recursive invocation of Construction 4.1 (of level $j+1$). Furthermore:

1. The residual local decoder that is parameterized by $(1, [n'], \emptyset, \lambda)$ will coincide with the original local decoder, denoted D.

2. The residual local decoders parameterized by $(q+1, R_\ell^{(q)}, \overline{H}', \alpha')$ will coincide with the evaluation trials made at the leaves in the recursion tree (defined by Construction 4.1). Indeed, while in case $j = q$ of Step 4 we tried all possible $r \in R_\ell^{(q)}$, the corresponding residual local decoder will try a single $r \in R_\ell^{(q)}$ selected uniformly at random.

3. We shall relate residual local decoders parameterized by $(j, R, \overline{H}, \alpha)$ to the residual decoders parameterized by $(j+1, R_\ell^{(j)}, \overline{H}', \alpha')$, where the latter parameters are derived as in Step 4 of Construction 4.1. This relation will be established analogously to the proof of Theorem 3.3.

With the foregoing motivation in mind, we define residual local decoders as follows.

Definition 4.8.1 (residual local decoders): *Fixing a q-query local decoder D for C, for $j \in [q+1]$, $R \subseteq [n']$, $\overline{H} = (H^{(1)}, ..., H^{(j-1)}) \subseteq [n]^{j-1}$, and $\alpha : \overline{H} \to \{0,1\}$, we define the following.*

- *A* $(j,R,\overline{H},\alpha)$-residual local decoder *is a randomized oracle machine that, on input* $\ell \in [k]$ *and oracle access to any* $w \in \{0,1\}^n$*, selects* $r \in R$ *uniformly at random, and emulates the execution of* $D(\ell)$ *on randomness* r*, while answering its queries according to* w *and* α*, analogously to the way that is detailed in the preamble of Step 2 of Construction 4.1. Specifically, for every* $j' \in [q]$*, the* ${j'}^{\text{th}}$ *query, denoted* $q_{j'} \overset{\text{def}}{=} q^{\alpha,w}_{r,j'}$*, is answered by* $w_{q_{j'}}$ *if* $q_{j'} \in \left([n] \setminus \bigcup_{j'' \in [\min(j',j-1)]} H^{(j'')}\right)$ *and by* $\alpha(q_{j'})$ *otherwise.*

- *For fixed* $\ell \in [k]$ *and* $x \in \{0,1\}^k$*, we say that the* $(j,R,\overline{H},\alpha)$*-residual local decoder is* $(1-\eta,\rho)$-safe (w.r.t ℓ and x) *if, on input* ℓ *and oracle access to any string that is* ρ*-close to* $C(x)$*, the decoder's output is in* $\{x_\ell,\perp\}$ *with probability at least* $1-\eta$ (equiv., it outputs $1-x_\ell$ with probability at most η).

- *For fixed* $\ell \in [k]$ *and* $x \in \{0,1\}^k$*, we say that the* $(j,R,\overline{H},\alpha)$*-residual local decoder is* $(1-\eta)$-successful (w.r.t ℓ and x) *if, on input* ℓ *and oracle access to* $C(x)$*, the decoder outputs* x_ℓ *with probability at least* $1-\eta$*.*

- *For fixed* $\ell \in [k]$ *and* $x \in \{0,1\}^k$*, we say that the* $(j,R,\overline{H},\alpha)$*-residual local decoder has* error probability at most η for relative distance ρ (w.r.t ℓ and x) *if it is both* $(1-\eta,\rho)$*-safe and* $(1-\eta)$*-successful* (w.r.t ℓ and x).

Indeed, the local decoder D constitutes a $(1,[n'],\emptyset,\lambda)$-residual local decoder of error probability at most $1/3$ for relative distance δ with respect to any ℓ and x, where δ is the decoding distance of D. On the other hand, each possible $\alpha^{(q)} : H^{(q)}_\ell \to \{0,1\}$ tried in Step 4 (of Construction 4.1) by each leaf in the recursion tree corresponds to a $(q+1,R^{(q)}_\ell,\overline{H}',\alpha')$-residual local decoder, where $(R,\overline{H},\alpha)$ are the auxiliary inputs associated with this leaf, and $R^{(q)}_\ell$, $\overline{H}'$ and $\alpha' : \overline{H}' \to \{0,1\}$ are as defined as in Construction 4.1 (i.e., $\overline{H}'$ combines $\overline{H}$ and $H^{(j)}_\ell$, whereas α' combines α and $\alpha^{(j)}$). The difference between the operation at the leaf (of the recursion tree of the global decoder) and the corresponding residual local decoder is that Step 4 takes a majority vote among all $r \in R^{(q)}_\ell$, whereas the corresponding $(q+1,R^{(q)}_\ell,\overline{H}',\alpha')$-residual local decoder selects $r \in R^{(q)}_\ell$ uniformly at random and uses its verdict.

