





Error-Correcting Codes for Nanopore Sequencing

Anisha Banerjee , *Student Member, IEEE*, Yonatan Yehezkeally , *Member, IEEE*,
 Antonia Wachter-Zeh , *Senior Member, IEEE*, and Eitan Yaakobi , *Senior Member, IEEE*

Abstract—Nanopore sequencing, superior to other sequencing technologies for DNA storage in multiple aspects, has recently attracted considerable attention. Its high error rates, however, demand thorough research on practical and efficient coding schemes to enable accurate recovery of stored data. To this end, we consider a simplified model of a nanopore sequencer inspired by Mao *et al.*, incorporating intersymbol interference and measurement noise. Essentially, our channel model passes a sliding window of length ℓ over a q -ary input sequence that outputs the *composition* of the enclosed ℓ bits, and shifts by δ positions with each time step. In this context, the composition of a q -ary vector x specifies the number of occurrences in x of each symbol in $\{0, 1, \dots, q - 1\}$. The resulting compositions vector, termed the *read vector*, may also be corrupted by t substitution errors. By employing graph-theoretic techniques, we deduce that for $\delta = 1$, at least $\log \log n$ symbols of redundancy are required to correct a single ($t = 1$) substitution. Finally, for $\ell \geq 3$, we exploit some inherent characteristics of read vectors to arrive at an error-correcting code that is of optimal redundancy up to a (small) additive constant for this setting. This construction is also found to be optimal for the case of reconstruction from two noisy read vectors.

Index Terms—Sequence reconstruction, DNA sequences, nanopore sequencing, error-correction codes, composition errors

Manuscript received 2 October 2023; revised 8 March 2024; accepted 18 March 2024. This material is based upon work supported by the National Science Foundation under Grant No. CCF 2212437. This work has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant agreement No. 801434). It was also funded by the European Union (ERC, DNAStorage, 865630). Additionally, this project has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement No. 101115134. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. The work of Yonatan Yehezkeally was supported by the Alexander von Humboldt Foundation under a Carl Friedrich von Siemens Post-Doctoral Research Fellowship. An earlier version of this paper was presented in part at the 2023 IEEE International Symposium on Information Theory (ISIT) [DOI: 10.1109/ISIT54713.2023.10206710]. (*Corresponding author: Anisha Banerjee.*)

Anisha Banerjee, Yonatan Yehezkeally, and Antonia Wachter-Zeh are with the Institute for Communications Engineering, School of Computation, Information and Technology, Technical University of Munich (TUM), 80333 Munich, Germany (e-mail: anisha.banerjee@tum.de; yonatan.yehezkeally@tum.de; antonia.wachter-zeh@tum.de). Eitan Yaakobi is with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel (e-mail: yaakobi@cs.technion.ac.il).

Copyright (c) 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

I. INTRODUCTION

The advent of DNA storage as an encouraging solution to our ever-increasing storage requirements has spurred significant research to develop superior synthesis and sequencing technologies. Among the latter, nanopore sequencing [1]–[3] appears to be a strong contender due to low cost, better portability, and support for longer reads. In particular, this sequencing process comprises transmuting a DNA fragment through a microscopic pore that holds ℓ nucleotides at each time instant and measuring the variations in the ionic current, which are influenced by the different nucleotides passing through. However, due to the physical aspects of this process, multiple kinds of distortions corrupt the readout. Firstly, the simultaneous presence of $\ell > 1$ nucleotides in the pore makes the observed current dependent on multiple nucleotides instead of just one, thus causing inter-symbol interference (ISI). Next, the passage of the DNA fragment through the pore is often irregular and may involve backtracking or skipping a few nucleotides, thereby leading to duplications or deletions. Furthermore, the measured current is accompanied by random noise, which might result in substitution errors.

Several attempts have been made to develop a faithful mathematical model for the nanopore sequencer. In particular, [4] proposed a channel model that embodies the effects of ISI, deletions, and random noise while establishing upper bounds on the capacity of this channel. The authors of [5] focused on a more deterministic model incorporating ISI and developed an algorithm to compute its capacity. Efficient coding schemes for this abstracted channel were also suggested. More recently, a finite-state Markov channel (FSMC)-based approach was adopted to formulate a model that accounts for ISI, duplications, and noisy measurements [6].

In this work, we adopt a specific variation of the model proposed in [4], which is also interesting owing to its resemblance with the transverse-read channel [7], which is relevant to racetrack memories. Expressly, we represent the process of nanopore sequencing as the concatenation of three channels, as depicted in Fig. 1. We may view the first stage as a sliding window of size ℓ passing through an input sequence and shifting by δ positions after each time instant, thereby producing a sequence of strings of ℓ consecutive symbols, or ℓ -mers. This component is parameterized by (ℓ, δ) and models the ISI effect, i.e., it reflects the dependence of the current variations on the ℓ consecutive nucleotides in the pore at any given time. Next, a memoryless channel converts this sequence of ℓ -mers into a sequence of discrete voltage levels according to a deterministic function, specifically the *composition*. (Note that this model for the DMC does not entertain the possibility that mapping of the ℓ -mers to voltage levels might depend on

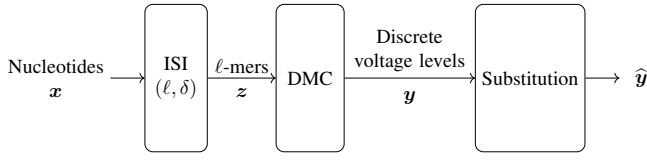


Fig. 1. Simplified model of a nanopore sequencer

the order of the bases in the nanopore.) Finally, the substitution channel captures the effect of random noise by introducing possible substitution errors into the sequence of voltage levels.

This work aims to design efficient error-correcting codes for nanopore sequencing. More specifically, as a starting point for future analysis, the channel mentioned above model is treated where at most one substitution occurs and $\delta = 1$. The problem is stated more formally as follows.

Let $\mathcal{R}_{\ell, \delta}(\mathbf{x})$ represent the channel output for an input $\mathbf{x} \in \Sigma_q^n$, given that no substitution affected the ℓ -mers. Now we seek to find a code $\mathcal{C} \subseteq \Sigma_q^n$ such that for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$, the Hamming distance between $\mathcal{R}_{\ell, \delta}(\mathbf{x}_1)$ and $\mathcal{R}_{\ell, \delta}(\mathbf{x}_2)$ strictly exceeds 2. In other words, one can uniquely deduce the channel input despite ISI and the subsequent occurrence of at most one substitution, provided it belongs to the code \mathcal{C} .

The rest of the manuscript is organized as follows. We establish relevant notation and terminology while discussing the underlying properties of read vectors in Section II. The results that follow, hold for all $(\ell, 1)$ -read vectors, where $\ell \geq 3$. In Section III, we employ graph-theoretic techniques from [8] to determine the minimum redundancy required by any code that corrects a single substitution error in an $(\ell, 1)$ -read vector. Section IV describes a redundancy-optimal instantiation of such a code. Subsequently, owing to the particular applicability of the reconstruction schema to DNA-based storage [9]–[15] in Section III-C, we find that this instantiation is also redundancy-optimal when reconstructing \mathbf{x} from two distinct noisy copies of $\mathcal{R}_{\ell, \delta}(\mathbf{x})$, each of which has suffered at most 1 substitution. Concluding remarks concerning future work are offered in Section V.

II. PRELIMINARIES

A. Notations and Terminology

In the following, we let Σ_q indicate the q -ary alphabet $\{0, 1, \dots, q-1\}$. Additionally, $[n]$ is used to denote the set $\{1, 2, \dots, n\}$. All uses of the log operator consider base q . Element-wise modulo operation on a vector, say $\mathbf{y} \in \Sigma_q^n$, is represented as

$$\mathbf{y} \bmod a \triangleq (y_1 \bmod a, y_2 \bmod a, \dots, y_n \bmod a). \quad (1)$$

For any vector $\mathbf{x} = (x_1, \dots, x_n)$, we refer to its substring $(x_i, x_{i+1}, \dots, x_j)$ as \mathbf{x}_i^j . The composition of a vector \mathbf{x} is denoted by $c(\mathbf{x}) \triangleq 0^{i_0} \dots (q-1)^{i_{q-1}}$, such that \mathbf{x} contains i_0 ‘0’s, i_1 ‘1’s and so on. We also define the L_1 -weight of the composition $c(\mathbf{x})$ as $|c(\mathbf{x})|_1 \triangleq i_1 + 2i_2 + \dots + (q-1)i_{q-1} = |\mathbf{x}|_1$. This operator may also be applied to a vector of compositions in the same spirit as in (1). By abuse of notation, when n is known from the context, we omit from $c(\mathbf{x})$ any symbol $x \in \Sigma_q$ such

that $i_x = 0$. Further, when convenient, we treat $c(\mathbf{x})$ as a formal monomial by using expressions of the form $c(\mathbf{x}) \cdot (c(\mathbf{y}))^{-1}$, and allow formal cancellations of the form, e.g., $0^{i_0} 1^{i_1} \cdot (0^{j_0} 1^{j_1})^{-1} = 0^{i_0} 1^{i_1} \cdot 0^{-j_0} 1^{-j_1} = 0^{i_0-j_0} 1^{i_1-j_1}$.

We also extensively use the Hamming distance, which is defined for any two vectors $\mathbf{x}, \mathbf{y} \in \Sigma^n$, for any alphabet Σ , as

$$d_H(\mathbf{x}, \mathbf{y}) = |\{i : i \in [n], x_i \neq y_i\}|.$$

Throughout this paper, we assume existence of integers n, ℓ , and δ that satisfy the relation $n + \ell \equiv 0 \pmod{\delta}$. The explicit definition of the channel output is now laid out as follows.

Definition 1. The (ℓ, δ) -read vector of any $\mathbf{x} \in \Sigma_q^n$ is of length $(n + \ell)/\delta - 1$ and is denoted by

$$\mathcal{R}_{\ell, \delta}(\mathbf{x}) \triangleq (c(\mathbf{x}_{\delta-\ell+1}^\delta), c(\mathbf{x}_{2\delta-\ell+1}^{2\delta}), \dots, c(\mathbf{x}_{n-\delta+1}^{n-\delta})),$$

where for brevity of notation we let $x_i = \phi$ for any $i < 1$ or $i > n$, i.e., a null element such that $c(\mathbf{y} \circ \phi) = c(\mathbf{y})$. $\mathcal{R}_{\ell, \delta}(\mathbf{x})_i$ is used to denote the i -th element of $\mathcal{R}_{\ell, \delta}(\mathbf{x})$, i.e., $\mathcal{R}_{\ell, \delta}(\mathbf{x})_i = c(\mathbf{x}_{i\delta-\ell+1}^{i\delta}) = c(\mathbf{x}_{\max(1, i\delta-\ell+1)}^{\min(i\delta, n)})$.

Remark: The above definition of an (ℓ, δ) -read vector appears similar to that of the (ℓ, δ) -transverse-read vector introduced in [7], except that the L_1 -weights are replaced by compositions and $\mathcal{R}_{\ell, \delta}(\mathbf{x})$ begins and ends with the compositions of substrings \mathbf{x}_1^δ and $\mathbf{x}_{n-\delta+1}^{n-\delta}$ respectively, even though its intermediate elements signify compositions of length- ℓ substrings. This is motivated by obtaining a current reading even when the DNA strand has only partially entered the nanopore as demonstrated in Figure 2.

In the following, we introduce some technical notation which will play an instrumental role in demonstrating the key properties of read vectors, namely the notion of derivatives and sub-derivatives of read vectors.

Definition 2. Let $\mathcal{R} = (c_1, \dots, c_k)$ where for each $1 \leq i \leq k$, c_k is a composition of some vector in Σ_q^ℓ . Then the derivative of \mathcal{R} is the length- $(k+1)$ formal-monomial-vector defined as

$$\Delta \triangleq (c_1 c_0^{-1}, c_2 c_1^{-1}, \dots, c_{k+1} c_k^{-1}),$$

where $c_0, c_{k+1} = \phi$ are included for uniformity. Observe that the differentiation $\mathcal{R} \mapsto \Delta$ is invertible.

Example 1. Consider $\mathbf{x} = (1, 2, 0, 1, 2, 2)$. As we demonstrate in Fig. 2, the $(3, 1)$ -read vector of \mathbf{x} is thus $\mathcal{R}_{3,1}(\mathbf{x}) = (1, 12, 012, 012, 012, 12^2, 2^2, 2)$. Evidently, $\mathcal{R}_{3,1}(\mathbf{x})_3 = 012$. The derivative of this read vector, as illustrated in Fig. 3, is $\Delta = (1, 2, 0, \phi, \phi, 20^{-1}, 1^{-1}, 2^{-1})$.

