Optimal Reconstruction Codes for Deletion Channels

Johan Chrisnata*†, Han Mao Kiah*, and Eitan Yaakobi†
*School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371
†Department of Computer Science, Technion — Israel Institute of Technology, Haifa, 3200003 Israel
Emails: johanchr001@ntu.edu.sg, hmkiah@ntu.edu.sg, yaakobi@cs.technion.ac.il

***Abstract*—The sequence reconstruction problem, introduced by Levenshtein in 2001, considers a communication scenario where the sender transmits a codeword from some codebook and the receiver obtains multiple noisy reads of the codeword. Motivated by modern storage devices, we introduced a variant of the problem where the number of noisy reads N is fixed (Kiah *et al.* 2020). Of significance, for the single-deletion channel, using $\log_2 \log_2 n + O(1)$ redundant bits, we designed a codebook of length n that reconstructs codewords from two distinct noisy reads.**

In this work, we show that $\log_2 \log_2 n - O(1)$ redundant bits are necessary for such reconstruction codes, thereby, demonstrating the optimality of our previous construction. Furthermore, we show that these reconstruction codes can be used in t-deletion channels (with $t \geqslant 2$) to uniquely reconstruct codewords from $n^{t-1} + O\left(n^{t-2}\right)$ distinct noisy reads.

I. Introduction

As our data needs surge, new technologies emerge to store these huge datasets. Interestingly, besides promising ultra-high storage density, certain emerging storage media, such as DNA based storage [1]–[4] and racetrack memories [5]–[7], rely on technologies that provide users with multiple cheap, albeit noisy, reads. In our companion paper [9], we proposed a *coding solution* to leverage on these multiple reads to increase the information capacity of these next-generation devices, or equivalently, reduce the number of redundant bits.

Our code design problem is based on the *sequence reconstruction problem*, formulated by Levenshtein [8]. In Levenshtein's seminal work, he considers a communication scenario where the sender transmits a codeword from some codebook and the receiver obtains multiple noisy reads of the codeword. The common setup assumes the codebook to be the entire space and the problem is to determine the minimum number of distinct reads N that is required to reconstruct the transmitted codeword. In constrast, in our problem, the parameter N is fixed and our task is to design a *codebook* such that every codeword can be uniquely reconstructed from any N distinct noisy reads.

Hence, our fundamental problem is then: how large can this codebook be? Or equivalently, what is the *minimum redundancy*? Modifying a code construction in [7], we provided in [9] a number of reconstruction codes for the single-edit channel and its variants with $\log_2 \log_2 n + O(1)$ bits of redundancy. In this paper, we focus on the converse of the problem and demonstrate that $\log_2 \log_2 n - O(1)$ redundant bits are *necessary*. To ease our exposition, we focus on channels with *deletions only* and our first contribution is to demonstrate this lower bound on redundancy for the case $N = 2$ and single deletions.

In our proof, we characterize the conditions when two single-deletion balls have intersection size two (i.e. when two different codewords result in two noisy reads through the single-deletion channel). In this same spirit, we determine when two single-deletion balls have intersection size one. Using this characterization, we show that the same reconstruction code for the single-deletion channel can be used to uniquely reconstruct codewords with approximately half the number of reads (as compared to the case with no coding) for the t-deletions channel with $t \geqslant 2$. We formally describe our problem and results next.

II. Preliminaries

Consider a data storage scenario described by an error-ball function. Formally, given an input space $\mathcal{X}$ and an output space $\mathcal{Y}$, an *error-ball* function B maps a *word* $\boldsymbol{x} \in \mathcal{X}$ to a subset of *noisy reads* $B(\boldsymbol{x}) \subseteq \mathcal{Y}$. Given a code $\mathcal{C} \subseteq \mathcal{X}$, we define the *read coverage* of $\mathcal{C}$, denoted by $\nu(\mathcal{C}; B)$, to be the quantity

$$\nu(\mathcal{C}; B) \triangleq \max\left\{|B(\boldsymbol{x}) \cap B(\boldsymbol{y})| : \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}, \boldsymbol{x} \neq \boldsymbol{y}\right\}.$$

In other words, $\nu(\mathcal{C}; B)$ is the maximum intersection between the error-balls of any two codewords in $\mathcal{C}$. The quantity $\nu(\mathcal{C}; B)$ was introduced by Levenshtein [8], where he showed that the number of reads required to reconstruct a codeword from $\mathcal{C}$ is at least $\nu(\mathcal{C}; B) + 1$. The problem to determine $\nu(\mathcal{C}; B)$ is referred to as the *sequence reconstruction problem*.

The sequence reconstruction problem was studied in a variety of storage and communication scenarios [7], [10]–[17]. In these cases, the codebook is usually assumed to be the entire space (all binary words of some fixed length) or a classical error-correcting code. However, in most storage scenarios, the number of noisy reads N is a fixed system parameter and when N is at most $\nu(\{0,1\}^n; B)$, we are unable to uniquely reconstruct the codeword. In [9], we propose the study of *code design* when the number of noisy reads is strictly less than $\nu(\{0,1\}^n; B)$. Formally, we say that $\mathcal{C}$ is an $(n, N; B)$-*reconstruction code* if $\mathcal{C} \subseteq \{0,1\}^n$ and $\nu(\mathcal{C}; B) < N$.