Our aim is to prove that, for every ℓ and x, with probability $1-o(1/k)$, each of the foregoing $(q+1,R^{(q)}_\ell,\overline{H}',\alpha')$-residual local decoders is $(0.51,0)$-safe with respect to ℓ and x, and that the leaf that corresponds to a path that fits $C(x)$ is 0.51-successful (when t is an ideally adequate pivot). This is done by relating each $(j,R,\overline{H},\alpha)$-residual local decoder to the $(j+1,R^{(j)}_\ell,\overline{H}',\alpha')$-residual local decoders defined by Step 4 of Construction 4.1.

Lemma 4.8.2 (relating a $(j,R,\overline{H},\alpha)$-residual local decoder to the corresponding $(j+1,R^{(j)}_\ell,\overline{H}',\alpha')$-residual local decoders): *Fixing* $(\ell,x) \in [k] \times \{0,1\}^k$ *and* $t \in \{3q+2,...,c-2\}$*, let* $j \in [q]$ *and* $(R,\overline{H},\alpha) \in 2^{[n']} \times 2^{[n]^{j-1}} \times \{0,1\}^*$ *such that* $|R| = (1 \pm (0.001/q))^{j-1} \cdot p^{j-1} \cdot n'$*,* $|\overline{H}| \le s_j \overset{\text{def}}{=} (j-1)\cdot n^{(t-1)/c}$*, and* $\alpha : \overline{H} \to \{0,1\}$*. For* $\eta < 1/2$*, suppose that the* $(j,R,\overline{H},\alpha)$*-residual local decoder is* $(1-\eta,\rho_j)$*-safe* (w.r.t ℓ and x)*, where* $\rho_j \overset{\text{def}}{=} (q+1-j)\cdot n^{((t-1)/c)-1}$*, and consider an* $(j+1,R',\overline{H}',\alpha')$*-residual local decoder derived as follows.*

1. *Defining* $\zeta^{(j)}_\ell$ *based on the uniform distribution on* R (i.e, $\Pr[\zeta^{(j)}_\ell = i] = \Pr_{r\in R}[q^{\alpha,C(x)}_{r,j} = i]$)*, consider the corresponding 3-way partition* $(H^{(j)}_\ell, M^{(j)}_\ell, L^{(j)}_\ell)$ *defined in Construction 4.6. If*

$\Pr[\zeta_\ell^{(j)} \in M_\ell^{(j)}] > 0.12/q$*, then the derivation of the* $(j+1,\cdot,\cdot,\cdot)$*-residual local decoder is aborted.*

(Note that this step aborts if and only if the 3-partition $(H_\ell^{(j)}, M_\ell^{(j)}, L_\ell^{(j)})$ leads Construction 4.6 to failure.)

2. *Otherwise, let* $R^{(j)}$ *be a random set generated as in Step 3 of Construction 4.1. If* $|R^{(j)}| \neq (1 \pm (0.001/q)) \cdot p \cdot |R|$*, then the derivation of the* $(j+1,\cdot,\cdot,\cdot)$*-residual local decoder is aborted.*

   (Note that this step aborts if and only if $R^{(j)}$ leads Construction 4.7 to failure.)

3. *Otherwise, select an arbitrary* $\alpha^{(j)} : H_\ell^{(j)} \to \{0,1\}$*, and let* $\alpha' : \overline{H}' \to \{0,1\}$ *be as in Step 4 of Construction 4.1; that is,* $\alpha'(i) = \alpha(i)$ *if* $i \in \overline{H}$ *and* $\alpha'(i) = \alpha^{(j)}(i)$ *otherwise.*

   (Note that $|\overline{H}'| < s_{j+1} = s_j + n^{(t-1)/c}$, because $|H_\ell^{(j)}| < n^{(t-1)/c}$.)