Definition 3. For any ℓ, δ where $\ell \geq \delta$, the α -th read sub-derivative, is used to indicate a specific subsequence of the derivative of $\mathcal{R}_{\ell, \delta}(\mathbf{x})$, and is defined for any $\alpha \in \{0, 1, \dots, \lfloor \frac{\ell}{\delta} \rfloor - 1\}$ as

$$\begin{aligned} \Delta_{\ell, \delta}^\alpha(\mathbf{x}) &\triangleq (\mathcal{R}(\mathbf{x})_{\alpha+1} \cdot \mathcal{R}(\mathbf{x})_\alpha^{-1}, \mathcal{R}(\mathbf{x})_{\alpha+\lfloor \frac{\ell}{\delta} \rfloor+1} \cdot \mathcal{R}(\mathbf{x})_{\alpha+\lfloor \frac{\ell}{\delta} \rfloor}^{-1}, \\ &\dots, \mathcal{R}(\mathbf{x})_{\alpha+k\lfloor \frac{\ell}{\delta} \rfloor+1} \cdot \mathcal{R}(\mathbf{x})_{\alpha+k\lfloor \frac{\ell}{\delta} \rfloor}^{-1}) \\ &= (c(\mathbf{x}_{\alpha\delta+1}^{(\alpha+1)\delta}) \cdot c(\mathbf{x}_{\alpha\delta-\ell+1}^{(\alpha+1)\delta-\ell})^{-1}, \\ &\dots, c(\mathbf{x}_{(\alpha+k\lfloor \frac{\ell}{\delta} \rfloor+1)\delta}^{(\alpha+k\lfloor \frac{\ell}{\delta} \rfloor+1)\delta}) \cdot c(\mathbf{x}_{(\alpha+k\lfloor \frac{\ell}{\delta} \rfloor)\delta-\ell+1}^{(\alpha+k\lfloor \frac{\ell}{\delta} \rfloor)\delta-\ell})^{-1}), \end{aligned}$$

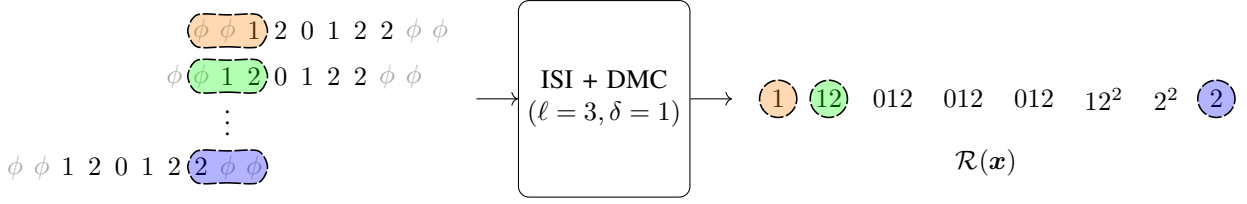


Fig. 2. Demonstration of Example 1 for the ISI (sliding window) + DMC (composition function) channel. Notice the leading and trailing $(\ell - 1) = 2$ ϕ s (marked in gray) are not a part of the input vector \mathbf{x} .

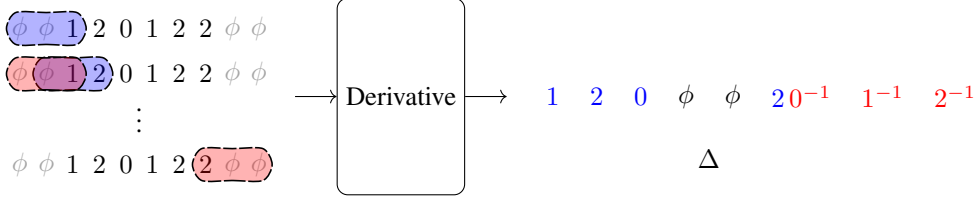


Fig. 3. Demonstration of a derivative of a vector of compositions (Definition 2) in Example 1. On the left, the compositions with positive exponents have been marked in blue, while those with negative exponents were marked in red.

where $k = \lfloor \frac{n+\ell-(\alpha+1)\delta}{\delta \lfloor \ell/\delta \rfloor} \rfloor$ and for brevity of notation we let $\mathcal{R}(\mathbf{x})_p = \phi$ and $x_m = \phi$ for any $p \notin \lfloor \frac{n+\ell-\delta}{\delta} \rfloor$ and $m \notin [n]$. We let $\Delta_{\ell,\delta}^\alpha(\mathbf{x})_i$ indicate the i -th element of $\Delta_{\ell,\delta}^\alpha(\mathbf{x})$, i.e.,

$$\Delta_{\ell,\delta}^\alpha(\mathbf{x})_i = \mathcal{R}(\mathbf{x})_{\alpha+(i-1)\lfloor \frac{\ell}{\delta} \rfloor+1} \cdot \mathcal{R}(\mathbf{x})_{\alpha+(i-1)\lfloor \frac{\ell}{\delta} \rfloor}^{-1}$$

Example 2. Reconsidering $\mathbf{x} = (1, 2, 0, 1, 2, 2)$ from Example 1, we note that $\Delta_{3,1}^0(\mathbf{x}) = (1, \phi, 1^{-1})$, $\Delta_{3,1}^1(\mathbf{x}) = (2, \phi, 2^{-1})$ and $\Delta_{3,1}^2(\mathbf{x}) = (0, 0^{-1}2, 2^{-1})$. Observe that when interleaved together, these sub-derivatives compose Δ (Figure 3).

When clear from the context, ℓ and δ will be removed from the preceding notations.

As mentioned earlier, [7] investigated a similar model designated as the transverse-read channel in connection with racetrack memories. Therein, the information limit of this channel was derived for different parameters, and several codes enabling unique reconstruction were proposed. Certain error-correcting codes were also presented for $\ell = 2$ and $\delta = 1$.

B. Properties of the Read Vectors

A closer look at the definitions in the last section reveals that not every vector of ℓ -compositions represents the read vector of some $\mathbf{x} \in \Sigma_q^n$, i.e., is *valid*. In this section, we first observe which vectors are valid and deduce specific properties that often enable us to detect errors, thereby assisting in designing error-correcting constructions of improved redundancies. To lucidly introduce these properties, we require the following notation. We will see in Lemma 1 that these allow us to determine precisely the set of valid read-vectors.

Definition 4. For $\mathbf{x} \in \Sigma_q^n$ and any $\alpha \in \{0, 1, \dots, \lfloor \frac{\ell}{\delta} \rfloor - 1\}$, let

$$C_{\ell,\delta}^\alpha(\mathbf{x}) \triangleq (c(\mathbf{x}_{\alpha\delta+1}^{(\alpha+1)\delta}), \dots, c(\mathbf{x}_{(\alpha+k\lfloor \frac{\ell}{\delta} \rfloor)\delta+1}^{(\alpha+k\lfloor \frac{\ell}{\delta} \rfloor+1)\delta})),$$

where $k = \lfloor \frac{n+\ell-(\alpha+1)\delta}{\delta \lfloor \ell/\delta \rfloor} \rfloor$, be a sequence of compositions.

Observe that for $\delta = 1$ each $C_{\ell,\delta}^\alpha(\mathbf{x})$ is a subsequence of \mathbf{x} , composed of the positions at indices $i \equiv \alpha + 1 \pmod{\lfloor \frac{\ell}{\delta} \rfloor}$,

and in particular there exists a bijection between Σ^n and the set of $\lfloor \frac{\ell}{\delta} \rfloor$ -tuple of length- $(k+1)$ vectors.

Example 3. Reconsidering $\mathbf{x} = (1, 2, 0, 1, 2, 2)$ from Example 1, we observe that $C_{3,1}^0(\mathbf{x}) = (1, 1)$, $C_{3,1}^1(\mathbf{x}) = (2, 2)$ and $C_{3,1}^2(\mathbf{x}) = (0, 2)$, which are evidently subsequences of \mathbf{x} . Under $\delta > 1$, these transform into composition vectors; for instance $C_{4,2}^0(\mathbf{x}) = (12, 2^2)$ and $C_{4,2}^1(\mathbf{x})(\mathbf{x}) = (01)$.

We now employ Definition 4 to state a necessary and sufficient condition for the existence of a unique $\mathbf{x} \in \Sigma_q^n$ that corresponds to a given vector of compositions for ℓ, δ , satisfying $\ell \bmod \delta = 0$.

Lemma 1. Take ℓ, δ satisfying $\ell \equiv 0 \pmod{\delta}$, and let $\{C_\alpha : \alpha \in \{0, 1, \dots, \frac{\ell}{\delta} - 1\}\}$ be any (ℓ/δ) arbitrary vectors of compositions, each belonging to vectors in Σ_q^δ , such that the length of C_α is $(\lfloor \frac{n+\ell-(\alpha+1)\delta}{\delta} \rfloor + 1)$. Let their respective derivatives be $\{\Delta_\alpha : \alpha \in \{0, 1, \dots, \frac{\ell}{\delta} - 1\}\}$. Then there exists $\mathbf{x} \in \Sigma_q^n$ such that $\Delta_\alpha = \Delta^\alpha(\mathbf{x})$ and $C_\alpha = C_{\ell,\delta}^\alpha(\mathbf{x})$, for all $\alpha \in \{0, 1, \dots, \frac{\ell}{\delta} - 1\}$. Further, when $\delta = 1$ this \mathbf{x} is unique.

Proof: Recall that owing to $x_i = \phi$ for all $i \notin [n]$, the following holds for any $\alpha \in \{0, 1, \dots, \frac{\ell}{\delta} - 1\}$.

$$\Delta^\alpha(\mathbf{x}) = (c(\mathbf{x}_{\alpha\delta+1}^{(\alpha+1)\delta}), c(\mathbf{x}_{\alpha\delta+\ell+1}^{(\alpha+1)\delta+\ell}) \cdot c(\mathbf{x}_{\alpha\delta+1}^{(\alpha+1)\delta})^{-1}, \dots, c(\mathbf{x}_{\alpha\delta+k\ell+1}^{(\alpha+1)\delta+k\ell}) c(\mathbf{x}_{\alpha\delta+(k-1)\ell+1}^{(\alpha+1)\delta+(k-1)\ell})^{-1}, c(\mathbf{x}_{\alpha\delta+k\ell+1}^{(\alpha+1)\delta+k\ell})^{-1}),$$

where $k = \lfloor \frac{n-(\alpha+1)\delta}{\delta} \rfloor + 1$. Evidently, by left-to-right (or right-to-left) reconstruction, we observe that $C_{\ell,\delta}^\alpha(\mathbf{x})$ can be uniquely deduced from $\Delta^\alpha(\mathbf{x})$. The other direction follows from the observation that $\Delta^\alpha(\mathbf{x})$ is essentially the derivative of $C_{\ell,\delta}^\alpha(\mathbf{x})$, in accordance with Definition 2. \blacksquare

Corollary 1. If $\ell \equiv 0 \pmod{\delta}$, then for any $\mathbf{x} \in \Sigma_q^n$ and $\alpha \in \{0, 1, \dots, \frac{\ell}{\delta} - 1\}$, the cumulative product of the first $m+1$ elements of $\Delta^\alpha(\mathbf{x})$ is $c(\mathbf{x}_{m+\alpha\delta+1}^{m\ell+(\alpha+1)\delta})^1$. Thus, $\Delta^\alpha(\mathbf{x})$

¹Analogous result exists for sum of last $m+1$ elements.

determines $C_{\ell,\delta}^\alpha(\mathbf{x})$, which in the special case of $\delta = 1$, is effectively $(x_{\alpha+1}, x_{\alpha+\ell+1}, \dots)$.

Since $\mathcal{R}(\mathbf{x})$ is in bijection with the set $\{\Delta^\alpha(\mathbf{x})\}_{\alpha \in \{0,1,\dots, \lfloor \frac{\ell}{\delta} \rfloor - 1\}}$, it follows that when $\ell \equiv 0 \pmod{\delta}$ (and, in particular, when $\delta = 1$) the set of valid read vectors is isomorphic to the set of $\lfloor \frac{\ell}{\delta} \rfloor$ -tuple of appropriately-long composition-vectors.

Corollary 2. *For $\delta = 1$ and any $\mathbf{x} \in \Sigma_q^n$, let $R(\mathbf{x})$ be either $\mathcal{R}(\mathbf{x})$ or $|\mathcal{R}(\mathbf{x})|_1 \bmod q$. Then $\mathbf{x} \in \Sigma_q^n$, \mathbf{x}_i^j can be uniquely determined, either from*

- 1) $\mathbf{x}_{i-\ell+1}^{i-1}$ and $(R(\mathbf{x})_i, R(\mathbf{x})_{i+1}, \dots, R(\mathbf{x})_j)$; or
- 2) $\mathbf{x}_{j+1}^{j+\ell-1}$ and $(R(\mathbf{x})_{i+\ell-1}, R(\mathbf{x})_{i+\ell}, \dots, R(\mathbf{x})_{j+\ell-1})$,

where for all $k \notin [n]$, $x_k = \phi$. Since for $p \in \{1, n\}$, $x_p = R(\mathbf{x})_p$, the first or last n elements of $R(\mathbf{x})$ suffice to reconstruct \mathbf{x} .

Proof: We restrict our attention to $R(\mathbf{x}) = |\mathcal{R}(\mathbf{x})|_1 \bmod q$ since it can readily be obtained from $\mathcal{R}(\mathbf{x})$ itself. By successively applying the fact that for $i \leq p \leq j$, one can recover x_p from the combined knowledge of $\mathbf{x}_{p-\ell+1}^{p-1}$ and $|\mathcal{R}_{\ell,1}(\mathbf{x})_p|_1 \bmod q = (\sum_{h=p-\ell+1}^p x_h) \bmod q$, we arrive at the statement of the corollary. The same argument also holds for right-to-left reconstruction. ■

Example 4. We reconsider $\mathcal{R}(\mathbf{x}) = (1, 12, 012, 012, 012, 12^2, 2^2, 2)$ from Example 1 and now wish to reconstruct \mathbf{x} from it. Recall that $\ell = 3$ and $\delta = 1$. Firstly, we observe that $x_1 = \mathcal{R}(\mathbf{x})_1 = 1$. Next, $c(\mathbf{x}_1^2) = \mathcal{R}(\mathbf{x})_2 = 12$, implying $x_2 = 2$. Such a left-to-right reconstruction of $\mathcal{R}(\mathbf{x})$ leads us to $\mathbf{x} = (1, 2, 0, 1, 2, 2)$, as in Example 1. Similarly, when given $|\mathcal{R}(\mathbf{x})|_1 \bmod 3 = (1, 0, 0, 0, 0, 2, 1, 2)$, one can infer from Definition 1 that $x_1 = |\mathcal{R}_{3,1}(\mathbf{x})_1|_1 \bmod 3 = 1$, $(x_1 + x_2) \bmod 3 = |\mathcal{R}(\mathbf{x})_2|_1 \bmod 3$ and so on, thereby leading to $\mathbf{x} = (1, 2, 0, 1, 2, 2)$ once again. Right-to-left reconstruction will yield the same result.

For ℓ, δ satisfying $\ell \equiv 0 \pmod{\delta}$ and $\mathbf{x} \in \Sigma_q^n$, $\mathcal{R}(\mathbf{x})$ and each $\Delta \in \{\Delta^\alpha(\mathbf{x})\}_{\alpha \in \{0,1,\dots, \frac{\ell}{\delta} - 1\}}$ bear some useful properties that assist us in the design of error-correcting constructions, regarding the products over a read vector and a read sub-derivative.