This gives rise to a *new quantity of interest* that measures the *trade-off between codebook redundancy and read coverage*. Specifically, given N and an error-ball B, we study the quantity

$$\rho(n, N; B) \triangleq \min\left\{n - \log|\mathcal{C}| : \mathcal{C} \subseteq \{0,1\}^n, \nu(\mathcal{C}; B) < N\right\}.$$

A. Levenshtein's Sequence Reconstruction Problem

In this work, we focus on channels that introduce *deletions only*. Specifically, let $\mathcal{D}_t(\boldsymbol{x})$ denote the deletion ball of $\boldsymbol{x}$ with exactly t deletions. Let $D_t(n)$ denote the maximum deletion ball size of words of length n, that is, $D_t(n) = \max\{|\mathcal{D}_t(\boldsymbol{x})| : \boldsymbol{x} \in \{0,1\}^n\}$. It is well known (see for example, [18]) that

$$D_t(n) = \sum_{i=0}^{t} \binom{n-t}{i} = n^t + O(n^{t-1}), \text{ for } 0 \leqslant t \leqslant n. \quad (1)$$

For convenience, we assign $D_t(n) = 0$ when $t < 0$ or $t > n$.

For purposes of brevity, we let $\nu_t(n)$ denote $\nu(\{0,1\}^n; \mathcal{D}_t)$, the read coverage of $\{0,1\}^n$. We have the following landmark result of Levenshtein.

Theorem 1 (Levenshtein [18])**.**

$$\nu_t(n) = 2D_{t-1}(n-2) = 2n^{t-1} + O(n^{t-2})\,. \tag{2}$$

Recently, the authors of [11] studied the sequence reconstruction problem when $\mathcal{C}$ is a single-deletion-correcting code or an $(n, 1; \mathcal{D}_1)$-reconstruction code. Namely, they showed that $\mathcal{C}$ allows unique reconstruction with significantly less reads (as compared to $\nu_t(n)$) for deletions with $t \geqslant 2$.

Theorem 2 ([11])**.** *Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two words of length $n \geqslant 7$. If $\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y}) = \varnothing$, then $|\mathcal{D}_t(\boldsymbol{x}) \cap \mathcal{D}_t(\boldsymbol{y})| \leqslant N_t^{(1)}(n)$ for $t \geqslant 2$, where*

$$\begin{aligned} N_t^{(1)}(n) &= 2D_{t-2}(n-4) + 2D_{t-2}(n-5) + 2D_{t-2}(n-7) \\ &\quad + D_{t-3}(n-6) + D_{t-3}(n-7) = 2n^{t-2} + O(n^{t-3}). \end{aligned} \tag{3}$$

Hence, if $\mathcal{C}$ is an $(n, 1; \mathcal{D}_1)$-reconstruction code, then $\nu(\mathcal{C}; \mathcal{D}_t) \leqslant N_t^{(1)}(n)$. So, for $t \geqslant 2$ and $n \geqslant 7$, $\mathcal{C}$ is also an $\left(n, N_t^{(1)}(n) + 1; \mathcal{D}_t\right)$-reconstruction code. Furthermore, this implies that $\rho\left(n, N_t^{(1)}(n) + 1; \mathcal{D}_t\right) \leqslant \log_2 n + O(1)$.

In the same spirit, we study the sequence reconstruction problem when the codebook $\mathcal{C}$ is an $(n, 2; \mathcal{D}_1)$-reconstruction code. Specifically, in Section IV, we show that if every channel introduces t deletions, then it is possible to uniquely reconstruct codewords from $\mathcal{C}$ with approximately $\nu_t(n)/2$ reads.

B. Reconstruction Codes with $N = 2$ for Single Deletions

We motivate the case for reconstruction codes in the context of the single-deletion channel. As mentioned early, when we use the whole space $\{0, 1\}^n$ as our codebook, we require $\nu_1(n) + 1 = 3$ noisy reads to uniquely reconstruct any codeword. Hence, we have $\rho(n, N; \mathcal{D}_1) = 0$ for $N \geqslant 3$.

In contrast, when $N = 1$, or, when we have only one noisy read, we recover the usual notion of error-correcting codes and the classical Varshamov-Tenengolts (VT) code is an $(n, 1; \mathcal{D}_1)$-reconstruction code whose redundancy is at most $\log_2(n+1)$ [19]. Hence, we have $\rho(n, 1; \mathcal{D}_1) = \log_2 n + \Theta(1)$. Therefore, it remains to ask: how should we design the codebook when we have only two noisy reads? Or, what is the value of $\rho\,(n, 2; \mathcal{D}_1)$?

Now, the first construction of a $(n, 2; \mathcal{D}_1)$-reconstruction code was proposed in [7] for the design of codes in racetrack memory. The codebook uses $\log_2 \log_2 n + O(1)$ redundant bits and in [9], we modified the construction to obtain codebooks that uniquely reconstruct codewords for the single-edit channel and its variants. The construction can be seen as a generalization of the classical VT code proposed by Levenshtein [19] and the shifted VT codes proposed by Schoeny *et al.* [20].

Definition 3 (Constrained Shifted VT Codes [7], [9])**.** For $n \geqslant P > 0$ and P even, let $c \in \mathbb{Z}_{1+P/2}$ and $d \in \mathbb{Z}_2$. The *constrained shifted VT code* $\mathcal{C}_{\mathrm{CSVT}}(n, P; c, d)$ is defined to be the set of all words $\boldsymbol{x} = x_1 x_2 \cdots x_n$ such that the following holds.