Then, with probability at least $1 - \exp(-\omega(n^{(t-1)/c}))$ *over the choice of* $R^{(j)}$*, either the derivation is aborted or the* $(j+1, R^{(j)}, \overline{H}', \alpha')$*-residual local decoder is* $(1-\eta-(0.13/q), \rho_{j+1})$*-safe* (w.r.t ℓ and x)*, where* $\rho_{j+1} = (q-j) \cdot n^{((t-1)/c)-1}$*. Furthermore, if the* $(j, R, \overline{H}, \alpha)$*-residual local decoder is* $(1-\eta)$*-successful* (w.r.t ℓ and x) *and* $\alpha'(\overline{H}') = C(x)_{\overline{H}'}$*, then, with probability at least* $1 - \exp(-\omega(n^{(t-1)/c}))$ *over the choice of* $R^{(j)}$*, either the derivation is aborted or the* $(j+1, R^{(j)}, \overline{H}', \alpha')$*-residual local decoder is* $(1-\eta-(0.13/q))$*-successful* (w.r.t ℓ and x).

Note that Lemma 4.8.2 does not say what happends when the derivation of the $(j+1,\cdot,\cdot,\cdot)$-residual local decoder is aborted. In such a case, the corresponding execution of the revised Construction 4.1 returns failure, and this is perfectly fine when considering the branches of the recursion tree that do not fit $C(x)$. But we need to show that such aborting occurs rarely on the branch that does fit $C(x)$, when t is an ideally adequate pivot. Jumping ahead, we note that this will be shown using Lemma 4.4 and Corollary 4.5, but the underlying probability space is not only over the choices of $R^{(j)}$ but rather also over the random generation of R by the $j-1$ prior iterations of Construction 4.1.

Proof: We focus on the main claim, which refers to safety. Given a $(j, R, \overline{H}, \alpha)$-residual local decoder that is $(1-\eta, \rho_j)$-safe, we shall show that, with overwhelimingly high probability (over the choice of $R^{(j)}$), the randomly derived $(j+1, R^{(j)}, \overline{H}', \alpha')$-residual local decoder (in case it was derived) is $(1-\eta-(0.13/q), \rho_{j+1})$-safe. Note that we may assume, without loss of generality, that the $(j+1, R^{(j)}, \overline{H}', \alpha')$-residual local decoder was indeed derived, because any derivation failure is counted in our favor.

Recall that we need to consider all n-bit long strings w' that are ρ_{j+1}-close to $C(x)$ and analyze the output of the randomly derived $(j+1, R^{(j)}, \overline{H}', \alpha')$-residual local decoder when given oracle access to w'. For each such w', we use w that combines w' and $\alpha^{(j)}$ (i.e., $w_i = \alpha^{(j)}(i)$ if $i \in H_\ell^{(j)}|$ and $w_i = w'_i$ otherwise), while observing that w is ρ_j-close to $C(x)$, because $|H_\ell^{(j)}| \leq n^{(t-1)/c}$ and $\rho_j = \rho_{j+1} + n^{((t-1)/c)-1}$. For this w, we will show that $R^{(j)}$ yields a "bad" residual decoder (i.e., one that is not $(1-\eta-(0.13/q), \rho_{j+1})$-safe) with probability $\exp(-\Omega(p^2 \cdot n^{t/c}))$. The proof is analogous to the argument at the end of the proof of Theorem 3.3. Specifically, we shall rely on the behavior of the original $(j, R, \overline{H}, \alpha)$-residual local decoder when given oracle access to w. Details follow.

We call $r \in R$ good for $w = (w', \alpha^{(j)})$ if the $(j, R, \overline{H}, \alpha)$-residual local decoder's output is in $\{x_\ell, \perp\}$ when it uses randomness r and is given oracle access to w. By the hypothesis (that the $(j, R, \overline{H}, \alpha)$-residual local decoder is $(1 - \eta, \rho_j)$-safe), the fraction of good r's in R is at least $1 - \eta > 1/2$. Furthermore, by the hypothesis that the first step did not abort, at most a $0.12/q$ fraction of the r's in R yield a $j^{\rm th}$ query that is in $M_\ell^{(j)}$. We denote the set of remaining good r's by G', while noting that $|G'| \geq (1 - \eta - (0.12/q)) \cdot |R|$.