Lemma 2. *For any ℓ, δ such that $\ell \equiv 0 \pmod{\delta}$, and all $\mathbf{x} \in \Sigma_q^n$, it holds that $\prod_{i=1}^{\frac{n+\ell}{\delta}-1} \mathcal{R}(\mathbf{x})_i = (c(\mathbf{x}))^{\ell/\delta}$. Further, $\prod_{i=1}^{\lfloor \frac{n-(\alpha+1)\delta}{\ell} \rfloor + 2} \Delta^\alpha(\mathbf{x})_i = c(\phi)$ for any $\alpha \in \{0, 1, \dots, \frac{\ell}{\delta} - 1\}$.*

Proof: Observe that for all $\alpha \in \{0, 1, \dots, \frac{\ell}{\delta} - 1\}$, we have

$$\prod_{i=0}^{\lfloor \frac{n-(\alpha+2)\delta}{\ell} \rfloor + 1} \mathcal{R}(\mathbf{x})_{\alpha+i\frac{\ell}{\delta}+1} = c(\mathbf{x}). \quad (2)$$

This naturally leads us to

$$\prod_{i=1}^{\frac{n+\ell}{\delta}-1} \mathcal{R}(\mathbf{x})_i = \prod_{\alpha=0}^{\ell/\delta-1} \prod_{i=0}^{\lfloor \frac{n-(\alpha+2)\delta}{\ell} \rfloor + 1} \mathcal{R}(\mathbf{x})_{\alpha+i\frac{\ell}{\delta}+1} = c(\mathbf{x})^{\ell/\delta}.$$

While one can arrive at $\prod_{i=1}^{\lfloor \frac{n-(\alpha+1)\delta}{\ell} \rfloor + 2} \Delta^\alpha(\mathbf{x})_i = c(\phi)$ directly from the definition, we may also use (2) to prove this as follows, denoting $k \triangleq \lfloor \frac{n-(\alpha+1)\delta}{\ell} \rfloor + 1$ for simplicity.

$$\begin{aligned} \prod_{i=1}^{k+1} \Delta^\alpha(\mathbf{x})_i &= \prod_{i=0}^k \mathcal{R}(\mathbf{x})_{\alpha+i\frac{\ell}{\delta}+1} \cdot \mathcal{R}(\mathbf{x})_{\alpha+i\frac{\ell}{\delta}}^{-1} \\ &= \left(\prod_{i=0}^k \mathcal{R}(\mathbf{x})_{\alpha+i\frac{\ell}{\delta}+1} \right) \cdot \left(\prod_{i=0}^k \mathcal{R}(\mathbf{x})_{\alpha+i\frac{\ell}{\delta}} \right)^{-1} \\ &= c(\mathbf{x}) \cdot c(\mathbf{x})^{-1} = \phi, \end{aligned}$$

where we let $\mathcal{R}(\mathbf{x})_p = \phi$ for all $p \notin [\frac{n+\ell-\delta}{\delta}]$. ■

Example 5. Recall from Example 1, that for $\ell = 3, \delta = 1$ and $\mathbf{x} = (1, 2, 0, 1, 2, 2)$, we had the following read sub-derivatives: $\Delta^0(\mathbf{x}) = (1, \phi, 1^{-1})$, $\Delta^1(\mathbf{x}) = (2, \phi, 2^{-1})$ and $\Delta^2(\mathbf{x}) = (0, 0^{-1}, 2^{-1})$. Observe that $\prod_{i=0}^3 \Delta^0(\mathbf{x})_i = \prod_{i=0}^3 \Delta^1(\mathbf{x})_i = \prod_{i=0}^4 \Delta^2(\mathbf{x})_i = \phi$.

The aforementioned properties lead to an important consequence regarding the minimum Hamming distance between two distinct, error-free read vectors.

Theorem 1. *When $\ell > 1$ and $\delta = 1$, for any two distinct $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$, $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) \geq 2$.*

Proof: Assume that $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 1$, and let i denote the unique index where $\mathcal{R}(\mathbf{x})$ and $\mathcal{R}(\mathbf{y})$ differ, i.e., $\mathcal{R}(\mathbf{x})_i \neq \mathcal{R}(\mathbf{y})_i$. From Lemma 2, we infer that

$$(c(\mathbf{x}) \cdot c(\mathbf{y})^{-1})^\ell = \prod_{j=1}^{\frac{n+\ell}{\delta}-1} \mathcal{R}(\mathbf{x})_j \cdot \mathcal{R}(\mathbf{y})_j^{-1} = \mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1}.$$

Since the left-most equality suggests that each positive and negative degree should be divisible by ℓ , and we know that the sum of degrees in each of $\mathcal{R}(\mathbf{x})_i$ and $\mathcal{R}(\mathbf{y})_i$ must be ℓ , the only possibility involves $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} = a^\ell b^{-\ell}$ for some $a, b \in \Sigma_q$, $a \neq b$.

However, denoting $\alpha \triangleq (i-1) \bmod \ell$ it also follows that $\Delta^\alpha(\mathbf{x})$ and $\Delta^\alpha(\mathbf{y})$ differ in a unique index, at which $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{x})_{i-1}^{-1} = (\mathcal{R}(\mathbf{y})_i a^\ell b^{-\ell}) \cdot \mathcal{R}(\mathbf{y})_{i-1}^{-1} \neq \mathcal{R}(\mathbf{y})_i \cdot \mathcal{R}(\mathbf{y})_{i-1}^{-1}$. Hence, by Lemma 2

$$\begin{aligned} c(\phi) &= \prod_{i=1}^{\lfloor \frac{n-(\alpha+1)\delta}{\ell} \rfloor + 2} \Delta^\alpha(\mathbf{x})_i \\ &= a^\ell b^{-\ell} \prod_{i=1}^{\lfloor \frac{n-(\alpha+1)\delta}{\ell} \rfloor + 2} \Delta^\alpha(\mathbf{y})_i = a^\ell b^{-\ell} c(\phi), \end{aligned}$$

in contradiction. ■

C. Error Model

Similar to [7], we study the occurrence of substitution errors in read vectors and design suitable error-correcting constructions. To suitably define what constitutes an error-correcting construction in our framework, we first define the set of vectors that may result from at most t substitutions on a vector $\mathbf{u} \in \Sigma^n$, for any alphabet Σ , as

$$B_t(\mathbf{u}) \triangleq \{\mathbf{v} \in \Sigma^n : d_H(\mathbf{u}, \mathbf{v}) \leq t\}. \quad (3)$$

In our application, we will only be interested in $B_t(\mathcal{R}(\mathbf{x}))$, for some $\mathbf{x} \in \Sigma_q^n$. Under this framework, we define an error-correcting code as follows.

Definition 5. A code \mathcal{C} is said to be a t -substitution read code for the parameters ℓ, δ , if for any two distinct $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$, it holds that $B_t(\mathcal{R}(\mathbf{x})) \cap B_t(\mathcal{R}(\mathbf{y})) = \emptyset$.

In words, \mathcal{C} is a t -substitution read code if obtaining any noisy version of any codeword, where at most t substitutions occur, allows one to uniquely reconstruct that codeword. The redundancy of \mathcal{C} is given by $n - \log |\mathcal{C}|$, where $|\mathcal{C}|$ denotes the code size, i.e., the number of codewords in \mathcal{C} .

This work focuses on the case when $\delta = 1$ and $t = 1$. To this end, we seek to find a code that can correct a single substitution error in the read vectors of its constituent codewords, i.e., a single-substitution read code. In the upcoming sections, we deduce that the redundancy of any such code is bounded from below by $\log \log n - \log \binom{q}{2} - o(1)$. Subsequently, we also construct a code that is near-optimal.

III. MINIMUM REDUNDANCY OF SINGLE-SUBSTITUTION READ CODES

To establish a lower bound on the redundancy required by a single-substitution read code, we first attempt to characterize the relationship between any two non-binary vectors $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$, that might be confusable after a single substitution in their respective read vectors.

A. Characterization of Confusable Read Vectors

To proceed in this direction, we first note from Theorem 1 that there exists no two distinct vectors $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$ that satisfy $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 1$ for any $\ell > 1$. Thus, we attempt to ascertain the conditions under which $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$ may occur, since \mathbf{x} and \mathbf{y} are confusable if and only if $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) \leq 2$.

Lemma 3. For $\ell \geq 3$, any two distinct vectors $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$ satisfy $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$ if and only if there exist distinct $i, j \in [n + \ell - 1]$, for which $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} = \mathcal{R}(\mathbf{x})_j^{-1} \cdot \mathcal{R}(\mathbf{y})_j$, $j \equiv i \pmod{\ell}$ and $\mathcal{R}(\mathbf{x})_r = \mathcal{R}(\mathbf{y})_r$ for all $r \notin \{i, j\}$.

Proof: Let $i < j$ represent the indices at which $\mathcal{R}(\mathbf{x})$ and $\mathcal{R}(\mathbf{y})$ differ, i.e., $\mathcal{R}(\mathbf{x})_i \neq \mathcal{R}(\mathbf{y})_i$ and $\mathcal{R}(\mathbf{x})_j \neq \mathcal{R}(\mathbf{y})_j$. As $(\mathcal{R}(\mathbf{x})_1^{i-1}, \mathcal{R}(\mathbf{x})_{j+1}^{n-\ell+1}) = (\mathcal{R}(\mathbf{y})_1^{i-1}, \mathcal{R}(\mathbf{y})_{j+1}^{n-\ell+1})$, we may infer from Corollary 2 that $\mathbf{x}_1^{i-1} = \mathbf{y}_1^{i-1}$ and $\mathbf{x}_{j-\ell+2}^n = \mathbf{y}_{j-\ell+2}^n$. As a consequence, we obtain $x_i \cdot y_i^{-1} = \mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} \neq c(\phi)$, i.e., $x_i \neq y_i$.

Similarly, $x_{j-\ell+1} \cdot y_{j-\ell+1}^{-1} = \mathcal{R}(\mathbf{x})_j \cdot \mathcal{R}(\mathbf{y})_j^{-1} \neq c(\phi)$. On account of Lemma 2, we also have

$$\begin{aligned} (x_i \cdot y_i^{-1})(x_{j-\ell+1} \cdot y_{j-\ell+1}^{-1}) &= \prod_{i=1}^{n+\ell-1} \mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} \\ &= c(\mathbf{x})^\ell c(\mathbf{y})^{-\ell}, \end{aligned}$$

hence the degree in $(x_i \cdot y_i^{-1})(x_{j-\ell+1} \cdot y_{j-\ell+1}^{-1})$ of each symbol in Σ_q is a multiple of ℓ . Since $\ell \geq 3$, it follows that $(x_i \cdot y_i^{-1})(x_{j-\ell+1} \cdot y_{j-\ell+1}^{-1}) = c(\phi)$. Because $x_i \neq y_i$, we have

$x_i = y_{j-\ell+1}$ and $y_i = x_{j-\ell+1}$ (i.e., $c(\mathbf{x}) = c(\mathbf{y})$), or, put differently, $\mathcal{R}(\mathbf{x})_j \cdot \mathcal{R}(\mathbf{y})_j^{-1} = x_i^{-1} y_i = \mathcal{R}(\mathbf{x})_i^{-1} \cdot \mathcal{R}(\mathbf{y})_i$.

Finally, if $j \not\equiv i, i + 1 \pmod{\ell}$ then under Lemma 2 we observe (denoting $\alpha \triangleq i \pmod{\ell}$)

$$\begin{aligned} c(\phi) &= \prod_k \Delta^\alpha(\mathbf{x})_k \\ &= \left(\prod_k \Delta^\alpha(\mathbf{y})_k \right) (\mathcal{R}(\mathbf{x})_i^{-1} \mathcal{R}(\mathbf{y})_i) \\ &= \mathcal{R}(\mathbf{x})_i^{-1} \mathcal{R}(\mathbf{y})_i, \end{aligned}$$

in contradiction. Similarly, if $j \equiv i + 1 \pmod{\ell}$ then

$$\begin{aligned} c(\phi) &= \prod_k \Delta^\alpha(\mathbf{x})_k \\ &= \left(\prod_k \Delta^\alpha(\mathbf{y})_k \right) (\mathcal{R}(\mathbf{x})_i^{-1} \mathcal{R}(\mathbf{y})_i) (\mathcal{R}(\mathbf{x})_j \mathcal{R}(\mathbf{y})_j^{-1}) \\ &= \mathcal{R}(\mathbf{x})_i^{-2} \mathcal{R}(\mathbf{y})_i^2, \end{aligned}$$

again in contradiction. Hence, $i \equiv j \pmod{\ell}$, concluding the proof. ■

The proof of Lemma 3 demonstrates that if $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$ and $i < j$ are the two indices at which $\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})$ differ, then $\mathbf{x}_1^{i-1} = \mathbf{y}_1^{i-1}$ and $\mathbf{x}_{j-\ell+2}^n = \mathbf{y}_{j-\ell+2}^n$. In what follows we demonstrate that $\mathbf{x}_i^{j-\ell+1}, \mathbf{y}_i^{j-\ell+1}$ are also necessarily constrained by this assumption, to a specific structure.

Lemma 4. For $\ell \geq 3$, any two vectors $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$ that satisfy $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$, i.e., $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} = \mathcal{R}(\mathbf{x})_j^{-1} \cdot \mathcal{R}(\mathbf{y})_j$ for some $i, j \in [n + \ell - 1]$ such that $i < j$, it must hold that $(y_i, y_{i+1}) = (x_{i+1}, x_i)$.

Proof: From Corollary 2 and $\mathcal{R}(\mathbf{x})_1^{i-1} = \mathcal{R}(\mathbf{y})_1^{i-1}$, we infer that $\mathbf{x}_1^{i-1} = \mathbf{y}_1^{i-1}$ and $\mathbf{x}_{j-\ell+2}^n = \mathbf{y}_{j-\ell+2}^n$. It directly follows from $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} = x_i \cdot y_i^{-1} \neq c(\phi)$ that $x_i \neq y_i$.