(i) $\mathrm{Syn}(\boldsymbol{x}) = c \pmod{1 + P/2}$.
(ii) $\sum_{i=1}^{n} x_i = d \pmod 2$.
(iii) The longest 2-periodic run in $\boldsymbol{x}$ is at most P.

Here, $\mathrm{Syn}(\boldsymbol{x})$ denotes the *VT syndrome* $\mathrm{Syn}(\boldsymbol{x}) \triangleq \sum_{i=1}^{n} i x_i$ and a *2-periodic* run refers to a continguous substring $x_i x_{i+1} \cdots x_j$ where $x_k = x_{k+2}$ for all $i \leqslant k \leqslant j-2$.

When $P = 2n$ and we remove Condition (ii)[1] we recover the classical VT code that corrects a single deletion. On the other hand, when we remove the Condition (iii), we recover the shifted VT code that is used in the correction of a single burst of deletions [20]. It was recently demonstrated that the CSVT code enables unique reconstruction whenever we have two distinct noisy reads.

Theorem 4 ([7], [9])**.** *For all choices of c and d, we have that $\mathcal{C}_{\mathrm{CSVT}}(n, P; c, d)$ is an $(n, 2; \mathcal{D}_1)$-reconstruction code. Furthermore, if we set $P = \lceil \log_2 n \rceil + 2$, the code $\mathcal{C}_{\mathrm{CSVT}}(n, P; c, d)$ has redundancy $\log_2 \log_2 n + O(1)$ for some choice of c and d. Thus, $\rho(n, 2; \mathcal{D}_1) \leqslant \log_2 \log_2 n + O(1)$.*

In this paper, we demonstrate that the codes in Theorem 4 are asymptotically *optimal*. Specifically, in Section III, we show that an $(n, 2; \mathcal{D}_1)$-reconstruction code requires at least $\log_2 \log_2 n - O(1)$ redundant bits.

To demonstrate this necessary condition, we first observe that $\nu_1(n) = 2$ and thus, we need to characterize pairs of words whose single-deletion balls have intersection size exactly two. To do so, we have the following definition of confusability.

Definition 5. Two words $\boldsymbol{x}$ and $\boldsymbol{y}$ are *Type-A-confusable* if $\boldsymbol{x} = \boldsymbol{uav}$ and $\boldsymbol{y} = \boldsymbol{u\bar{a}v}$ for some subwords $\boldsymbol{a}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ such that $|\boldsymbol{a}| \geqslant 2$, $\bar{\boldsymbol{a}}$ is the complement of $\boldsymbol{a}$, and $\boldsymbol{a} = a_1 a_2 \ldots a_j$ is an *alternating sequence*, that is, $\boldsymbol{a}$ is 2-periodic and $a_1 \neq a_2$.

The following characterization was demonstrated in [9].

Lemma 6 (Type-A-confusability [9])**.** *Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be binary words. We have that $|\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y})| = 2$ if and only if $\boldsymbol{x}$ and $\boldsymbol{y}$ are Type-A-confusable.*

In Section IV, we derive an analogous result that characterizes when two single-deletion balls intersect at exactly one word. Using this characterization, we then analyse the read coverage of an $(n, 2; \mathcal{D}_1)$-reconstruction code.

C. Main Contributions

In summary, our contributions are as follows.

- In Section III, we consider the case where $t = 1$ and $N = 2$, and demonstrate that a $(n, 2; \mathcal{D}_1)$-reconstruction code requires at least $\log_2 \log_2 n - O(1)$ bits of redundancy. Therefore, the CSVT code constructed in Theorem 4 is asymptotically optimal and we have that $\rho(n, 2; \mathcal{D}_1) = \log \log n + \Theta(1)$. Furthermore, we have the complete solution for ρ in the case for $t = 1$.

  Theorem 7. *The value $\rho(n, N; \mathcal{D}_1)$ satisfies*

$$\rho(n, N; \mathcal{D}_1) = \begin{cases} \log_2 n + \Theta(1), & \text{when } N = 1, \\ \log_2 \log_2 n + \Theta(1), & \text{when } N = 2, \\ 0, & \text{when } N \geqslant 3. \end{cases}$$

  Theorem 7 shows that as the number of noisy reads increases, the optimal number of redundant bits required is gracefully reduced from $\log_2 n + \Theta(1)$ to $\log_2 \log_2 n + \Theta(1)$, and then to zero.

[1] When $P = 2n$, then any 2-periodic run is at most $n < P$. Hence, Condition (iii) is always true.

- In Section IV, we consider the case where $t \geqslant 2$ and show that if $|\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y})| = 1$, then $|\mathcal{D}_t(\boldsymbol{x}) \cap \mathcal{D}_t(\boldsymbol{y})| \leqslant D_{t-1}(n-1) + \nu_{t-2}(n-3)$. Hence, for the special case of $t = 2$, an $(n, 2; \mathcal{D}_1)$-reconstruction code can uniquely reconstruct codewords with $n+1$ distinct reads. By refining our arguments, we show that with appropriate choice of P, the constrained SVT codes from Theorem 4 can uniquely reconstruct codewords with less than $n+1$ distinct reads.