Towards applying Lemma 3.1, we consider a graph on the vertex set R such that r and s are connected if and only if they lead to the same $j^{\rm th}$ query (i.e., $q_{r,j}^{\alpha,C(x)} = q_{s,j}^{\alpha,C(x)}$) and this query is in $L_\ell^{(j)}$. Indeed, we use the fact that each r that yields a $j^{\rm th}$ query in $H_\ell^{(j)}$ is included in $R^{(j)}$ at random, *independently of any other event*, whereas vertices that yield a $j^{\rm th}$ query in $M_\ell^{(j)}$ are ignored (i.e., their corresponding random variable is identically zero). Hence, the only non-isolated vertices are those that yield a $j^{\rm th}$ query in $L_\ell^{(j)}$, and the degree of each such vertex is at most $n^{-t/c} \cdot |R|$. Now, applying Lemma 3.1 to the current context (i.e., considering random variables that indicate whether $r \in (R^{(j)} \cap G')$), while recalling that $|R^{(j)}| = (1 \pm (0.001/q)) \cdot p \cdot |R|$, since the second step did not abort, we get

$$\begin{aligned}
&{\rm Pr}_{R^{(j)}}\left[|R^{(j)} \cap G'| < \left(1 - \eta - \frac{0.13}{q}\right) \cdot |R^{(j)}|\right] \\
&\leq\ {\rm Pr}_{R^{(j)}}\left[|R^{(j)} \cap G'| < \left(1 - \eta - \frac{0.13}{q}\right) \cdot \left(1 + \frac{0.001}{q}\right) \cdot p \cdot |R|\right] \\
&\leq\ {\rm Pr}_{R^{(j)}}\left[|R^{(j)} \cap G'| < \left(1 - \eta - \frac{0.12}{q}\right) \cdot p \cdot |R| - \frac{0.01}{q} \cdot p \cdot |R| + \frac{0.001}{q} \cdot p \cdot |R|\right] \\
&\leq\ {\rm Pr}_{R^{(j)}}\left[|R^{(j)} \cap G'| < p \cdot |G'| - \frac{0.009}{q} \cdot p \cdot |R|\right] \\
&\leq\ \exp\left(-\Omega\left((0.009 \cdot p/q)^2 \cdot \frac{|R|}{n^{-t/c} \cdot |R|}\right)\right) \\
&=\ \exp(-\Omega(p^2 \cdot n^{t/c})).
\end{aligned}$$

Using $p > n^{-1/2c} \log n$, we have $p^2 \cdot n^{t/c} = \omega(n^{(t-1)/c} \cdot \log n)$, which implies

$${\rm Pr}_{R^{(j)}}\left[|R^{(j)} \cap G'| < \left(1 - \eta - \frac{0.13}{q}\right) \cdot |R^{(j)}|\right] \ =\ \exp(-\omega(n^{(t-1)/c} \cdot \log n)). \tag{13}$$

Combining Eq. (13) with a union bound on all n-bit long strings (i.e., $w' \in [n]$) that are ρ_{j+1}-close to $C(x)$, we conclude that the probability that the $(j + 1, R^{(j)}, \overline{H}', \alpha')$-residual local decoder is not $(1 - \eta - (0.13/q), \rho_{j+1})$-safe is at most

$$\begin{aligned}
&\sum_{i=0}^{(q-j)\cdot n^{(t-1)/c}} \binom{n}{i} \cdot \exp(-\omega(n^{(t-1)/c} \cdot \log n)) \\
&<\ \exp(O(n^{(t-1)/c} \cdot \log n) - \omega(n^{(t-1)/c} \cdot \log n))
\end{aligned}$$

and the main claim of the lemma follows.

Turning to the furthermore claim and assuming that $\alpha'(\overline{H}') = C(x)_{\overline{H}'}$, we define $r \in R$ as good if the $(j, R, \overline{H}, \alpha)$-residual local decoder outputs x_ℓ when it uses randomness r and is given oracle access to $C(x)$. By the hypothesis that the $(j, R, \overline{H}, \alpha)$-residual local decoder is $(1 - \eta)$-successful, the fraction of good r's in R is at least $1 - \eta > 1/2$. Proceeding as in the main case, while defining

G' accordingly, we observe that Eq. (13) still holds. This establishes the furthermore claim (where in this case there is no need for a union bound). ■

Conclusion. The probability bound in Lemma 4.8.2 is small enough to withstand, for any $j \in [q]$, a union bound over all possible admissible values of the parameters $(R, \overline{H}, \alpha)$ and over all possible choices of $\alpha^{(j)}$'s.[20] Recalling that the $(1, [n'], \emptyset, \lambda)$-residual local decoder is $(2/3, \rho_1)$-safe, the foregoing implies that, with overwhelmingly high probability (certainly $1 - o(1/k)$), all leaves in the recursion tree generated by an execution of Construction 4.1 (with $j = 1$) correspond to $(q+1, \cdot, \cdot, \cdot)$-residual local decoders are $((2/3) - 0.13, 0)$-safe. This implies that, with probability $1 - o(1/k)$, the global decoder does not output a wrong (bit) value.