Note that Lemma 3 suggests $j - i \geq \ell \geq 3$. Thus, we must have $\mathcal{R}(\mathbf{x})_{i+1} = \mathcal{R}(\mathbf{y})_{i+1}$, or equivalently, $\mathcal{R}(\mathbf{x})_{i+1} \cdot \mathcal{R}(\mathbf{y})_{i+1}^{-1} = c(\phi)$. Since $\mathbf{x}_{i-\ell+2} = \mathbf{y}_{i-\ell+2}$, the preceding requirement essentially translates to $c(\mathbf{x}_{i+1}^{i+1}) \cdot c(\mathbf{y}_{i+1}^{i+1})^{-1} = c(\phi)$. Since $x_i \neq y_i$, we conclude that $x_{i+1} = y_i$ and $y_{i+1} = x_i$. ■

Example 6. We return to $\mathbf{x} = (1, 2, 0, 1, 2, 2)$ from Example 1, and recall that $\ell = 3, \delta = 1$. Also recall that we observed that $\mathcal{R}(\mathbf{x}) = (1, 12, 012, 012, 012, 12^2, 2^2, 2)$. Now also consider $\mathbf{y} = (2, 1, 0, 2, 1, 2)$ for which $\mathcal{R}(\mathbf{y}) = (2, 12, 012, 012, 012, 12^2, 12, 2)$. Evidently, $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$ and $i = 1, j = 7$ are the indices at which $\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})$ differ. Note that, indeed, $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} = 1^1 2^{-1} = \mathcal{R}(\mathbf{y})_j \cdot \mathcal{R}(\mathbf{x})_j^{-1}$ and $j \equiv i \pmod{\ell}$, in keeping with Lemma 3. Also, as suggested by Lemma 4, it holds that $(x_i, x_{i+1}) = (1, 2) = (y_{i+1}, y_i)$. Before concluding, also observe, peculiarly, that $(x_{i+\ell}, x_{i+\ell+1}) = (y_{i+\ell+1}, y_{i+\ell})$ and $\mathbf{x}_{i+2}^{i+\ell-1} = \mathbf{y}_{i+2}^{i+\ell-1}$ (note that, here, $j - \ell = i + \ell$, hence the pattern connecting $\mathbf{x}_i^{j-\ell+1}, \mathbf{y}_i^{j-\ell+1}$ is fully found). The forthcoming analysis demonstrate that this is not a coincidence.

Further inspection reveals how this pattern of symbol alternation in the q -ary vectors may carry over to subsequent indices.

Lemma 5. For $\ell \geq 3$, consider two vectors $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$, such that for some $i, j \in [n + \ell - 1]$, $i < j$, $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} = \mathcal{R}(\mathbf{x})_j^{-1} \cdot \mathcal{R}(\mathbf{y})_j$, and for all $p \notin \{i, j\}$, $\mathcal{R}(\mathbf{x})_p = \mathcal{R}(\mathbf{y})_p$. Assume for some $t \geq i$ that $\mathbf{x}_{t-\ell+2}^{t-1} = \mathbf{y}_{t-\ell+2}^{t-1}$, $(x_t, x_{t+1}) = (y_{t+1}, y_t)$. Then, one of the following conditions will hold.

- 1) $\mathbf{x}_{t+2}^n = \mathbf{y}_{t+2}^n$ and $j = t + \ell$; or
- 2) $\mathbf{x}_{t+2}^{t+\ell-1} = \mathbf{y}_{t+2}^{t+\ell-1}$, $(x_{t+\ell}, x_{t+\ell+1}) = (x_t, x_{t+1})$, $(y_{t+\ell}, y_{t+\ell+1}) = (y_t, y_{t+1})$ and $j > t + \ell + 1$.

Proof: Let m indicate the smallest index strictly greater than $t + 1$ for which $x_m \neq y_m$; if $\mathbf{x}_{t+2}^n = \mathbf{y}_{t+2}^n$, consider $m = \infty$. Further recall throughout the proof that from $\mathcal{R}(\mathbf{x})_p = \mathcal{R}(\mathbf{y})_p$ for $j < p < n + \ell$ and Corollary 2, it follows that $\mathbf{x}_{j-\ell+2}^n = \mathbf{y}_{j-\ell+2}^n$.

We start by noting that $m \geq t + \ell$; indeed, if $m < t + \ell$ then $\mathcal{R}(\mathbf{y})_m \cdot \mathcal{R}(\mathbf{x})_m^{-1} = y_m \cdot x_m^{-1} \neq c(\phi)$, hence $j = m$ but $\mathbf{x}_{j-\ell+2}^n = \mathbf{y}_{j-\ell+2}^n$ contradicts $x_{t+1} \neq y_{t+1}$. We continue this proof by cases.

Case 1) If $m > t + \ell$, we deduce that $\mathcal{R}(\mathbf{y})_{t+\ell} \cdot \mathcal{R}(\mathbf{x})_{t+\ell}^{-1} = y_{t+1} \cdot x_{t+1}^{-1} \neq c(\phi)$. Thus, $j = t + \ell$, implying $\mathbf{x}_{t+2}^n = \mathbf{y}_{t+2}^n$ (in particular, this case is only possible when $m = \infty$).

Case 2) If $m = t + \ell$, then $\mathbf{x}_{j-\ell+2}^n = \mathbf{y}_{j-\ell+2}^n$ implies that $j \geq t + 2\ell - 1 > t + \ell + 1$.

Observe that $\mathcal{R}(\mathbf{y})_{t+\ell} \cdot \mathcal{R}(\mathbf{x})_{t+\ell}^{-1} = (y_{t+\ell} y_{t+1}) \cdot (x_{t+\ell} x_{t+1})^{-1} = c(\phi)$, and hence $(x_{t+\ell}, y_{t+\ell}) = (y_{t+1}, x_{t+1})$. In turn, $\mathcal{R}(\mathbf{y})_{t+\ell+1} \cdot \mathcal{R}(\mathbf{x})_{t+\ell+1}^{-1} = c(\mathbf{y}_{t+\ell+1}^{t+\ell}) \cdot c(\mathbf{x}_{t+\ell+1}^{t+\ell})^{-1} = c(\phi)$ now implies $(x_{t+\ell+1}, y_{t+\ell+1}) = (y_{t+\ell}, x_{t+\ell})$. ■

Example 7. We refer back to Example 6 to demonstrate the implications of Lemma 5, and note that when $t \in \{1, 4\}$, we have $\mathbf{x}_{t-\ell+2}^{t-1} = \mathbf{y}_{t-\ell+2}^{t-1}$, $\mathbf{x}_t^{t+1} = (1, 2)$ and $\mathbf{y}_t^{t+1} = (2, 1)$. As before, let j denote the last index where $\mathcal{R}(\mathbf{y})_j \cdot \mathcal{R}(\mathbf{x})_j^{-1} \neq c(\phi)$. Now when $t = 4$, it holds that $\mathbf{x}_{t+2}^n = \mathbf{y}_{t+2}^n$ and $j = t + \ell$. On the other hand, when $t = 1$, we observe that $\mathbf{x}_{t+2}^{t+\ell-1} = \mathbf{y}_{t+2}^{t+\ell-1}$, $\mathbf{x}_{t+\ell}^{t+\ell+1} = (1, 2)$ and $\mathbf{y}_{t+\ell}^{t+\ell+1} = (2, 1)$ and $j = t + 2\ell$.

Upon successive applications of Lemma 5 in conjunction with Lemma 3 and Lemma 4, we arrive at the following theorem.

Theorem 2. For $\ell \geq 3$ and any $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$, the following statements are equivalent:

- 1) $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$.
- 2) There exist distinct $i, j \in [n + \ell - 1]$, $j \equiv i \pmod{\ell}$, such that $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} = \mathcal{R}(\mathbf{x})_j^{-1} \cdot \mathcal{R}(\mathbf{y})_j \neq c(\phi)$ and $\mathcal{R}(\mathbf{x})_r = \mathcal{R}(\mathbf{y})_r$ for all $r \notin \{i, j\}$.
- 3) There exist $p \geq 1$ and $i \in [n - (p-1)\ell - 1]$ such that for all $m \in \{0, 1, \dots, p-1\}$ it holds that $\mathbf{x}_{i+ml}^{i+ml+1} = (a, b)$, $\mathbf{y}_{i+ml}^{i+ml+1} = (b, a)$ where $a, b \in \Sigma_q$ and $a \neq b$, and $x_r = y_r$ for all $r \notin \bigcup_{m \in \{0, 1, \dots, p-1\}} \{i + ml, i + ml + 1\}$.

Further, if these conditions hold, then $j = i + p\ell$ in the above notation.

B. An Upper Bound on the Code Size

We derive a lower bound on the redundancy required by a single-substitution read code by adopting the approach employed in [8]. More precisely, we consider a graph $\mathcal{G}(n)$

containing vertices corresponding to all vectors in Σ_q^n . Any two vertices in $\mathcal{G}(n)$ that signify two distinct q -ary vectors, say $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$, are considered to be adjacent if and only if $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$. Therefore, any independent set (i.e., a subset of vertices of $\mathcal{G}(n)$, wherein no two vertices are adjacent) is a single-substitution read code. Before further detailing our proof approach, we introduce some relevant definitions.

Definition 6. A clique in a graph \mathcal{G} is a subset of vertices of \mathcal{G} , wherein any two vertices are adjacent. A clique cover \mathcal{Q} is then a collection of cliques, such that every vertex in \mathcal{G} belongs to at least one clique in \mathcal{Q} .

The following graph-theoretic result is well-known [16].

Theorem 3. If \mathcal{Q} is a clique cover, then the size of any independent set is at most $|\mathcal{Q}|$.

For the remainder of this section, we seek to define a clique cover \mathcal{Q} by utilizing Theorem 2. By Theorem 3, the size of such a clique cover will serve as an upper bound on the cardinality of a single-substitution read code.

Definition 7. Let $\mathcal{G}'(n)$ be the graph whose vertices are all vectors in Σ_q^n , and an edge connects $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$ if and only if $\{\mathbf{x}, \mathbf{y}\} = \{\mathbf{u} \circ (ab)^j \circ \mathbf{v}, \mathbf{u} \circ (ba)^j \circ \mathbf{v}\}$, for some j , sub-strings \mathbf{u}, \mathbf{v} and $a, b \in \Sigma_q$ where $a \neq b$.

Observe that when $q = 2$, the preceding definition is identical to that in [8, Sec. IV].

Our method of proof would be to pull back a clique-cover from \mathcal{G}' based on the non-binary extension of [8, Lem. 7], i.e., Lemma 6, into \mathcal{G} . To do that, we have the following definition.

Definition 8. For a positive integer p , define a permutation π_p on Σ_q^n as follows. For all $\mathbf{x} \in \Sigma_q^n$, arrange the coordinates of $\mathbf{x}_1^{p\ell \lfloor n/(p\ell) \rfloor}$ in a matrix $X \in \Sigma_q^{p \lfloor n/(p\ell) \rfloor \times \ell}$, by row (first fill the first row from left to right, then the next, etc.). Next, partition X into sub-matrices of dimension $p \times 2$ (if ℓ is odd, we ignore X 's right-most column). Finally, going through each sub-matrix (from left to right, and then top to bottom), we concatenate its rows to obtain $\pi_p(\mathbf{x})$ (where unused coordinates from \mathbf{x} are appended arbitrarily).

More precisely, for all $0 \leq i < \lfloor \frac{n}{p\ell} \rfloor$, $0 \leq j < \lfloor \frac{\ell}{2} \rfloor$ and $0 \leq k < p$ denote

$$\mathbf{x}^{(i,j,k)} \triangleq x_{(ip+k)\ell+2j+1} x_{(ip+k)\ell+2j+2};$$

then

$$\mathbf{x}^{(i,j)} \triangleq \mathbf{x}^{(i,j,0)} \circ \dots \circ \mathbf{x}^{(i,j,p-1)}$$

and

$$\mathbf{x}^{(i)} \triangleq \mathbf{x}^{(i,0)} \circ \dots \circ \mathbf{x}^{(i, \lfloor \ell/2 \rfloor - 1)}.$$

Then $\pi_p(\mathbf{x}) = \mathbf{x}^{(0)} \circ \dots \circ \mathbf{x}^{(\lfloor n/p\ell \rfloor - 1)} \circ \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}}$ is composed of all coordinates of \mathbf{x} not previously included.

Example 8. Recall from Example 6 that for $\mathbf{x} = (1, 2, 0, 1, 2, 2)$ and $\mathbf{y} = (2, 1, 0, 2, 1, 2)$ it holds that $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$. To obtain $\pi_p(\mathbf{x})$ and $\pi_p(\mathbf{y})$ for $p = 2$, note that

$$X = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 2 & 2 \end{bmatrix}, Y = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 1 & 2 \end{bmatrix}.$$

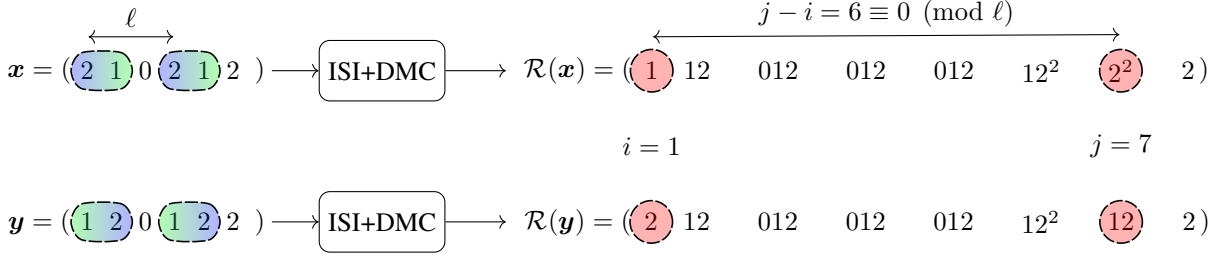


Fig. 4. Demonstration of Theorem 2 through Examples 6 and 7. For $\ell = 3, \delta = 1$, we observe that $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$. The vectors \mathbf{x}, \mathbf{y} differ in pairs of indices that are separated by ℓ positions (Lemma 4), while the mismatching indices in $\mathcal{R}(\mathbf{x})$ and $\mathcal{R}(\mathbf{y})$ are always spaced apart by a multiple of ℓ

Since ℓ is odd, we ignore the last column in X and Y and partition the respective results into 2×2 sub-matrices to ultimately obtain $\pi_p(\mathbf{x}) = (1, 2, 1, 2, 0, 2)$ and $\pi_p(\mathbf{y}) = (2, 1, 2, 1, 0, 2)$ (here, unused coordinates were appended in the order of their indices).