III. Lower Bound for $t = 1$ and $N = 2$

In this section, we provide a lower bound on the number of redundant bits of an $(n, 2; \mathcal{D}_1)$-reconstruction code $\mathcal{C}$, or equivalently, an upper bound on the size of $\mathcal{C}$. To this end, we borrow graph theoretic tools and consider the graph $\mathcal{G}(n)$ whose vertices correspond to $\{0,1\}^n$. The vertices $\boldsymbol{x}$ and $\boldsymbol{y}$ are adjacent if and only if $|\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y})| = 2$, or equivalently, $\boldsymbol{x}$ and $\boldsymbol{y}$ are Type-A-confusable.

Hence, $\mathcal{C}$ is an $(n, 2; \mathcal{D}_1)$-reconstruction code if and only if the corresponding set of vertices are independent in $\mathcal{G}(n)$.

Definition 8. A collection $\mathcal{Q}$ of cliques is a *clique cover* of $\mathcal{G}$ if every vertex in $\mathcal{G}$ belongs to some clique in $\mathcal{Q}$.

We have the following fact from graph theory (see for example, [22]).

Theorem 9. *If $\mathcal{Q}$ is a clique cover, then the size of any independent set is at most $|\mathcal{Q}|$.*

Hence, our objective is to construct a clique cover for $\mathcal{G}(n)$. To this end, we consider another parameter ℓ, and set $m = \lfloor n/(2\ell) \rfloor$ and $r = n - 2\ell m$. We divide each word of length n into m blocks of length 2ℓ and one block of length r. Set

$$\Lambda = \left\{(01)^j(10)^{\ell-j} : j \in [\ell]\right\} \cup \left\{(10)^j(01)^{\ell-j} : j \in [\ell]\right\}$$

where $(01)^0 = (10)^0$ is the empty word, and $\widetilde{\Lambda} = \{0,1\}^{2\ell} \setminus \Lambda$. So, $|\Lambda| = 2\ell$ and $|\widetilde{\Lambda}| = 2^{2\ell} - 2\ell$. To construct our clique cover $\mathcal{Q}(n, \ell)$, we consider two types of cliques. The first type of cliques are singletons of the form

$$S_{\boldsymbol{x}} = \{\boldsymbol{x}\}, \text{ where } \boldsymbol{x} \in \widetilde{\Lambda}^m \times \{0,1\}^r.$$

The second type of cliques are cliques of size ℓ. Here, we define

$$\Gamma = \left\{(\boldsymbol{u}, \boldsymbol{w}, i) : \boldsymbol{u} \in \widetilde{\Lambda}^{i-1}, \boldsymbol{w} \in \{0,1\}^{2\ell(m-i)+r}, i \in [m]\right\},$$

where the $\widetilde{\Lambda}^0 = \{0,1\}^0$ is the set containing an empty word. For each $\boldsymbol{z} = (\boldsymbol{u}, \boldsymbol{w}, i)$, we define two sets of vertices (which we later show to be cliques of size ℓ):

$$Q_{\boldsymbol{z}}^{(0)} = \left\{\boldsymbol{u}(01)^j(10)^{\ell-j}\boldsymbol{w} : j \in [\ell]\right\},$$
$$Q_{\boldsymbol{z}}^{(1)} = \left\{\boldsymbol{u}(10)^j(01)^{\ell-j}\boldsymbol{w} : j \in [\ell]\right\}.$$

We then define

$$\mathcal{Q}(n, \ell) = \left\{S_{\boldsymbol{x}} : \boldsymbol{x} \in \widetilde{\Lambda}^m \times \{0,1\}^r\right\} \cup \left\{Q_{\boldsymbol{z}}^{(0)}, Q_{\boldsymbol{z}}^{(1)} : \boldsymbol{z} \in \Gamma\right\}.$$

Lemma 10. *$\mathcal{Q}(n, \ell)$ is a clique cover for $\mathcal{G}(n)$.*

Proof. Clearly, all singletons are cliques. Next, we show that the ℓ-set $Q_{\boldsymbol{z}}^{(\mu)}$ is a clique for all $\boldsymbol{z} \in \Gamma$ and $\mu \in \{0,1\}$. We assume $\mu = 0$ and the proof for $\mu = 1$ is similar.

Let $\boldsymbol{x} = \boldsymbol{u}(01)^i(10)^{\ell-i}\boldsymbol{w}$ and $\boldsymbol{y} = \boldsymbol{u}(01)^j(10)^{\ell-j}\boldsymbol{w}$ be two words in $Q_{\boldsymbol{z}}^{(0)}$. Without loss of generality, let $i < j$. Then we can rewrite $\boldsymbol{x}$ and $\boldsymbol{y}$ as

$$\boldsymbol{x} = \boldsymbol{u}(01)^i(10)^{j-i}(10)^{\ell-j}\boldsymbol{w},$$
$$\boldsymbol{y} = \boldsymbol{u}(01)^i(01)^{j-i}(10)^{\ell-j}\boldsymbol{w}.$$

Thus, $\boldsymbol{x}$ and $\boldsymbol{y}$ are Type-A-confusable and so, $\boldsymbol{x}$ and $\boldsymbol{y}$ are adjacent in $\mathcal{G}(n)$. Therefore, $Q_{\boldsymbol{z}}^{(0)}$ is a clique.