Using Lemma 4.4 and Corollary 4.5 it follows that, when using an ideally adequate pivot t and answering all queries according to $C(x)$, with overwhelmingly high probability (certainly $1 - o(1/k)$), the sequence of $R^{(j)}$'s does not lead to abort. Recalling that the $(1, [n'], \emptyset, \lambda)$-residual local decoder is $2/3$-successful, the furthermore claim of Lemma 4.8.2 implies that, with overwhelmingly high probability (certainly $1 - o(1/k)$), the leaf in the recursion tree that fits $C(x)$ corresponds to a $(q+1, \cdot, \cdot, \cdot)$-residual local decoder that is $((2/3) - 0.13)$-successful. Hence, with probability $1 - o(1/k)$, at least one leaf returns the correct value, whereas no leaf returns a wrong (bit) value.

Hence, on input ℓ and oracle access to $C(x)$, when using an ideally adequate pivot t, with probability $1 - o(1/k)$, the global decoder outputs the correct value x_ℓ. As indicated at the very beginning of the proof, this establishes the claim of the theorem, since we used $p = n^{-1/cq} \cdot \log k$, where that $c = 3 \cdot (10q + 2) \cdot q < 30 \cdot (q+1)^2$. ■

Digest. The core of the proof of Theorem 4.8 is the definition of residual local decoders and their analysis. These residual local decoder allow to define a notion of good choices (i.e., r's) at any level and branch of the recursion tree (of the revised Construction 4.1). The core of this analysis is captured by Lemma 4.8.2, which holds for any $p > n^{-1/2c} \cdot \log n$, where $c = \Theta(q^2)$. In contrast, Lemma 4.4 relies on $p > n^{-1/cq}$, where this reliance occurs (only) in showing that, with overwhelmingly high probability, Condition 1 of Claim 4.4.1 holds. We believe that there is a way of getting rid of this deficiency, and analyzing the global decoder (or a version of it) when using any $p > n^{-1/2c} \cdot \log n$ (or so). Such an analysis would resolve the following

Open Problem 4.9 (improving the bound of Theorem 4.8): *Prove that if $C : \{0,1\}^k \to \{0,1\}^n$ is a general q-query relaxed LDC, then $n > k^{1+\Omega(1/q^2)}$ must hold.*

We mention that, as shown recently [6], in the case of *linear codes*, there is no gap between (one-sided error) non-adaptive local decoders and (two-sided error) adaptive local decoders.

5 Concluding remarks

When comparing our presentation to the previous work of [9, 4] one should bear in mind that [9] treats only the case of non-adaptive one-sided error local decoders, which we treat in Section 2, whereas the more general cases are treated in [4].

[20] Note that this union bound is over $2^n \cdot \binom{n}{(j-1) \cdot n^{(t-1)/c}} \cdot 2^{j \cdot n^{(t-1)/c}} < n^{j \cdot n^{(t-1)/c}}$ events, whereas the union bound employed at the end of the proof of Lemma 4.8.2 was over $\sum_{i=0}^{(q-j) \cdot n^{(t-1)/c}} \binom{n}{i} < n^{(q-j) \cdot n^{(t-1)/c}}$ events. The point is that for every $j \in [q]$ we apply a union bound over $\exp(\Theta(n^{(t-1)/c} \cdot \log n))$ events.

Recall that, while our lower bound for the non-adaptive case improves over the one in [9, 4], our lower bound for the general (adaptive) case is inferior. We believe that the approach used in Section 4 (for the general case) can meet the bound established in the non-adaptive case, but we left the closing of this gap as an open problem (see Problem 4.9).

5.1 On the dependence on q in the exponent of the bounds

It seems that the strategy of designing a global decoder that operate based on a random sample of bits in the codeword has no hope of yielding a lower bound better than the one of Theorem 2.2 (i.e., $n > k^{1+\frac{1}{q-1}-o(1)}$), even for the case of q-query (non-relaxed) LDCs. The reason is that if we sample each location in the codeword with probability p, then the probability that we see a specific q-tuple is p^q, where recovery of the message requires $O(n) \cdot p^q = \Omega(1)$, which yields a lower bound of the type $n = \Omega(k/p) = \Omega(k/n^{-1/q})$.