Next, we detail the non-binary extension of the clique-cover used in [8, Section IV].

Definition 9. For a positive integer p , let

$$\Lambda_{p,a,b} \triangleq \{(\mathbf{v})^j(\mathbf{t})^{p-j} : j \in [p], \{\mathbf{v}, \mathbf{t}\} = \{ab, ba\}\},$$

where $\mathbf{v}^0 = \mathbf{t}^0$ is the empty word, and $\tilde{\Lambda}_{p,a,b} \triangleq \Sigma_q^{2p} \setminus \Lambda_{p,a,b}$. Further, let

$$\Gamma \triangleq \left\{ (\mathbf{u}, \mathbf{w}, a, b) : i \in [m], \mathbf{u} \in \tilde{\Lambda}_{p,a,b}^{i-1}, \mathbf{w} \in \Sigma_q^{2p(m-i)}, a, b \in \Sigma_q, a \neq b \right\},$$

where $m = \lfloor \frac{\ell}{2} \rfloor \lfloor \frac{n}{p\ell} \rfloor$, and $\tilde{\Lambda}_p^0$ is the singleton containing the empty word. Then, for all $\gamma = (\mathbf{u}, \mathbf{w}, a, b) \in \Gamma$ define

$$Q_\gamma^{(0)} \triangleq \{(\mathbf{u}(ab)^h(ba)^{p-h}\mathbf{w} : h \in [p]\},$$

$$Q_\gamma^{(1)} \triangleq \{(\mathbf{u}(ba)^h(ab)^{p-h}\mathbf{w} : h \in [p]\}.$$

Finally, let

$$\mathcal{Q}(m, p) \triangleq \left\{ \{\mathbf{x}\} : \mathbf{x} \in \tilde{\Lambda}_p^m \right\} \cup \left\{ Q_\gamma^{(0)}, Q_\gamma^{(1)} : \gamma \in \Gamma \right\},$$

where $\tilde{\Lambda}_p = \Sigma_q^{2p} \setminus \bigcup_{\substack{a,b \in \Sigma_q \\ a \neq b}} \Lambda_{p,a,b}$.

Example 9. For $p = 2, a = 1$ and $b = 2$, we obtain $\Lambda_{p,a,b} = \{(1, 2, 2, 1), (1, 2, 1, 2), (2, 1, 1, 2), (2, 1, 2, 1)\}$. Revisiting Example 8, we observe that for $\gamma = (\mathbf{u}, \mathbf{w}, 1, 2) \in \Gamma$, where $\mathbf{u} = \tilde{\Lambda}_{p,a,b}^0$ and $\mathbf{w} = (2)$,

$$Q_\gamma^{(0)} = \{(1, 2, 0, 2, 1, 2), \mathbf{x} = (1, 2, 0, 1, 2, 2)\},$$

$$Q_\gamma^{(1)} = \{(2, 1, 0, 1, 2, 2), \mathbf{y} = (2, 1, 0, 2, 1, 2)\}.$$

It follows from Theorem 2 that $Q_\gamma^{(0)} \cup Q_\gamma^{(1)}$ forms a clique.

Lemma 6. $\mathcal{Q}(m, p)$ is a clique-cover of $\mathcal{G}^l(2pm)$, where $m = \lfloor \frac{\ell}{2} \rfloor \lfloor \frac{n}{p\ell} \rfloor$.

This is the non-binary analogue of [8, Lemma 7], and the proof is relegated to the appendix. We now define a clique-cover for the graph $\mathcal{G}(n)$ in the theorem below.

Theorem 4. Let

$$\mathcal{Q}_p \triangleq \{ \pi_p^{-1}(Q \times \{\mathbf{z}\}) : Q \in \mathcal{Q}(m, p), \mathbf{z} \in \Sigma_q^{n-2pm} \},$$

where $\pi_p^{-1}(A) \triangleq \{ \mathbf{u} \in \Sigma_q^n : \pi_p(\mathbf{u}) \in A \}$. Then, \mathcal{Q}_p is a clique-cover in $\mathcal{G}(n)$.

Proof: First, observe that it readily follows from $\bigcup \mathcal{Q}(m, p) = \Sigma_q^{2pm}$ that $\bigcup \mathcal{Q}_p = \Sigma_q^n$. It is therefore left to prove that every element of \mathcal{Q}_p is a clique of $\mathcal{G}(n)$.

Next, observe for all $Q \in \mathcal{Q}(m, p)$ and $\mathbf{z} \in \Sigma_q^{n-2pm}$ that either Q is a singleton, or all elements $\mathbf{y} \in Q \times \{\mathbf{z}\}$ agree on all coordinates y_k except $2(i-1)p < k \leq 2ip$ for some $i \in [m]$, and $\mathbf{y}_{2(i-1)p} \in \{(ab)^h(ba)^{p-h}, (ba)^h(ab)^{p-h}\}$ for some $h \in [p]$ and $a, b \in \Sigma_q$ where $a \neq b$. That is, either $\pi_p^{-1}(Q \times \{\mathbf{z}\})$ is a singleton, or all elements $\mathbf{x} \in \pi_p^{-1}(Q \times \{\mathbf{z}\})$ agree on all coordinates except, in the notation of Definition 8, $\mathbf{x}^{(i,j)}$ for some $0 \leq i < \lfloor \frac{n}{p\ell} \rfloor$, $0 \leq j < \lfloor \frac{\ell}{2} \rfloor$, and $\mathbf{x}^{(i,j)} \in \{(ab)^h(ba)^{p-h}, (ba)^h(ab)^{p-h}\}$ for some $h \in [p]$. That is, $\mathbf{x}^{(i,j,k)} = ab(ba)$ for all $0 \leq k < h$, and $\mathbf{x}^{(i,j,k)} = ba$ (respectively, ab) for all $h \leq k < p$. By Theorem 2, it holds that $d_H(\mathcal{R}(\mathbf{x}_1), \mathcal{R}(\mathbf{x}_2)) = 2$ for all $\mathbf{x}_1, \mathbf{x}_2 \in \pi_p^{-1}(Q \times \{\mathbf{z}\})$. ■

Finally, we can obtain a lower bound on the redundancy of a single-substitution read code from the following result on the size of a clique-cover.

Lemma 7.

$$|\mathcal{Q}_p| = q^n \left[\left(1 - \binom{q}{2} \frac{2p}{q^{2p}} \right)^m + \frac{1}{p} \binom{q}{2} \left(1 - \left(1 - \frac{2p}{q^{2p}} \right)^m \right) \right],$$

where $m = \lfloor \frac{\ell}{2} \rfloor \lfloor \frac{n}{p\ell} \rfloor$.

Proof: Since the number of singletons is given by

$$|\tilde{\Lambda}_p^m| = \left(q^{2p} - \binom{q}{2} 2p \right)^m,$$

while the number of cliques of size p evaluates to

$$\begin{aligned}
2|\Gamma| &= 2 \binom{q}{2} \sum_{i=1}^m |\tilde{\Lambda}_{p,a,b}^{i-1}| \cdot q^{2p(m-i)} \\
&= 2 \binom{q}{2} \sum_{i=1}^m (q^{2p} - 2p)^{i-1} q^{2p(m-i)} \\
&= 2q^{2p(m-1)} \binom{q}{2} \sum_{i=1}^m \left(1 - \frac{2p}{q^{2p}} \right)^{i-1} \\
&= q^{2pm} \frac{1}{p} \binom{q}{2} \left(1 - \left(1 - \frac{2p}{q^{2p}} \right)^m \right).
\end{aligned}$$

Hence,

$$|\mathcal{Q}(m, p)| = q^{2pm} \left[\left(1 - \binom{q}{2} \frac{2p}{q^{2p}}\right)^m + \frac{1}{p} \binom{q}{2} \left(1 - \left(1 - \frac{2p}{q^{2p}}\right)^m\right) \right],$$

and the claim follows. \blacksquare

By using $\left(1 - \frac{2p}{q^{2p}}\right) \geq \left(1 - \binom{q}{2} \frac{2p}{q^{2p}}\right)$, it readily follows that for any positive integer p ,

$$\log|\mathcal{Q}_p| \leq n - \log(p) + \log\left(p\left(1 - \frac{2p}{q^{2p}}\right)^m + \binom{q}{2}\right).$$

Based on $m \geq \lfloor \frac{n}{2p} \rfloor - \lfloor \frac{\ell}{2} \rfloor$ we may further bound

$$\log|\mathcal{Q}_p| \leq n - \log(p) + \log\left(p\left(1 - \frac{2p}{q^{2p}}\right)^{\lfloor n/2p \rfloor - \lfloor \ell/2 \rfloor} + \binom{q}{2}\right).$$

By employing the non-binary extension of [8, Lemma 9], as stated in Appendix B, we find that letting $p = \lceil \frac{1}{2}(1 - \epsilon) \log(n) \rceil$ for any $0 < \epsilon < 1$ yields $p\left(1 - \frac{2p}{q^{2p}}\right)^{\lfloor n/2p \rfloor} = o(1)$, hence based on Theorem 3 we arrive at the following theorem.

Theorem 5. *The minimum redundancy of any single-substitution read code is at least*

$$\log \log(n) - \log\left(\frac{q}{2}\right) - o(1).$$

C. Error correction with multiple reads

It is well known that existing DNA synthesis technologies tend to produce many duplicates of each strand and that the process of PCR amplification, used during sequencing, augments the number of copies even more, albeit at the cost of introducing errors [17]–[20]. As a consequence, the problem of reconstructing the channel input from multiple noisy versions at the receiver is of immense practical relevance [9]–[15]. Investigating how the availability of multiple noisy reads could lower the minimum redundancy required by an error-correcting code is similarly pertinent [8], [21].

With this in mind, we first consider the following lemma to see if and how multiple noisy reads might be leveraged to construct more efficient codes for correcting errors in (ℓ, δ) -read vectors.

Lemma 8. *Exactly one of the following conditions holds for $\ell \geq 3$ and any two distinct $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$.*

- 1) $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$ and $|B_1(\mathcal{R}(\mathbf{x})) \cap B_1(\mathcal{R}(\mathbf{y}))| = 2$;
or
- 2) $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) > 2$ and $|B_1(\mathcal{R}(\mathbf{x})) \cap B_1(\mathcal{R}(\mathbf{y}))| = \emptyset$.

Proof: Since Theorem 1 already precludes the possibility of $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 1$ and the case of $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) > 2$ follows from the triangle inequality, we proceed to prove the remaining case wherein $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) = 2$, i.e., \mathbf{x}, \mathbf{y} satisfy the conditions stated in Theorem 2.

More specifically, there exist distinct $i, j \in [n + \ell - 1]$ such that $\mathcal{R}(\mathbf{x})_i \cdot \mathcal{R}(\mathbf{y})_i^{-1} = \mathcal{R}(\mathbf{y})_j \cdot \mathcal{R}(\mathbf{x})_j^{-1} \neq \phi$ and for all

$k \notin \{i, j\}$, $\mathcal{R}(\mathbf{x})_k = \mathcal{R}(\mathbf{y})_k$. This implies that $B_1(\mathcal{R}(\mathbf{x})) \cap B_1(\mathcal{R}(\mathbf{y}))$ is exactly the following.

$$\begin{aligned} & \{(\mathcal{R}(\mathbf{x})_1, \dots, \mathcal{R}(\mathbf{x})_{i-1}, \mathcal{R}(\mathbf{y})_i, \mathcal{R}(\mathbf{x})_{i+1}, \dots, \mathcal{R}(\mathbf{x})_{n+\ell-1}), \\ & (\mathcal{R}(\mathbf{x})_1, \dots, \mathcal{R}(\mathbf{x})_{j-1}, \mathcal{R}(\mathbf{y})_j, \mathcal{R}(\mathbf{x})_{j+1}, \dots, \mathcal{R}(\mathbf{x})_{n+\ell-1})\} \\ & = \{(\mathcal{R}(\mathbf{y})_1, \dots, \mathcal{R}(\mathbf{y})_{j-1}, \mathcal{R}(\mathbf{x})_j, \mathcal{R}(\mathbf{y})_{j+1}, \dots, \\ & \mathcal{R}(\mathbf{y})_{n+\ell-1}), (\mathcal{R}(\mathbf{y})_1, \dots, \mathcal{R}(\mathbf{y})_{i-1}, \mathcal{R}(\mathbf{x})_i, \mathcal{R}(\mathbf{y})_{i+1}, \\ & \dots, \mathcal{R}(\mathbf{y})_{n+\ell-1})\}. \end{aligned}$$

Hence, the first case directly follows. \blacksquare

As in Section III-B, we wish to derive a lower bound on the redundancy required by a code that can reconstruct the channel input given two distinct noisy versions of its read vector. Such a code requires that for any two distinct codewords $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$ it holds that $|B_1(\mathcal{R}(\mathbf{x})) \cap B_1(\mathcal{R}(\mathbf{y}))| < 2$.

Nevertheless, Lemma 8 suggests that such codes are in fact identical to single-substitution read codes as defined in Definition 5. As a consequence, we arrive at the following lemma.

Lemma 9. *The minimum redundancy of any code able to correct a single substitution given two distinct noisy copies is at least*

$$\log \log(n) - \log\left(\frac{q}{2}\right) - o(1).$$

Remark: Although this result mirrors that of the standard substitution channel [9], it is nevertheless unexpected that the minimum redundancy requirement remains the same, unlike the single deletion channel or single edit channel [8], [21], owing to the different spaces over which the channel outputs are defined and especially the special characteristics imbued in the outputs of the ISI channel that aid in error correction.