It remains to show that any word $\boldsymbol{x} \in \{0,1\}^n$ belongs to some clique in $\mathcal{Q}(n, \ell)$. If $\boldsymbol{x} \in \widetilde{\Lambda}^m \times \{0,1\}^r$, then $\boldsymbol{x} \in S_{\boldsymbol{x}}$. Otherwise, $\boldsymbol{x} \notin \widetilde{\Lambda}^m \times \{0,1\}^r$ and one of the m subblocks of $\boldsymbol{x}$ belongs to Λ. Let the ith subblock be the first subblock from the left that belongs to Λ. Hence, this subblock is either of the form $(01)^j(10)^{\ell-j}$ or $(10)^j(01)^{\ell-j}$ for some $j \in [\ell]$. In the first case, $\boldsymbol{x}$ belongs to $Q_{(\boldsymbol{u},\boldsymbol{w},i)}^{(0)}$ where $\boldsymbol{u}$ is the first $(i-1)$ subblocks and $\boldsymbol{w}$ is the last $(m-i+1)$ subblocks. In the second case, $\boldsymbol{x}$ belongs to $Q_{(\boldsymbol{u},\boldsymbol{w},i)}^{(1)}$ where $\boldsymbol{u}$ and $\boldsymbol{w}$ are similarly defined. ■

Example 11. Set $\ell = 2$ and so, $\Lambda = \{0110, 0101, 1001, 1010\}$. When $n = 12$ and $m = 3$, a possible element $\boldsymbol{z}$ in Γ is the triple $(0000, 1000, 2)$ and the cliques corresponding to $\boldsymbol{z}$ are

$$Q_{\boldsymbol{z}}^{(0)} = \{000001101000, 000001011000\},$$
$$Q_{\boldsymbol{z}}^{(1)} = \{000010011000, 000010101000\}.$$

For general $n = 2m\ell$, since $|\widetilde{\Lambda}| = 12$, the number of singletons is 12^m. Furthermore, the number of ℓ-cliques is $2|\Gamma|$. Since the size of Γ is given by $\sum_{i=1}^m 12^{i-1}2^{4(m-i)} = 2^{n-2}(1-(3/4)^m)$, we have that the size of the clique cover $\mathcal{Q}(n, 2)$ is

$$2 \cdot \left(2^{n-2}(1-(3/4)^m)\right) + 12^m = 2^{n-1}(1+o(1)).$$

So, $\log|\mathcal{Q}(n,2)| = n-1+o(1)$ and an $(n, 2; \mathcal{D}_1)$-reconstruction code requires at least one redundant bit asymptotically. ■

To obtain the lower bound of $\log_2 \log_2 n - o(1)$ redundant bits, we refine our analysis by allowing ℓ to grow with n.

Now, we write $\lambda = |\widetilde{\Lambda}| = 2^{2\ell} - 2\ell$. Similar to the analysis in Example 11, we have the following lemma.

Lemma 12. *The size of $\mathcal{Q}(n, \ell)$ is given by*

$$2^n \left\{\left(1 - \frac{2\ell}{2^{2\ell}}\right)^{\lfloor \frac{n}{2\ell} \rfloor} + \frac{1}{\ell}\left(1 - \left(1 - \frac{2\ell}{2^{2\ell}}\right)^{\lfloor \frac{n}{2\ell} \rfloor}\right)\right\}.$$

Proof. Recall that $n = 2m\ell + r$, where $0 \leq r < 2\ell$. The number of singletons is $2^r\lambda^m$, while the number of ℓ-cliques is $2|\Gamma|$, where $|\Gamma| = \sum_{i=1}^m \lambda^{i-1}2^{2\ell(m-i)+r}$. Hence, the size of $\mathcal{Q}(n, \ell)$ is

$$2^r\lambda^m + 2\sum_{i=1}^m \lambda^{i-1}2^{2\ell(m-i)+r} = 2^r\lambda^m + 2^{n-2\ell+1}\frac{(\frac{\lambda}{2^{2\ell}})^m - 1}{\frac{\lambda}{2^{2\ell}} - 1}.$$

Straightforward manipulations then yield the lemma. ■

We set $\ell = \lceil \frac{1}{2}(1-\epsilon)\log_2 n \rceil$ where $0 < \epsilon < 1$ and write $f(n) = \left(1 - \frac{2\ell}{2^{2\ell}}\right)^{\lfloor n/(2\ell) \rfloor}$. Hence,

$$\log_2|\mathcal{Q}(n,\ell)| = n - \log_2 \ell + \log_2(1 + (\ell-1)f(n))$$
$$\leq n - \log_2 \ell + \log_2(1 + \ell f(n)).$$

Since $\log_2 \ell \geq \log_2 \log_2 n - O(1)$, it suffices to show that $\log_2(1 + \ell f(n)) = o(1)$.