We stress that the foregoing lower bound was not obtained in the case of q-query *relaxed* LDCs (rLDCs). In that case, even in the non-adaptive case, we only obtained bounds of the type $n > k^{1+\Omega(1/q^2)}$. So far, we did not attempt to optimize the constant in the Omega-notation, but let us do it now. First, as stated in the digest at the end of Section 2, in the case of non-adaptive rLDC with one-sided error, we can derive a lower bound of $n > k^{1+\frac{1}{2(q+1)^2}}$. Actually, the constant 2 can be replaced by any constant larger than 3/2, because all that we need is that the probability of making an "intermediate query" be smaller than 2/3. In the case of general non-adaptive rLDC, Theorem 3.3 states a lower bound of $n > k^{1+\frac{1}{20(q+1)^2}}$, but the constant 20 can be replaced by any constant larger than 12, because all that we need is that the probability of making an "intermediate query" be smaller than 1/6. Actually, replacing Lemma 3.1 by an analogous "multiplicative Chernoff bound" (which holds (in this case) for identical random variables) we can replace the constant 20 by any constant larger than 6. (As mentioned at the beginning of this section, in the case of general (adaptive) rLDC, Theorem 4.8 only establishes a lower bound of $n > k^{1+\frac{1}{O(q^3)}}$, which we believe to be improvable.)

Needless to say, discussing the constant in the exponent misses the real issue, which is closing the gap between an $n > k^{1+\Omega(1/q^2)}$ lower bound and the $n \leq k^{1+O(1/q)}$ upper bound. Recall that a q-query relaxed LDC of length $n = k^{1+o(1/q)}$ would yield a separation between relaxed LDCs and (non-relaxed) LDCs. However, we tend to conjecture that q-query relaxed LDCs require length $n \geq k^{1+\Omega(1/q)}$.[21]

Open Problem 5.1 (closing the gap between the lower and upper bounds for rLDCs): *Prove or disprove the conjecture that if $C : \{0,1\}^k \to \{0,1\}^n$ is a q-query relaxed LDC, then $n > k^{1+\Omega(1/q)}$ must hold.*

While the problem is implicitly stated for general rLDCs, even a lower bound that refers only to non-adaptive local decoders that have one-sided error would be interesting.

We stress that improving the lower bound on rLDCs is relevant to the project of separating rLDCs from LDCs only in case that they prevent proving a separation by presenting rLDCs that

[21] Our conjecture should be taken with a grain of salt. We were wrong twice about q-query (non-relaxed) LDCs: Once conjecturing an $n \geq \exp(k^{\Omega(1/q)})$ lower bound, and later an $n \geq \exp(k^{1/F(q)})$ lower bound for an unspecified $F : \mathbb{N} \to \mathbb{N}$.

beat the LDC lower bound. Recalling that the best known LDC lower bound (of [12])[22] is essentially $n = \Omega(k/\log k)^{1+\frac{1}{\lceil q/2\rceil-1}}$, whereas the rLDC upper bound (of [2]) is $n = O(k^{1+O(1/q)})$, such a development seems unlikely. Hopefully, a separation will be proven by increasing the lower bound on LDCs; specifically, by showing that q-query (non-relaxed) LDC requires length $n \geq k^{1+\omega(1/q)}$. More generally, we pose the following problem.

Open Problem 5.2 (separating rLDCs from LDCs): *Present a natural context in which relaxed LDCs are superior to general* (non-relaxed) *LDCs.*

Needless to say, a separation between the lengths of codes having general q-query decoders is the holy grail, but also results regarding one-sided error non-adaptive decoders or linear codes would be most welcome.

5.2 Applicability to robust local algorithms

It seems that our treatment can be applied to the general notion of *robust local algorithms* as defined in [4]. This notion generalizes both LDCs and property testing, and a relaxation of it covers also rLDCs.

It is instructive to present robust local algorithms as a generalization of LDCs; specifically, by considering a locally decodable code $C : \{0,1\}^k \to \{0,1\}^n$. Loosely speaking, *local algorithms* are randomized oracle machines that, given an explicit (secondary) input (e.g., $\ell \in [k]$), make a constant number of queries to their main input (i.e., an n-bit string), and solve a promise problem (e.g., distinguishing $\{C(x) : x_\ell = 1\}$ from $\{C(x) : x_\ell = 0\}$). Needless to say, this is possible only if the YES-instances (e.g., $\{C(x) : x_\ell = 1\}$) are far from the NO-instances (e.g., $\{C(x) : x_\ell = 0\}$). Such an algorithm is called robust if it answers correctly, with high probability, also when the input is close to the promise set (i.e., the set of YES and NO-instances). The *relaxed* version allows these algorithm to fail (but not to output a wrong answer), whenever the input is close to the promise set (but it is still required to answer correctly when the input is in the promise set).