However, Lemma 8 also implies the following: given three distinct noisy copies of the $(\ell, 1)$ -read vector of any $\mathbf{x} \in \Sigma_q^n$, one can uniquely reconstruct $\mathcal{R}(\mathbf{x})$ and thereby \mathbf{x} . Therefore, no redundancy at all is required in this case.

IV. SINGLE SUBSTITUTION READ CODES

It is already implied by Corollary 2 that a redundancy of $t \log n$ symbols suffices to correct at most t substitutions in the $(\ell, 1)$ -read vector. However, according to Theorem 5, a more efficient code may exist for the $t = 1$ case. This section introduces such a construction that is of optimal redundancy up to an additive constant.

We define a specific permutation for any $\mathbf{x} \in \Sigma_q^n$ as well as its $(\ell, 1)$ -read vector $\mathcal{R}(\mathbf{x})$ as

$$\begin{aligned} \mathbf{x}^\pi & \triangleq C_{\ell,1}^0(\mathbf{x}) \circ C_{\ell,1}^1(\mathbf{x}) \circ \dots \circ C_{\ell,1}^{\ell-1}(\mathbf{x}), \\ \mathcal{R}^\pi(\mathbf{x}) & \triangleq \mathcal{R}^0(\mathbf{x}) \circ \mathcal{R}^1(\mathbf{x}) \circ \dots \circ \mathcal{R}^{\ell-1}(\mathbf{x}), \end{aligned}$$

where $\mathcal{R}^{i-1}(\mathbf{x}) = (\mathcal{R}(\mathbf{x})_i, \mathcal{R}(\mathbf{x})_{i+\ell}, \dots, \mathcal{R}(\mathbf{x})_{i+k\ell})$ and $k = \lfloor \frac{n+\ell-1-i}{\ell} \rfloor$ for all $i \in [\ell]$. Recall from Definition 4 that $C_{\ell,1}^\alpha(\mathbf{x})$ refers to a subsequence of \mathbf{x} .

Example 10. *Reconsidering $\mathbf{x} = (1, 2, 0, 1, 2, 2)$ from Examples 1 and 3, we may verify that*

$$\begin{aligned} \mathbf{x}^\pi & = (1, 1, 2, 2, 0, 2), \\ \mathcal{R}^\pi(\mathbf{x}) & = (1, 012, 2^2, 12, 012, 2, 012, 12^2). \end{aligned}$$

To simplify presentation, we also define the following.

Definition 10. Let $a\text{-RLL}_q(n)$ be the set of all length- n q -ary vectors whose runs are of length at most a .

Definition 11. For $n, a > 0$ where $n \geq a$, and a code $C \subseteq \Sigma_q^n$, we say that C is an a -bounded single-substitution-correcting code, and denote $C \in \text{BS}_q(n, a)$, if a decoder of C can correct a single substitution given knowledge of the error location within a segment of length a .

We defer the construction of $\text{BS}_q(n, a)$ codes, and the study of their respective redundancies, to the end of this section.

Finally, we propose the following code to correct a single substitution in $(\ell, 1)$ -read vectors for $\ell \geq 3$.

Construction 1.

$$\mathcal{C}(n, \ell) = \left\{ \mathbf{x} \in \Sigma_q^n : C_{\ell, 1}^i(\mathbf{x}) \in \lceil \log qn \rceil\text{-RLL}_q(k_i) \right. \\ \left. \begin{aligned} & \forall i \in \{0, 1, \dots, \ell - 1\}; \\ & |\mathcal{R}^\pi(\mathbf{x})|_1 \bmod q \in \\ & \text{BS}_q(n + \ell - 1, 2\lceil \log(qn) \rceil + 2) \end{aligned} \right\},$$

where $k_i = \lfloor \frac{n-i-1}{\ell} \rfloor + 2$.

To prove that $\mathcal{C}(n, \ell)$ is a single-substitution read code, we first show how the inherent characteristics of read vectors reveal some information on the substitution error, particularly in regard to its composition and its location.

Lemma 10. If a substitution error affects the $(\ell, 1)$ -read vector of some $\mathbf{x} \in \Sigma_q^n$ where $\ell \geq 3$, thus producing a noisy copy $\mathcal{R}(\mathbf{x})'$, then there exist $\alpha, \beta \in \Sigma_\ell$ where $\alpha \equiv (\beta + 1) \bmod \ell$, such that $\prod_i \Delta^\beta(\mathbf{x}'_i) = (\prod_i \Delta^\alpha(\mathbf{x}'_i))^{-1} \neq c(\phi)$, and for all $\gamma \notin \{\alpha, \beta\}$, $\prod_i \Delta^\gamma(\mathbf{x}'_i) = c(\phi)$. This implies that

1) the composition error is

$$\prod_i (\mathcal{R}(\mathbf{x}'_i) \cdot \mathcal{R}(\mathbf{x})_i^{-1}) = \prod_i \Delta^\beta(\mathbf{x}'_i) \\ = \left(\prod_i \Delta^\alpha(\mathbf{x}'_i) \right)^{-1};$$

2) the error occurred at an index $k \in [n + \ell - 1]$, where $k \equiv \alpha \pmod{\ell}$.

Proof: Suppose the concerned substitution error occurs at index $k \in [n + \ell - 1]$. Thus, the noisy read vector can be expressed as $\mathcal{R}(\mathbf{x})' = (\mathcal{R}(\mathbf{x}'_1), \dots, \mathcal{R}(\mathbf{x}'_{n-\ell+1}))$, where $\mathcal{R}(\mathbf{x}'_k) \neq \mathcal{R}(\mathbf{x})_k$ and $\mathcal{R}(\mathbf{x}'_p) = \mathcal{R}(\mathbf{x})_p$ for all $p \neq k$.

Denoting $\alpha \triangleq k \bmod \ell$, $\beta \triangleq (k - 1) \bmod \ell$, observe that $\Delta^\beta(\mathbf{x})'$ and $\Delta^\alpha(\mathbf{x})'$ no longer uphold Lemma 2. Instead,

$$\prod_i \Delta^\beta(\mathbf{x}'_i) = \prod_i \Delta^\alpha(\mathbf{x}'_i)^{-1} \\ = \mathcal{R}(\mathbf{x}'_k) \cdot \mathcal{R}(\mathbf{x})_k^{-1},$$

which is the composition error. The preceding equation suggests that the error occurred somewhere in $\mathcal{R}^\beta(\mathbf{x})'$, which is a subsequence of $\mathcal{R}(\mathbf{x})'$. Alternatively, we say that the decoder can only infer the error position up to the modulo class of k . ■

Next, we show that some composition substitutions are trivial to correct.

Lemma 11. Say a composition substitution corrupts the i -th index of $\mathcal{R}(\mathbf{x})$ to $\mathcal{R}(\mathbf{x}'_i)$. This error is readily correctable if any of the following conditions holds

- 1) Denoting $\mathcal{R}(\mathbf{x}'_i) = 0^{i_0} \dots (q-1)^{i_{q-1}}$, if it does not hold that $0 \leq i_j \leq \ell$ for all $j \in \Sigma_q$, and $\sum_{j=0}^{q-1} i_j = \min\{i, \ell, n - i + 1\}$.
- 2) At least one of $\mathcal{R}(\mathbf{x}'_i) \mathcal{R}(\mathbf{x})_{i-1}^{-1}$ or $\mathcal{R}(\mathbf{x})_{i+1} \mathcal{R}(\mathbf{x}'_i)^{-1}$ is neither ϕ nor of the form $a \cdot b^{-1}$ for any $a, b \in \Sigma_q$.

Proof: Suppose the error occurred at index k . Then, we may express the noisy read vector as $\mathcal{R}(\mathbf{x})' = (\mathcal{R}(\mathbf{x}'_1), \dots, \mathcal{R}(\mathbf{x}'_{n+\ell-1}))$, where $\mathcal{R}(\mathbf{x}'_k) \neq \mathcal{R}(\mathbf{x})_k$ and $\mathcal{R}(\mathbf{x}'_p) = \mathcal{R}(\mathbf{x})_p$ for all $p \neq k$.

Since in the first case, it follows directly from Definition 1 that the error can be detected and corrected by Corollary 2, we direct our attention to the second case. On account of $\delta = 1$, we know that for any $p \in [n + \ell - 2]$, it should hold that $\mathcal{R}(\mathbf{x})_{p+1} \cdot \mathcal{R}(\mathbf{x})_p^{-1} = x_{p+1} x_{p+1-\ell}^{-1}$, which is either evaluates to ϕ or stays in the form $a \cdot b^{-1}$, where $a, b \in \Sigma_q$ and $a \neq b$. Say $\mathcal{R}(\mathbf{x}'_i) \mathcal{R}(\mathbf{x})_{i-1}^{-1}$ violates this. As a result, we immediately infer that $k \in \{i - 1, i\}$. However, since $\mathcal{R}(\mathbf{x}'_i) \cdot \mathcal{R}(\mathbf{x})_{i-1}^{-1}$ and $i \bmod \ell$ can be deduced due to Lemma 10, we are able to conclude that $k = i$, and thereby correct the error. ■

Example 11. $\mathcal{R}_{3,1}(\mathbf{v})' = (1, 12, 012, 012, 2^3, 12^2, 2^2, 2)$ arises from a single substitution in the $(3, 1)$ -read vector of some $\mathbf{v} \in \Sigma_3^6$. Since $\mathcal{R}(\mathbf{v}'_5) \mathcal{R}(\mathbf{v})_4'^{-1} = 0^{-1} 2^2 1^{-1}$, we know that either $\mathcal{R}(\mathbf{v})_4$ or $\mathcal{R}(\mathbf{v}'_5)$ is erroneous. Also, since $\prod_i \Delta^1(\mathbf{v}'_i) = (\prod_i \Delta^2(\mathbf{v}'_i))^{-1} = 0^{-1} 1^{-1} 2^2$, we use Lemma 10 to conclude that the composition error is $0^{-1} 1^{-1} 2^2$ and that the error location, say k , satisfies $k \bmod \ell = 2$. Thus $k = 5$, and we can reverse the substitution error by applying $\mathcal{R}(\mathbf{v})_5 \leftarrow \mathcal{R}(\mathbf{v}'_5) \cdot (0^{-1} 1^{-1} 2^2)^{-1}$, to finally obtain $\mathcal{R}_{3,1}(\mathbf{v}) = (1, 12, 012, 012, 012, 12^2, 2^2, 2)$, which corresponds to $\mathbf{v} = (1, 2, 0, 1, 2, 2) = \mathbf{x}$ from Example 1.

Due to Lemma 11, we focus for the rest of the section on proving that $\mathcal{C}(n, \ell)$ can correct a single substitution that is not readily correctable by Lemma 11. Next, we demonstrate that the index of such substitutions may be narrowed down.

Example 12. $\mathcal{R}_{3,1}(\mathbf{v})' = (1, 12, 012, 02^2, 012, 12^2, 2^2, 2)$ arises from a substitution in the $(3, 1)$ -read vector of some $\mathbf{v} \in \Sigma_3^6$. As $\prod_i \Delta^0(\mathbf{v}'_i) = \prod_i \Delta^1(\mathbf{v}'_i)^{-1} = 1^{-1} 2$, Lemma 10 suggests that the erroneous composition differs from the true composition by a factor of $1^{-1} 2$ and occurred somewhere in $(\mathcal{R}(\mathbf{v})'_1, \mathcal{R}(\mathbf{v})'_4, \mathcal{R}(\mathbf{v})'_7)$. Now assigning $\mathcal{R}(\mathbf{v})'_1 \leftarrow \mathcal{R}(\mathbf{v})'_1 \cdot 12^{-1}$ yields an invalid read vector since by definition, $\mathcal{R}(\mathbf{x})_1 \in \Sigma_q$. On the contrary, assigning $\mathcal{R}(\mathbf{v})'_4 \leftarrow \mathcal{R}(\mathbf{v})'_4 \cdot 12^{-1}$ or $\mathcal{R}(\mathbf{v})'_7 \leftarrow \mathcal{R}(\mathbf{v})'_7 \cdot 12^{-1}$ alters $\mathcal{R}(\mathbf{v})'$ into the $(3, 1)$ -read vector of $\mathbf{v} = (1, 2, 0, 1, 2, 2)$ or $\mathbf{v} = (1, 2, 0, 2, 1, 2)$ respectively.

Henceforth, we represent the subsequence reconstructed using Corollary 1 from left to right with a noisy read sub-derivative, say $\Delta^\beta(\mathbf{x})'$, as $\widehat{C}^\beta(\mathbf{x}) \triangleq (\widehat{x}_{\beta+1}, \widehat{x}_{\beta+1+\ell}, \dots, \widehat{x}_{\beta+1+\lfloor \frac{n-\beta-1}{\ell} \rfloor \ell})$. Analogously,

$\widetilde{C}^\beta(\mathbf{x})$ corresponds to right to left reconstruction. The following lemma outlines how a single substitution in a read vector, say $\mathcal{R}(\mathbf{x})$, affects the estimate of the input \mathbf{x} , obtained via left-to-right reconstruction.

Lemma 12. *Let a substitution at index k on $(\ell, 1)$ -read vector of $\mathbf{x} \in \Sigma_q^n$ where $\ell \geq 3$, produce $\mathcal{R}(\mathbf{x})'$. For $\beta \triangleq (k-1) \bmod \ell$, if there exists $i > 0$ such that $\mathcal{R}(\mathbf{x})'_{k+i\ell} \neq \mathcal{R}(\mathbf{x})'_{k+i\ell-1}$, then $\{\widehat{x}_k, \widehat{x}_{k+i\ell}\} \not\subseteq \Sigma_q$, where $\widehat{x}_k, \widehat{x}_{k+i\ell}$ are elements of $\widetilde{C}^\beta(\mathbf{x})$.*

Proof: Since $\mathcal{R}(\mathbf{x})'_k$ alone is erroneous, we infer that $(\widehat{x}_{\beta+1}, \widehat{x}_{\beta+\ell+1}, \dots, \widehat{x}_{k-\ell}) = (x_{\beta+1}, x_{\beta+\ell+1}, \dots, x_{k-\ell})$ and $\widehat{x}_{k+i\ell} \cdot x_{k+i\ell}^{-1} = \mathcal{R}(\mathbf{x})'_k \cdot \mathcal{R}(\mathbf{x})_k^{-1}$ for all $i \geq 0$.