Lemma 13. *We have that $\lim_{n\to\infty} \ell f(n) = 0$, or equivalently, $\lim_{n\to\infty} \ln(\ell f(n)) = -\infty$.*

Proof. First, we show that

$$\lim_{n\to\infty} \frac{\ln \ell}{\ln f(n)} = 0. \tag{4}$$

Note that for $0 < x < 1$, we have $|\ln(1-x)| \geq x$. Therefore,

$$\begin{aligned}
\lim_{n\to\infty} \left| \frac{\ln \ell}{\ln f(n)} \right| &= \lim_{n\to\infty} \left| \frac{\ln \ell}{\lfloor \frac{n}{2\ell} \rfloor \ln\left(1 - \frac{2\ell}{2^{2\ell}}\right)} \right| \\
&\leq \lim_{n\to\infty} \left| \frac{\ln \ell}{\left(\frac{n}{2\ell} - 1\right) \frac{2\ell}{2^{2\ell}}} \right| \\
&= \lim_{n\to\infty} \left| \frac{2^{2\ell} \ln \ell}{n - 2\ell} \right| \\
&\leq \lim_{n\to\infty} \left| \frac{2^{2+(1-\epsilon)\log_2 n} \ln \ell}{n - 2\ell} \right| \\
&= 4 \lim_{n\to\infty} \left| \frac{n \ln \ell}{n^{\epsilon}\,(n - 2\ell)} \right| = 0
\end{aligned}$$

which implies (4). Note that since $\lim_{n\to\infty} \ln \ell = \infty$, and $f(n) < 1$ for sufficiently large n, combined with (4), this implies that $\lim_{n\to\infty} \ln f(n) = -\infty$. Therefore, together with (4), we have the following:

$$\begin{aligned}
\lim_{n\to\infty} \ln(\ell f(n)) &= \lim_{n\to\infty} \ln \ell + \ln f(n) \\
&= \lim_{n\to\infty} (\ln f(n)) \left(1 + \frac{\ln \ell}{\ln f(n)}\right) \\
&= \lim_{n\to\infty} \ln f(n) \lim_{n\to\infty} \left(1 + \frac{\ln \ell}{\ln f(n)}\right) = -\infty. \blacksquare
\end{aligned}$$

Therefore, the results in this section can be summarized in following theorem.

Theorem 14. *Let $\mathcal{C}$ be an $(n, 2; \mathcal{D}_1)$-reconstruction code. For $\epsilon > 0$, we have that*

$$\log_2 |\mathcal{C}| \leqslant n - \log_2 \log_2 n + \log_2(1 - \epsilon) + o(1). \tag{5}$$

Therefore, $\rho(n, 2; \mathcal{D}_1) \geqslant \log_2 \log_2 n - O(1)$. Combining with Theorem 4, we have that $\rho(n, 2; \mathcal{D}_1) = \log_2 \log_2 n + \Theta(1)$.

In the arXiv version of [9], we generalize the construction in Theorem 4 for the nonbinary alphabet. A similar analysis can also demonstrate these nonbinary reconstruction codes are also asymptotically optimal. Details can be found in the arXiv version of this paper [23].

IV. Reconstruction Codes for $t \geqslant 2$ Deletions

In this section, we consider two words $\boldsymbol{x}$ and $\boldsymbol{y}$ whose single-deletion balls have intersection size one and study the size of intersection of their corresponding t-deletion balls for $t \geqslant 2$. We have the following result.

Theorem 15. *Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be binary words of length $n \geqslant 6$ and $t \geqslant 2$. If $|\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y})| = 1$, then we have that*

$$\begin{aligned}
|\mathcal{D}_t(\boldsymbol{x}) \cap \mathcal{D}_t(\boldsymbol{y})| &\leqslant D_{t-1}(n-1) + \nu_{t-1}(n-3) \\
&= n^{t-1} + O(n^{t-2}) \text{ for fixed values of } t.
\end{aligned} \tag{6}$$

Furthermore, when $t < n/2$, the inequality is strict.

Due to the space constraints, we omit the detailed proof of Theorem 15 and the interested reader may refer to the arXiv version of this paper [23].

Instead, we look at the implication of this theorem. Suppose that we have an $(n, 2; \mathcal{D}_1)$-reconstruction code $\mathcal{C}$. Then the intersection size of the single-deletion balls of any two codewords in $\mathcal{C}$ is at most one. Applying Theorems 2 and 15, we have that the read coverage $\nu(\mathcal{C}; \mathcal{D}_t)$ is at most $N'_t(n) \triangleq \max\left\{N_t^{(1)}(n), N_t^{(2)}(n)\right\}$ where $N_t^{(2)}(n) = D_{t-1}(n-1) + \nu_{t-1}(n-3)$. Hence, $\mathcal{C}$ is an $(n, N'_t(n)+1; \mathcal{D}_t)$-reconstruction code. For fixed values of t, we have that $N'_t(n) = N_t^{(2)}(n)$ for sufficiently large n and also observe that $N_t^{(2)}(n) \sim \nu_t(n)/2$, or, $\lim_{n\to\infty} N_t^{(2)}(n)/\nu_t(n) = 1/2$. Therefore, by sacrificing $\log_2 \log_2 n + O(1)$ bits of information, the codes in Theorem 4 are able to uniquely reconstruct codewords with half the number of noisy reads (as compared to no coding). Note also that by Theorem 2, if the number of redundancy is roughly $\log_2 n$, then the number of noisy reads has to be $2n^{t-2} + O(n^{t-3})$. We summarize our discussion with the following theorem.

Theorem 16. *Let $n \geqslant 6$ and $t \geqslant 2$. Set $N_t^{(2)}(n) = D_{t-1}(n-1) + \nu_{t-1}(n-3)$ and $N'_t(n) = \max\left\{N_t^{(1)}(n), N_t^{(2)}(n)\right\}$. If $\mathcal{C}$ is an $(n, 2; \mathcal{D}_1)$-reconstruction code, then $\mathcal{C}$ is also an $(n, N'_t(n)+1; \mathcal{D}_t)$-reconstruction code. Furthermore, when the value of t is fixed, this implies that $\rho\left(n, N_t^{(2)}(n)+1; \mathcal{D}_t\right) \leqslant \log_2 \log_2 n + O(1)$.*

When $t = 2$, we have that $N_2^{(2)}(n) = n+1$. In what follows, we focus on this special case and show that constrained SVT codes in Definition 3 are able to uniquely reconstruct codewords with strictly less than $n+1$ reads.