Indeed, the local decoders of LDCs (resp, rLDCs) are a special case of robust (resp., relaxed robust) local algorithms. The correspondence to property testing (cf. [7]) may be less clear. Nevertheless, tolerant testers (cf. [14] or [7, Sec. 12.1]), with fixed proximity and tolerance parameters, are robust local algorithms. Standard property testers are local algorithms, but their robustness applies only to NO-instances.[23]

In any case, the crux of our proof is the presentation of a global decoder that works based on a sample of the input bits. Hence, following [4], we observe that if a problem has a robust local algorithm (even in the relaxed sense), then it has a "sample-based" local algorithm that uses $n^{1-O(1/q^3)}$ samples, where q is the number of queries made by the robust local algorithm (and n is the length of the main input). In analogy to sample-based testers [8] (see also [7, Sec. 12.3]), sample-based local algorithms query (or rather sample) their main input at uniformly and *independently* distributed locations.

[22] Indeed, the $\Theta(\log k)^{\frac{1}{\lceil q/2\rceil-1}}$ factor improvement of [15] is not relevant to our argument.

[23] Specifically, for a property Π and proximity parameter ϵ, letting $\Gamma_\delta(\Pi)$ denote all strings that are δ-far from Π, we may view an ϵ-tester for Π as a local algorithm with (one-sided) robustness ϵ' for distinguishing Π from $\Gamma_{\epsilon+\epsilon'}(\Pi)$. On the other hand, even a non-robust local algorithm for distinguishing Π from $\Gamma_{\epsilon+\epsilon'}(\Pi)$ constitutes an $(\epsilon+\epsilon')$-tester for Π.

Acknowledgements

I am deeply indebted to Guy Goldberg for an oral presentation of [9]. This presentation has been the starting point of my own work. We are also grateful to Noga Alon for calling our attention to [1] and providing us with the alternative proof of Lemma 3.1, which is reproduced in the appendix with his permission.

References

[1] Noga Alon and Zoltan Furedi, Spanning subgraphs of random graphs. *Graphs and Combinatorics*, Vol. 8, pages 91–94, 1992.

[2] Vahid Asadi and Igor Shinkar. Relaxed Locally Correctable Codes with Improved Parameters. In *48th ICALP*, pages 18:1–18:12, 2021.

[3] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding. *SIAM Journal on Computing*, Vol. 36 (4), pages 889–974, 2006.

[4] Marcel Dall'Agnol, Tom Gur and Oded Lachish. A Structural Theorem for Local Algorithms with Applications to Coding, Testing, and Privacy. In *32nd ACM-SIAM Symposium on Discrete Algorithms*, pages 1651–1665, 2021.

[5] Klim Efremenko. 3-query locally decodable codes of subexponential length. *SICOMP*, Vol. 41 (6), pages 1694–1703, 2012.

[6] Guy Goldberg. Linear Relaxed Locally Decodable and Correctable Codes Do Not Need Adaptivity and Two-Sided Error. In preparation, 2023.

[7] Oded Goldreich. *Introduction to Property Testing.* Cambridge University Press, 2017.

[8] Oded Goldreich and Dana Ron. On Sample-Based Testers. *ACM Trans. Comput. Theory*, Vol. 8 (2), pages 7:1–7:54, 2016.

[9] Tom Gur and Oded Lachish. On the Power of Relaxed Local Decoding Algorithms. *SIAM Journal on Computing*, Vol. 50 (2), pages 788–813, 2021.

[10] Andras Hajnal and Endre Szemeredi Proof of a conjecture of P. Erdos. In *Combinatorial Theory and its Applications*, Vol. 2 (Editors P. Erdos, A. Renyi, and V. Sos). North-Holland, London, pp. 601–623, 1970.

[11] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *32nd ACM Symposium on the Theory of Computing*, pages 80–86, 2000.

[12] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and System Science*, Vol. 69(3), pages 395–420, 2004.

[13] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-Rate Locally Correctable and Locally Testable Codes with Sub-Polynomial Query Complexity. *Journal of the ACM*, Vol. 64 (2), pages 11:1–11:42, 2017.

[14] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Science*, Vol. 72(6), pages 1012–1042, 2006.

[15] David P. Woodruff. A quadratic lower bound for three-query linear locally decodable codes over any field. *Journal of Computer Science and Technology*, Vol. 27 (4), pages 678–686, 2012.