Assume $\widehat{x}_k \in \Sigma_q$ and $i > 0$ is minimum such that $\mathcal{R}(\mathbf{x})'_{k+i\ell} \neq \mathcal{R}(\mathbf{x})'_{k+i\ell-1}$. Note that, equivalently, $\mathcal{R}(\mathbf{x})_{k+i\ell} \neq \mathcal{R}(\mathbf{x})_{k+i\ell-1}$; i.e., for all $0 < j < i$,

$$\Delta^\beta(\mathbf{x})'_{\frac{k-\beta-1}{\ell}+j+1} = \Delta^\beta(\mathbf{x})_{\frac{k-\beta-1}{\ell}+j+1} = c(\phi)$$

and $\Delta^\beta(\mathbf{x})'_{\frac{k-\beta-1}{\ell}+i+1} = \Delta^\beta(\mathbf{x})_{\frac{k-\beta-1}{\ell}+i+1} = x_{k+i\ell}x_k^{-1}$, where $x_{k+i\ell} \neq x_k$. It follows that

$$\begin{aligned} \widehat{x}_{k+i\ell} &= \prod_{j=1}^{\frac{k-\beta-1}{\ell}+i+1} \Delta^\beta(\mathbf{x})'_j \\ &= \widehat{x}_k \cdot \prod_{j=1}^i \Delta^\beta(\mathbf{x})'_{\frac{k-\beta-1}{\ell}+j+1} \\ &= \widehat{x}_k \cdot (x_{k+i\ell}x_k^{-1}), \end{aligned}$$

and since by assumption $\widehat{x}_k \in \Sigma_q \setminus \{x_k\}$ and $x_k \neq x_{k+i\ell}$, we have $\widehat{x}_{k+i\ell} \notin \Sigma_q$. ■

Corollary 3. *Let $\mathcal{R}(\mathbf{x})'$ arise from a substitution at index k on $(\ell, 1)$ -read vector of $\mathbf{x} \in \Sigma_q^n$ where $\ell \geq 3$. For $\beta \triangleq (k-1) \bmod \ell$, if there exists $j > 0$ such that $\mathcal{R}(\mathbf{x})'_{k-j\ell} \neq \mathcal{R}(\mathbf{x})'_{k-j\ell-1}$, then $\{\widetilde{x}_k, \widetilde{x}_{k-j\ell}\} \not\subseteq \Sigma_q$, where $\widetilde{x}_k, \widetilde{x}_{k-j\ell}$ are elements of $\widetilde{C}^\beta(\mathbf{x})$.*

A consequence of Lemma 10 and the preceding results is that reconstruction with any corrupted read subderivative from left to right and right to left might help us narrow in on the position of the substitution error. This is stated more formally as follows (for an illustration of the following lemma, see Figure 5).

Lemma 13. *For $\ell \geq 3$, let $\mathcal{R}(\mathbf{x})'$ be a noisy $(\ell, 1)$ -read vector of $\mathbf{x} \in \Sigma_q^n$, such that for some $\alpha, \beta \in \Sigma_\ell$, where $\alpha \equiv \beta + 1 \pmod{\ell}$, $\prod_i \Delta^\beta(\mathbf{x})'_i = \prod_i \Delta^\alpha(\mathbf{x})_i^{-1} \neq c(\phi)$. Reconstruction by Corollary 1 with $\Delta^\beta(\mathbf{x})'$ from left to right (respectively, right to left) yields $\widetilde{C}^\beta(\mathbf{x})$ ($C^\beta(\mathbf{x})$) for which we define i (j) as the minimum (maximum) index at which $\widehat{x}_{\beta+i\ell+1} \notin \Sigma_q$ ($\widetilde{x}_{\beta+j\ell+1} \notin \Sigma_q$), or $i = \lfloor \frac{n-\beta-1}{\ell} \rfloor + 1$ ($j = -1$) if no such index exists. Then, it holds that for all $j+1 < h < i$, $\mathcal{R}(\mathbf{x})'_{\beta+h\ell+1} = \mathcal{R}(\mathbf{x})'_{\beta+h\ell}$ and the error position in $\mathcal{R}(\mathbf{x})'$, say k , satisfies $\frac{k-\beta-1}{\ell} \in \{j+1, j+2, \dots, i\}$.*

Example 13. *We reconsider $\mathcal{R}_{3,1}(\mathbf{v})'$ from Example 12. From $\Delta^0(\mathbf{v})' = (1, 1^{-1}2, 1^{-1})$, we reconstruct $\widetilde{C}^0(\mathbf{v}) = (1, 2)$ and $\widetilde{C}^0(\mathbf{v}) = (1^2 2^{-1}, 1)$. Since $\widetilde{C}^0(\mathbf{v}) \in \Sigma_3^2$ and $\widetilde{v}_1 \notin \Sigma_3$, we set*

$i = 2$ and $j = 0$ in accordance with Lemma 13. Thus, either $\mathcal{R}(\mathbf{x})'_4$ or $\mathcal{R}(\mathbf{v})'_7$ is noisy, implying that $\mathbf{v} = (1, 2, 0, 1, 2, 2)$ or $\mathbf{v} = (1, 2, 0, 2, 1, 2)$ respectively.

Lemma 13 suggests that attempting reconstruction with a noisy read sub-derivative may help narrow down the error location even further. This finally allows us to arrive at

Theorem 6. *For $\ell \geq 3$, $\mathcal{C}(n, \ell)$ is a single-substitution read code.*

Proof: Let $\mathcal{R}(\mathbf{x})'$ arise from a single substitution on $(\ell, 1)$ -read vector of some $\mathbf{x} \in \mathcal{C}(n, \ell)$. In light of Lemma 11, this proof is dedicated to composition errors of the form ab^{-1} .

Upon identifying $\alpha, \beta \in \Sigma_\ell$ where $\alpha \equiv \beta + 1 \pmod{\ell}$, such that $\prod_i \Delta^\beta(\mathbf{x})'_i = \prod_i \Delta^\alpha(\mathbf{x})_i^{-1} \neq c(\phi)$, we attempt reconstruction with $\Delta^\beta(\mathbf{x})'$ from left to right and from right to left to obtain $\widetilde{C}^\beta(\mathbf{x})$ and $C^\beta(\mathbf{x})$ respectively, and define indices i and j according to Lemma 13. Since for all $j+1 < h < i$, $\mathcal{R}(\mathbf{x})'_{\beta+h\ell+1} \cdot \mathcal{R}(\mathbf{x})'_{\beta+h\ell}^{-1} = \phi$, and a run of ' ϕ 's in $\Delta^\beta(\mathbf{x})'$ can be of length at most $2\lceil \log(qn) \rceil - 1$ as a consequence of the run-length constraint in $\mathcal{C}(n, \ell)$ and Lemma 1, we infer that $i - j - 2 \leq 2\lceil \log(qn) \rceil - 1$.

From Lemma 13, we know that the error exists somewhere in $(\mathcal{R}(\mathbf{x})'_{\beta+(j+1)\ell+1}, \mathcal{R}(\mathbf{x})'_{\beta+(j+2)\ell+1}, \dots, \mathcal{R}(\mathbf{x})'_{\beta+i\ell+1})$, which is evidently a substring of $\mathcal{R}^\pi(\mathbf{x})'$ and has a length of at most $2\lceil \log(qn) \rceil + 1$. Since an error of the form ab^{-1} , where $a \neq b$, surely reflects as a single substitution in $|\mathcal{R}^\pi(\mathbf{x})'|_1 \bmod q$, which belongs to a code that corrects a substitution error localized to a window of $2\lceil \log(qn) \rceil + 1$ symbols, we can uniquely recover $|\mathcal{R}^\pi(\mathbf{x})|_1 \bmod q$, and by Corollary 2, also \mathbf{x} . ■

Since the preceding theorem establishes $\mathcal{C}(n, \ell)$ as a single-substitution read code, we now propose a specific instantiation of it, by means of the following realization of an a -bounded single-substitution-correcting code.

Definition 12. *For any $p > 0$ define*

$$\mathbf{H} = \underbrace{[\mathbf{H}_p \quad \mathbf{H}_p \quad \dots \quad \mathbf{H}_p]}_{\frac{n(q-1)}{q^{p-1}} \text{ times}}$$

where \mathbf{H}_p represents the parity-check matrix of a Hamming code² of order p . Here, if $\frac{q^p-1}{q-1}$ does not divide n , we append as many additional columns of \mathbf{H}_p from the right, so that $\mathbf{H} \in \Sigma_q^{p \times n}$. Then, for any $\mathbf{s} \in \Sigma_q^n$, let

$$\mathcal{C}_s \triangleq \{\mathbf{x} \in \Sigma_q^n : \mathbf{H}\mathbf{x} = \mathbf{s}\}.$$

Theorem 7. *For every $\mathbf{s} \in \Sigma_q^p$, $\mathcal{C}_s \in \text{BS}(n, a)$ where $a = \frac{q^p-1}{q-1}$ (i.e., \mathcal{C}_s is an a -bounded single-substitution-correcting code).*

Proof: Let $\mathbf{v} \in \Sigma_q^n$ denote an erroneous received word that results from a single substitution in a codeword of \mathcal{C}_s . In particular, the substitution error is known to have occurred at one of the indices in $\{i, i+1, \dots, i+a-1\}$, where $i \in [n-a+1]$.

² \mathbf{H}_p forms a projective representative (up to a scalar multiple) of all non-zero vectors in Σ_q^p .

$$\begin{array}{l}
\Delta^\beta(\mathbf{x})' = (\dots \quad \phi \quad \text{Error } db^{-1} \quad \phi \quad \dots) \\
\text{Left-to-right reconstruction } \widehat{C}^\beta(\mathbf{x}) = (\dots \quad a \quad b \quad b \quad \dots \quad b \quad d \quad d \quad \dots \quad d \quad \text{dcb}^{-1} \quad \dots) \\
\text{Right-to-left reconstruction } \widetilde{C}^\beta(\mathbf{x}) = (\dots \quad \dots \quad \dots \quad \text{bd}^{-1} \quad b \quad \dots \quad b \quad c \quad \dots)
\end{array}$$

i
 j

Fig. 5. Illustration of Lemma 13. The red entry in $\Delta^\beta(\mathbf{x})'$ indicates the location of the composition substitution that replaced ϕ with db^{-1} . Thus attempts at reconstructing $(x_{\beta+1}, x_{\beta+\ell+1}, \dots)$ leads to erroneous estimates. In particular, left-to-right and right-to-left reconstruction via Corollary 1 yields symbol estimates $\notin \Sigma_q$, at the indices i and j respectively, that are marked in blue.

Consider the matrix \mathbf{H}' formed by extracting the columns of \mathbf{H} at indices $i, i+1, \dots, i+a-1$. While for $i \bmod \frac{q^p-1}{q-1} = 1$ it evidently holds that $\mathbf{H}' = \mathbf{H}_p$, in all other cases, \mathbf{H}' is simply a permutation of the columns of \mathbf{H}_p . Hence, \mathbf{H}' always corresponds to a q -ary Hamming code of order p , implying that the decoder can correct a single substitution in \mathbf{v} in the aforementioned length- a window. More precisely, if $\mathbf{H}'\mathbf{v}_i^{i+a-1} - \mathbf{s}$ is a scalar multiple of the j -th column of \mathbf{H}' , then the decoder concludes that the error location is $i+j-1$ and its value is given by said multiple. ■

Remark: Note that for $q=2$, Construction 1 with \mathcal{C}_s as the a -bounded single-substitution-correcting code, is similar to that defined in Construction 1 of the conference version of this work.

This particular choice of an a -bounded single-substitution-correcting code in $\mathcal{C}(n, \ell)$ implies the following upper bound on its minimum required redundancy.

Lemma 14. *The minimum required redundancy of $\mathcal{C}(n, \ell)$ is at most*

$$\log \log n + \log \left(2(q-1) + \frac{5q-4}{\log n} \right) + 2.$$

Proof: Observe that the a -RLL $_q(n)$ constraint as specified in Definition 10 is equivalent to the (d, k) -RLL constraint [22], i.e., restricting each zero run to be of length at least $d=0$ and at most $k=a-1$ (see [23]). Now since the first constraint in Construction 1 implies that $C_{\ell,1}^0(\mathbf{x}) \circ \dots \circ C_{\ell,1}^{\ell-1}(\mathbf{x})$ belongs to a superset of $\lceil \log qn \rceil$ -RLL (n) , we deduce that this run-length restriction necessitates a redundancy of at most one symbol, as indicated by [23, Section III-B].

Next, set $p = \lceil \log(2(q-1)\lceil \log(qn) \rceil + q) \rceil$ and observe $\frac{q^p-1}{q-1} \geq 2\lceil \log(qn) \rceil + 1$. For any $\mathbf{s} \in \Sigma_q^p$, then, we may use \mathcal{C}_s as a $(2\lceil \log(qn) \rceil + 1)$ -bounded single-substitution-correcting code of length $n + \ell - 1$ in Construction 1, and denote the resulting code by $\mathcal{C}_s(n, \ell)$. In particular, \mathcal{C}_s corrects a single substitution error in any contiguous window of length $2\lceil \log(qn) \rceil + 1$ symbols in $|\mathcal{R}^\pi(\mathbf{x})|_1 \bmod q$.