To do so, we require a stronger version of Theorem 15 for the case $t = 2$. The proof is omitted due to space constraints.

Theorem 17. *Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be words of length $n \geqslant 4$ that are Type-B-confusable. If $\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y}) = \{\boldsymbol{z}\}$ and then we have that*

$$\left| \left(\mathcal{D}_2(\boldsymbol{x}) \cap \mathcal{D}_2(\boldsymbol{y})\right) \setminus \mathcal{D}_1(\boldsymbol{z}) \right| \leqslant 2, \tag{7}$$

Next, we make the following simple observation.

Lemma 18. *Let $\boldsymbol{z}$ be a word of length n. If the length of any alternating run in a word $\boldsymbol{z}$ is at most P, then the number of runs in $\boldsymbol{z}$ is at most $n - \lceil n/P \rceil + 1$.*

Proof. Let $S = \{i \in \mathbb{Z} : \boldsymbol{z}_i = \boldsymbol{z}_{i+1}\}$. We order the elements of S and call them $s_1, s_2, ..., s_{|S|}$ from smallest to biggest. We want to show that $|S| \geq \lceil n/P \rceil - 1$. Note that $s_{i+1} - s_i \leq P$ for all $i \geq 1$, $s_1 \leq P$ and $s_{|S|} \geq n - P$, since otherwise there would be an alternating run of length more than P.

Suppose on the contrary that $|S| < \lceil n/P \rceil - 1$, this implies that $s_{|S|} = s_1 + \sum_{i=1}^{|S|-1} s_{i+1} - s_i \leq |S|P \leq (\lceil n/P \rceil - 2)\, P < (n/P + 1 - 2)P = n - P$, which contradicts that $s_{|S|} \geq n - P$. Therefore $|S| \geq \lceil n/P \rceil - 1$, and hence the number of runs in $\boldsymbol{z}$ is at most $n - \lceil n/P \rceil + 1$. $\blacksquare$

Recall that by design, the length of any alternating run of any codeword $\boldsymbol{x}$ in a constrained SVT code is at most P. Hence, the same property holds for any word $\boldsymbol{z}$ in the single-deletion ball of $\boldsymbol{x}$. So, we can apply Lemma 18 and provide a tighter bound on the size of $\mathcal{D}_1(\boldsymbol{z})$.

Proposition 19. *For any $c \in \mathbb{Z}_{1+P/2}$ and $d \in \mathbb{Z}_2$, the constrained SVT code $\mathcal{C}_{\mathrm{CSVT}}(n, P; c, d)$ is an $(n, N_P; \mathcal{D}_2)$-reconstruction code where $N_P = \max\{n - \lceil n/P \rceil + 4, 7\}$.*

Proof. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be distinct codewords. Then $|\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y})| \leqslant 1$ and it remains to show that $|\mathcal{D}_2(\boldsymbol{x}) \cap \mathcal{D}_2(\boldsymbol{y})| < N_P$.

If $|\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y})| = 0$, then Theorem 2 states that $|\mathcal{D}_2(\boldsymbol{x}) \cap \mathcal{D}_2(\boldsymbol{y})| \leqslant 6 < N_P$.

If $|\mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y})| = 1$, then let $\{\boldsymbol{z}\} = \mathcal{D}_1(\boldsymbol{x}) \cap \mathcal{D}_1(\boldsymbol{y})$. Since the length of any alternating run of $\boldsymbol{x}$ is at most P, by Lemma 18, the number of runs in $\boldsymbol{x}$ is at most $n - \lceil n/P \rceil + 1$. Since $\boldsymbol{z}$ is a subword of $\boldsymbol{x}$, the number of runs in $\boldsymbol{z}$ is at most the number of runs in $\boldsymbol{x}$, which is $n - \lceil n/P \rceil + 1$. Applying (7), we have that $|\mathcal{D}_2(\boldsymbol{x}) \cap \mathcal{D}_2(\boldsymbol{y})| = |\mathcal{D}_1(\boldsymbol{z})| + 2 \leq n - \lceil n/P \rceil + 3 < N_P$, as required. ∎

Let $P \geqslant 4$. It is well-known (see for example, [7]) that the number of length-n words whose 2-periodic run is at most P is $4F_{P-1}(n-2)$, where

$$F_\ell(n) = \begin{cases} 2^n, & \text{if } 0 \leqslant n \leqslant \ell - 1, \\ \sum_{i=1}^{\ell} F_\ell(n-i), & \text{otherwise.} \end{cases}$$

Hence, we have the following lower bound on the size.