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

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

Appendix: An alternative proof of Lemma 3.1

The following proof was suggested to us by Noga Alon, and is based on two observations. The first observation is that the proof of Lemma 3.1 only needs a partition of the graph to $O(d)$ independent sets that are each of size $\Omega(|V|/d)$. The second observation is that such a partition is easy to find. Actually, a stronger claim holds.

Claim: *There exists a linear-time algorithm that, for every $d, d', n \in \mathbb{N}$, given an n-vertex graph of maximal degree at most d, finds a partition of the vertex set to $d + d'$ independent sets, each of size at least $(n - d - d' + 1)/(2d + d')$ and at most $(n + d' - 1)/d'$.*

For $d' = d$, we get a lower-bound of $(n - 2d + 1)/3d$ and an upper-bound of $(n + d - 1)/d$.

Proof: We used a straightforward greedy algorithm for coloring the graph $G = ([n], E)$ in $d + d'$ colors. It proceeds in iterations such that, at each iteration, it picks an uncolored vertex and assign it a colors that is (different from its colored neighbors and is) used least frequently so far. That is, starting with $S_1 = \cdots = S_{d+d'} = \emptyset$, at iteration $i \in [n]$, we let $S_j \leftarrow S_j \cup \{i\}$ *if vertex i does not neighbor any vertex in S_j, and S_j is a smallest set that contains no neighbor of i.*

To prove that $\max_{i\in[d+d']}\{|S_i|\} \leq (n + d' - 1)/d'$, we pick a largest set S_i, and let $m = |S_i|$. Looking at the iteration in which the last vertex v was added to S_i, we observe that at that iteration at least $(d + d' - 1) - d$ other sets (i.e., the sets containing no neighbor of v) had size at least $m - 1$ each, becuase otherwise v would have been added to one of the violating sets. Hence, $m + (d' - 1) \cdot (m - 1) \leq n$, and $m \leq (n + d' - 1)/d'$ follows.

To prove that $\min_{i\in[d+d']}\{|S_i|\} \geq (n - d - d' + 1)/(2d + d')$, we pick a smallest set S_i, and let $m = |S_i|$. Here we consider the set of iterations, denoted T, in which S_i was not increased although it contains no neighbour of the examined vertex; that is, $t \in T$ if neither t nor a neighbor of t is in S_i. On the one hand, $|T| \geq n - (d+1) \cdot m$, because $(d+1) \cdot m$ upper bounds the number of vertices that are either in S_i or neighbor some vertex in S_i. On the other hand, at each iteration $t \in T$, the vertex t was added to a set that had size at most m at time $t - 1$ (or else t should have been added to S_i instead). For each $j \in [d + d'] \setminus \{i\}$, looking at the last iteration in which an element was addded to S_j, we conclude that $|T \cap S_j| \leq m + 1$, which implies $|T| \leq (d + d' - 1) \cdot (m + 1)$. Hence, $n - (d + 1) \cdot m \leq (d + d' - 1) \cdot (m + 1)$, and $m \leq (n - d - d' + 1)/(2d + d')$ follows. ■

Comment: Lemma 3.1 replaces a weaker result that we used in an earlier draft, which showed a probability bound of $\exp(-\Omega(\gamma^3 \cdot |V|/(d+1)))$ rather than $(d+1) \cdot \exp(-\Omega(\gamma^2 \cdot |V|/(d+1)))$. This weaker result was proved by finding a sequence of $t = O(d \cdot \log(1/\gamma))$ disjoint independent sets, denoted $S_1, ..., S_t$, that cover at least $(1 - 0.5\gamma) \cdot |V|$ of the elements of V and satisfy $|S_i| \geq \gamma|V|/2$ for every $i \in [t]$. This sequence can be found by a greedy algorithm that, in each step, finds an independent set that covers a $1/(d + 1)$ fraction of the elements that were not covered so far. Specifically, for each $i \in [t]$, letting U_i denote the set of elements that were not covered by the first i steps (with $U_0 = V$), we have $|U_i| \leq \frac{d}{d+1} \cdot |U_{i-1}|$, which implies $|U_t| \leq (d/(d+1))^t \cdot |V| < \gamma \cdot |V|/2$. Furthermore, we may assume, without loss of generality that $S_i \overset{\text{def}}{=} U_{i-1} \setminus U_i$ has cardinality $\lceil |U_{i-1}|/(d + 1)\rceil$.