Note that for all choices of $\mathbf{s} \in \Sigma_q^p$, the respective codes \mathcal{C}_s are pairwise disjoint and collectively, they partition the entire space $\Sigma_q^{n+\ell-1}$. Thus, using the pigeonhole principle, we observe that

$$q^{n-1} \leq |\lceil \log qn \rceil\text{-RLL}(n)|$$

$$\begin{aligned}
&\leq \left| \left\{ \mathbf{x} \in \Sigma_q^n : C_{\ell,1}^i(\mathbf{x}) \in \lceil \log qn \rceil\text{-RLL}_q(k_i) \right. \right. \\
&\quad \left. \left. \forall i \in \{0, 1, \dots, \ell-1\} \right\} \right| \\
&= \left| \bigcup_{\mathbf{s} \in \Sigma_q^p} \mathcal{C}_s(n, \ell) \right| = \sum_{\mathbf{s} \in \Sigma_q^p} |\mathcal{C}_s(n, \ell)|,
\end{aligned}$$

and therefore there exists a choice of $\mathbf{s} \in \Sigma_q^p$ for which $|\mathcal{C}_s(n, \ell)| \geq q^{n-1-p}$, i.e., it requires at most $p+1$ redundant symbols. Finally, since we have

$$\begin{aligned}
p &= \lceil \log(2(q-1)\lceil \log(qn) \rceil + q) \rceil \\
&= \log(2(q-1)\log(q^2n) + q) + 1 \\
&= \log(2(q-1)\log n + 4(q-1) + q) + 1 \\
&= \log \log n + \log \left(2(q-1) + \frac{5q-4}{\log n} \right) + 1,
\end{aligned}$$

the statement of the lemma follows. ■

V. CONCLUSION

The primary objective of this work was to initiate a line of research dedicated to error-correcting codes that attempt to incorporate the dominant physical aspects of nanopore sequencing. The channel model we adopted incorporates the intersymbol interference aspect of the sequencer as a window that slides over the incoming DNA strand and outputs the composition of the corresponding substrings in this strand. The measurement noise in the current readout is modeled as substitution errors in the resulting vector of compositions. We observed how, in doing so, the correction of a single substitution can be accomplished with $\log \log n + O(1)$ redundant symbols instead of $\log n$ symbols necessitated by the standard case, i.e., when the decoder is agnostic to the channel model. This result understandably encourages us to further investigate this channel model under multiple substitution errors as well as more error settings, e.g., deletions and duplications. Examining this channel in the context of Levenshtein's sequence reconstruction problem is also an exciting avenue to pursue.

ACKNOWLEDGMENTS

The authors are grateful for valuable suggestions, given by the two anonymous reviewers and associate editor, which greatly improved the readability of this paper.

REFERENCES

- [1] D. Deamer, M. Akeson, and D. Branton, “Three decades of nanopore sequencing,” *Nature Biotechnology*, vol. 34, no. 5, pp. 518–524, May 2016.
- [2] A. H. Laszlo, I. M. Derrington, B. C. Ross, H. Brinkerhoff, A. Adey, I. C. Nova, J. M. Craig, K. W. Langford, J. M. Samson, R. Daza, K. Doering, J. Shendure, and J. H. Gundlach, “Decoding long nanopore sequencing reads of natural DNA,” *Nature Biotechnology*, vol. 32, no. 8, pp. 829–833, Aug. 2014.
- [3] J. Kasianowicz, E. Brandin, D. Branton, and D. Deamer, “Characterization of individual polynucleotide molecules using a membrane channel,” *Proc. Natl. Acad. Sci.*, vol. 93, no. 24, pp. 13 770–13 773, Nov. 1996.
- [4] W. Mao, S. N. Diggavi, and S. Kannan, “Models and information-theoretic bounds for nanopore sequencing,” *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3216–3236, Apr. 2018.
- [5] R. Hulett, S. Chandak, and M. Wootters, “On coding for an abstracted nanopore channel for DNA storage,” in *Proc. IEEE Intl. Symp. on Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 2021, pp. 2465–2470.
- [6] B. McBain, E. Viterbo, and J. Saunderson, “Finite-state semi-markov channels for nanopore sequencing,” in *Proc. IEEE Intl. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 216–221.
- [7] Y. M. Chee, A. Vardy, V. K. Vu, and E. Yaakobi, “Transverse-read-codes for domain wall memories,” *IEEE J. Sel. Areas Inf. Theory.*, vol. 4, pp. 784–793, 2023.
- [8] J. Chrisnata, H. M. Kiah, and E. Yaakobi, “Correcting deletions with multiple reads,” *IEEE Trans. Inf. Theory*, vol. 68, no. 11, pp. 7141–7158, Nov. 2022.
- [9] V. Levenshtein, “Efficient reconstruction of sequences,” *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 2–22, Jan. 2001.
- [10] V. I. Levenshtein, “Efficient Reconstruction of Sequences from Their Subsequences or Supersequences,” *J. Combin. Theory Ser. A*, vol. 93, no. 2, pp. 310–332, Feb. 2001.
- [11] M. Abu-Sini and E. Yaakobi, “On Levenshtein’s Reconstruction Problem Under Insertions, Deletions, and Substitutions,” *IEEE Trans. Inf. Theory*, vol. 67, no. 11, pp. 7132–7158, Nov. 2021.
- [12] T. Batu, S. Kannan, S. Khanna, and A. McGregor, “Reconstructing strings from random traces,” in *Proc. Fifteenth Annual ACM-SIAM Symp. Disc. Alg.*, ser. SODA ’04. USA: Society for Industrial and Applied Mathematics, Jan. 2004, pp. 910–918.
- [13] V. L. Phuoc Pham, K. Goyal, and H. M. Kiah, “Sequence Reconstruction Problem for Deletion Channels: A Complete Asymptotic Solution,” in *Proc. IEEE Intl. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 992–997.
- [14] R. Gabrys and E. Yaakobi, “Sequence Reconstruction Over the Deletion Channel,” *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2924–2931, Apr. 2018.
- [15] Y. Yehezkeally and M. Schwartz, “Reconstruction Codes for DNA Sequences With Uniform Tandem-Duplication Errors,” *IEEE Trans. Inf. Theory*, vol. 66, no. 5, pp. 2658–2668, May 2020.
- [16] D. E. Knuth, “The sandwich theorem,” *The Electronic Journal of Combinatorics*, vol. 1, no. 1, p. A1, Apr. 1994.
- [17] G. M. Church, Y. Gao, and S. Kosuri, “Next-Generation Digital Information Storage in DNA,” *Science*, vol. 337, no. 6102, pp. 1628–1628, Sep. 2012.
- [18] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipo, and E. Birney, “Towards Practical, High-Capacity, Low-Maintenance Information Storage in Synthesized DNA,” *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013.
- [19] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen, C. N. 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*, vol. 36, no. 3, pp. 242–248, Mar. 2018.
- [20] S. M. H. T. Yazdi, H. M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, “DNA-Based Storage: Trends and Methods,” *IEEE Trans. Mol., Bio. and Multi-Scale Commun.*, vol. 1, no. 3, pp. 230–248, Sep. 2015.
- [21] K. Cai, H. M. Kiah, T. T. Nguyen, and E. Yaakobi, “Coding for Sequence Reconstruction for Single Edits,” *IEEE Trans. Inf. Theory*, vol. 68, no. 1, pp. 66–79, Jan. 2022.
- [22] B. H. Marcus, R. M. Roth, and P. H. Siegel, “An Introduction to Coding for Constrained Systems,” Oct. 2001, unpublished Lecture Notes. [Online]. Available: www.math.ubc.ca/~marcus/Handbook
- [23] M. Levy and E. Yaakobi, “Mutually uncorrelated codes for DNA storage,” *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3671–3691, Jun. 2019.

APPENDIX

A. Proof of Lemma 6, non-binary extension of [8, Lemma 7]

Proof: Since all singletons are cliques, we endeavor to show that for all $\gamma = (\mathbf{u}, \mathbf{w}, a, b) \in \Gamma$, $\mathcal{Q}_\gamma^{(0)}$ is a clique. The proof for $\mathcal{Q}_\gamma^{(1)}$ follows similarly.

For any two vectors in $\mathcal{Q}_\gamma^{(0)}$, say $\mathbf{x} = \mathbf{u}(ab)^i(ba)^{p-i}\mathbf{w}$ and $\mathbf{y} = \mathbf{u}(ab)^j(ba)^{p-j}\mathbf{w}$, we may assume $i < j$ without loss of generality, and observe that

$$\begin{aligned}\mathbf{x} &= \mathbf{u}(ab)^i(ba)^{j-i}(ba)^{p-j}\mathbf{w}, \\ \mathbf{y} &= \mathbf{u}(ab)^i(ab)^{j-i}(ba)^{p-j}\mathbf{w}.\end{aligned}$$

By Definition 7, \mathbf{x} and \mathbf{y} are clearly adjacent, implying that $\mathcal{Q}_\gamma^{(0)}$ is a clique.

Now to show that each vector $\mathbf{x} \in \Sigma_q^{2pm}$ belongs to at least one clique in $\mathcal{Q}(m, p)$, note that we either have $\mathbf{x} \in \tilde{\Lambda}_p^m$, or one of the m subblocks of \mathbf{x} lies in $\Lambda_{p,a,b}$, for some $a, b \in \Sigma_q$. In the former case, \mathbf{x} constitutes a singleton and is accounted for by $\mathcal{Q}(m, p)$, while in the latter case, assuming that the i th subblock is the first that lies in $\Lambda_{p,a,b}$, we deduce that \mathbf{x} belongs to the clique $\mathcal{Q}(\mathbf{u}, \mathbf{w}, a, b)$ where $\mathbf{x}_1^{2p(i-1)} = \mathbf{u} \in \tilde{\Lambda}_{p,a,b}^{i-1}$, while $\mathbf{x}_{2pi+1}^{2pm} = \mathbf{w} \in \Sigma_q^{2p(m-i)}$. ■

B. Non-binary extension of [8, Lemma 9]

Lemma 15. For $p = \lfloor \frac{1}{2}(1 - \epsilon) \log(n) \rfloor$, we have $\lim_{n \rightarrow \infty} p \left(1 - \frac{2p}{q^{2p}}\right)^{\lfloor \frac{n}{2p} \rfloor} = 0$.

Proof: Based on $1 - x \leq e^{-x}$ we observe

$$\begin{aligned}& p \left(1 - \frac{2p}{q^{2p}}\right)^{\lfloor n/2p \rfloor} \\ & \leq p \exp\left(-\frac{2p}{q^{2p}} \left(\frac{n}{2p} - 1\right)\right) \\ & = p \exp\left(-\frac{n - 2p}{q^{2 \lceil \frac{1}{2}(1-\epsilon) \log(n) \rceil}}\right) \\ & \leq p \exp\left(-\frac{n - 2p}{q^{2n}} n^\epsilon\right) \\ & \leq \log(n) \cdot \exp\left(-\frac{n - \log(n)}{q^{2n}} n^\epsilon\right) \xrightarrow{n \rightarrow \infty} 0.\end{aligned}$$

Anisha Banerjee (S’22) received the B.E. degree in electrical engineering from Jadavpur University, India, and the M.Sc. degree (Hons.) in communications engineering from the Technical University of Munich (TUM), Munich, Germany, in 2018 and 2021 respectively. She is currently pursuing the Ph.D. degree with the Coding and Cryptography Group, Institute of Communications Engineering, TUM, under the supervision of Prof. Wachter-Zeh. Her research interests include coding theory and information theory and their applications to storage, with a special focus on insertion and deletion errors.

Yonatan Yehezkeally (S'12–M'20) received the B.Sc. degree (*cum laude*) in mathematics and the M.Sc. (*summa cum laude*) and Ph.D. degrees in electrical and computer engineering from Ben-Gurion University of the Negev, Beer-Sheva, Israel, in 2013, 2017, and 2020 respectively. He is currently a Carl Friedrich von Siemens Post-Doctoral Research Fellow of the Alexander von Humboldt Foundation, with the Associate Professorship of Coding and Cryptography (Prof. Wachter-Zeh), School of Computation, Information and Technology, Technical University of Munich. His research interests include coding theory and algorithms, particularly with applications to novel storage media, with a focus on DNA-based storage and nascent sequencing technologies. They further include combinatorial analysis and structures, as well as algebraic structures.

Antonia Wachter-Zeh (S'10–M'14–SM'20) received the M.Sc. degree in communications technology from Ulm University, Germany, and the Ph.D. degree from Ulm University and from Université de Rennes 1, Rennes, France, in 2009 and 2013 respectively. She is an Associate Professor at the Technical University of Munich (TUM), Munich, Germany, in the School of Computation, Information and Technology. From 2013 to 2016, she was a Post-Doctoral researcher at the Technion — Israel Institute of Technology, Haifa, Israel, and from 2016 to 2020 a Tenure-Track Assistant Professor at TUM. Her research interests are coding theory, cryptography and information theory and their application to storage, communications, privacy, security and machine learning. She is a recipient of the DFG Heinz Maier-Leibnitz-Preis and of an ERC Starting Grant. She is currently an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY.

Eitan Yaakobi (S'07–M'12–SM'17) received the B.A. degree in mathematics, the B.A. degree in computer science, and the M.Sc. degree in computer science from the Technion — Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California at San Diego, La Jolla, CA, USA, in 2011. He is currently an Associate Professor with the Department of Computer Science, Technion — Israel Institute of Technology. He also holds a courtesy appointment with the Department of Electrical and Computer Engineering (ECE), Technion — Israel Institute of Technology. From 2011 to 2013, he was a Post-Doctoral Researcher with the Department of Electrical Engineering, California Institute of Technology, and the Center for Memory and Recording Research, University of California at San Diego. Since 2016, he has been with the Center for Memory and Recording Research, University of California at San Diego. From 2018 to 2022, he was with the Institute of Advanced Studies, Technical University of Munich, where he held a four-year Hans Fischer Fellowship, funded by the German Excellence Initiative and the EU 7th Framework Program. From August 2023 to January 2024, he was a Visiting Associate Professor at the School of Physical and Mathematical Sciences at Nanyang Technological University. His research interests include information and coding theory with applications to non-volatile memories, associative memories, DNA storage, data storage and retrieval, and private information retrieval. He was a recipient of several grants, including the ERC Consolidator Grant and the EIC Pathfinder Challenge. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship from 2010 to 2011. From 2020 to 2023, he served as an Associate Editor for Coding and Decoding for the IEEE TRANSACTIONS ON INFORMATION THEORY.