Corollary 20. *For $P \geqslant 4$, set $N_P = \max\{n - \lceil n/P \rceil + 4, 7\}$. Then there exists an $(n, N_P; \mathcal{D}_2)$-reconstruction code of size at least $4F_{P-1}(n-2)/(P+2)$.*

To conclude, for codelengths $n \in \{127, 255, 1023\}$, we vary the parameter P in the constrained SVT codes and compute the corresponding values of N_P and redundancy. The numerical results are given in Table I. As expected, as we decrease the value of P, the number of required reads also decreases. However, the number of redundant bits also increases significantly and in this case (where P is small), the VT code uses significantly less redundant bits. For completeness, we list the values of read-coverage and redundancy of a VT code of length n and the space $\{0,1\}^n$ (corresponding to the uncoded case).

n	P	Read Coverage	Redundancy	Remarks
127	–	6	7.00	VT code
127	6	108	6.016	–
127	8	114	4.018	–
127	10	117	3.753	–
127	–	250	0.00	$\{0,1\}^n$
255	–	6	8.00	VT code
255	8	226	4.762	–
255	10	232	3.935	–
255	12	236	3.894	–
255	–	506	0.00	$\{0,1\}^n$
1023	–	6	10.00	VT code
1023	8	898	9.22	–
1023	10	923	5.03	–
1023	12	940	4.17	–
1023	14	952	4.09	–
1023	–	2042	0.00	$\{0,1\}^n$

TABLE I: List of constrained SVT codes and their read coverage and redundancy

REFERENCES

[1] G. M. Church, Y. Gao, and S. Kosuri. "Next-generation digital information storage in DNA," *Science*, 337(6102):1628–1628, 2012.

[2] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney. "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, 494:77–80, 2013.

[3] S. Yazdi, H. M. Kiah, E. R. Garcia, J. Ma, H. Zhao, and O. Milenkovic. DNA-based storage: Trends and methods. *IEEE Trans. Molecular, Biological, Multi-Scale Commun.*, 1(3):230–248, 2015.

[4] L. Organick, S. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Racz, G. Kamath, P. Gopalan, B. Nguyen, C. Takahashi, S. Newman, H.-Y. Parker, C. Rashtchian, K. Stewart, G. Gupta, R. Carlson, J. Mulligan, D. Carmean, G. Seelig, L. Ceze, and K. Strauss. Random access in large-scale DNA data storage. *Nature Biotechnology*, 36(3), 242, 2018.

[5] S. S. Parkin, M. Hayashi, and L. Thomas, "Magnetic domain-wall racetrack memory," *Science*, vol. 320, pp. 190–194, 2008.

[6] Y. Zhang, C. Zhang, J. Nan, Z. Zhang, X. Zhang, J.-O. Klein, D. Ravelosona, G. Sun, and W. Zhao. "Perspectives of racetrack memory for large-capacity on-chip memory: From device to system," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 63, no. 5, pp. 629–638, 2016.

[7] Y. M. Chee, H. M. Kiah, A. Vardy, E. Yaakobi, and V. K. Vu. "Coding for racetrack memories," *IEEE Trans. on Information Theory*, 2018.

[8] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Trans. on Information Theory*, 47(1), pp. 2–22, 2001.

[9] H. M. Kiah, T. T. Nguyen and E. Yaakobi, "Coding for Sequence Reconstruction for Single Edits," In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Accepted Mar 2020. (*arXiv preprint arxiv:2001.01376)*

[10] M. Cheraghchi, R. Gabrys, O. Milenkovic and J. Ribeiro, "Coded trace reconstruction," *arXiv preprint arxiv:1903.09992*, 2019

[11] R. Gabrys, and E. Yaakobi. "Sequence reconstruction over the deletion channel," *IEEE Trans. on Information Theory*, 64(4), pp.2924-2931, 2018.

[12] E. Konstantinova, "On reconstruction of signed permutations distorted by reversal errors," *Discrete Math.*, vol. 308, pp. 974–984, Mar. 2008.

[13] V. I. Levenshtein and J. Siemons, "Error graphs and the reconstruction of elements in groups," *J. Combinat. Theory, A*, vol. 116, no. 4, pp. 795–815, 2009.

[14] F. Sala, R. Gabrys, C. Schoeny, and L. Dolecek, "Exact reconstruction from insertions in synchronization codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2428–2445, Apr. 2017.

[15] Y. Yehezkeally and M. Schwartz. "Reconstruction codes for DNA sequences with uniform tandem-duplication errors," In *Information Theory (ISIT), 2018 IEEE International Symposium on*, pages 2535–2539. IEEE, 2018.

[16] M. Abu Sini, and E. Yaakobi, "Reconstruction of Sequences in DNA Storage". In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019.

[17] V. Junnnila, T. Laihonen, and T. Lehtila, "The Levenshtein's channel and the list size in information retrieval" In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019.

[18] V. I. Levenshtein, "Efficient Reconstruction of Sequences from Their Subsequences or Supersequences," *Journal of Combinatorial Theory, Series A*, 93, pp. 310–332, 2001.

[19] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[20] C. Schoeny, A. Wachter-Zeh, R. Gabrys, and E. Yaakobi. "Codes correcting a burst of deletions or insertions." *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 1971–1985, 2017.

[21] Y. Liron and M. Langberg. "A Characterization of the Number of Subsequences Obtained via the Deletion Channel" *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2300–2312, 2015.

[22] D. E. Knuth. "The sandwich theorem". *The Electronic Journal of Combinatorics*, A1, 1994.

[23] J. Chrisnata, H. M. Kiah, and E. Yaakobi, "Coding for Sequence Reconstruction for Single Edits," *arXiv preprint arxiv:2004.06032*, 2020