

Improved Coding Over Sets for DNA-Based Data Storage

Hengjia Wei¹ and Moshe Schwartz², *Senior Member, IEEE*

Abstract—Error-correcting codes over sets, with applications to DNA storage, are studied. The DNA-storage channel receives a set of sequences, and produces a corrupted version of the set, including sequence loss, symbol substitution, symbol insertion/deletion, and limited-magnitude errors in symbols. Various parameter regimes are studied. New bounds on code parameters are provided, which improve upon known bounds. New codes are constructed, at times matching the bounds up to lower-order terms or small constant factors.

Index Terms—Error-correcting codes, DNA storage, coding over sets.

I. INTRODUCTION

DUE to recent developments in DNA sequencing and synthesis technologies, storing data in DNA strands has gained a lot of interest in recent years. One notable feature of DNA-based storage is its ultrahigh storage densities of 10^{15} – 10^{20} bytes per gram of DNA, as demonstrated in recent experiments (see [21, Table 1]). Additionally, a DNA strand is easy to maintain and remains stable over millennia. These features make the DNA strand a suitable medium to store massive amounts of data.

DNA strands can be treated as sequences composed of four types of nucleotides A, T, G, and C. In order to produce or read the strands with an acceptable error rate, the lengths of the synthetic DNA strands cannot be too long, usually hundreds of nucleotides. Thus, the data in a DNA storage system is stored as a set of relatively short strands, each of which holds a fraction of the whole data. These short DNA strands are stored inside a solution and do not preserve the order in which they were stored. The goal of the sequencer is to read these strands and reconstruct the data without knowledge of the order of the sequences, even in the presence of errors.

The unordered manner of data storing in DNA storage systems motivates the study of *coding problem over sets*, following several papers on this topic [3], [6], [9], [10], [15]–[17], [20]. In [10], the authors studied the storage model where the errors are a combination of loss of sequences, as well as symbol errors inside the sequences, such as insertions, deletions, and substitutions.

Manuscript received September 18, 2020; revised April 12, 2021; accepted September 28, 2021. Date of publication October 13, 2021; date of current version December 23, 2021. This work was supported in part by the Israel Science Foundation (ISF) under Grant 270/18. (*Corresponding author: Hengjia Wei.*)

The authors are with the School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 8410501, Israel (e-mail: hjwei05@gmail.com; schwartz@ee.bgu.ac.il).

Communicated by E. Yaakobi, Associate Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2021.3119584

Some lower and upper bounds were derived on the cardinality of optimal error-correcting codes that are suitable for this model. Several explicit code constructions are also proposed for various error regimes. Later, [16], [17] adapted the model of [10]. In [16], it was assumed that no sequences are lost and a given number of symbol substitutions occur. Codes which have logarithmic redundancy in both the number of sequences and the length of the sequences have been proposed therein. In [17], a new metric was introduced to establish a uniform framework to combat both sequence loss and symbol substitutions, and Singleton-like and Plotkin-like bounds on the cardinality of optimal codes were derived. A related model was discussed in [6], where unordered multisets are received and errors are counted by sequences, no matter how many symbol errors occur inside the sequences. [9], [15] discussed the indexing technique to deal with the unordered nature of DNA storage. Additionally, codes that can be used as primer addresses were proposed in [3], [20] to equip the DNA storage system with random-access capabilities.

In this paper, we continue the study of coding over sets. We follow the model of [10] and present improved bounds and constructions. We also extend the error model to include limited-magnitude errors, following the recent application presented in [5]. Our main contributions are:

- 1) We derive some new lower bounds on the redundancy of codes which can protect against *substitutions* or *deletions*. These results, together with some existence results, demonstrate that correcting deletions requires fewer redundancy bits than correcting substitutions. Note that a similar observation was made in [10], but only in the regime where there is no sequence loss and only a single symbol error occurs, whereas our results are proved for two broad parameter ranges.
- 2) We propose several *explicit* constructions of codes having redundancy that is logarithmic in the number of sequences M , whereas the corresponding explicit constructions in [9], [10] require $\Theta(M^c)$ bits of redundancy with $c > 0$ a constant number.
- 3) We also study another error model, where data is represented by vectors of integers that may suffer from *limited-magnitude errors* in some of their entries. This model is motivated by a recently proposed method of encoding information in DNA sequences which can optimize the amount of information bits per synthesis time unit [5]. We utilize our explicit code constructions for substitutions to combat limited-magnitude errors.

TABLE I
LOWER AND UPPER BOUNDS ON THE REDUNDANCY OF OPTIMAL $(s, t, \varepsilon)_{\mathbb{T}}$ -CORRECTING CODES. LOW ORDER TERMS ARE OMITTED

Channel	Previous Lower Bound	Ref.	Imp. Lower Bound	Ref.	Upper Bound	Ref.
$(s, t, \bullet)_{\mathbb{L}}$	$(s+t)L + t \log M$	[10, Cor. 1]	$(s+2t)L$	Cor. 5	$(s+2t)L$	[10, Const. 2]
$(s, t, \bullet)_{\mathbb{S}}$			$(s+2t)L$	Cor. 5	$(s+2t)L$	[10, Const. 2]
$(s, t, \bullet)_{\mathbb{D}}$			$(s+t)L$	Cor. 7	$(s+t)L$	[10, Const. 2]
$(0, t, \varepsilon)_{\mathbb{S}}$	$t \log M + t\varepsilon \log L$	[10, Thm. 7]			$2t \log M + 2t\varepsilon \log L$	[10, Thm. 3]
$(0, t, \varepsilon)_{\mathbb{D}}$	$t\varepsilon \log L$	[10, Thm. 9]	$\lfloor t/2 \rfloor \log M$	Thm. 15	$t \log M + 2t\varepsilon \log(L/2)$ $4\varepsilon \log L$ when $t = 1$	[10, Thm. 4] Thm. 14

TABLE II

REDUNDANCY OF THE CODE CONSTRUCTIONS. LOW ORDER TERMS ARE OMITTED. THE SYMBOL * MEANS EXPLICIT ENCODING FOR THE CORRESPONDING CONSTRUCTION IS UNKNOWN. IN THE FIRST ROW, $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}, \mathbb{L}\}$, AND $\delta = s + 2t$ IF $\mathbb{T} \in \{\mathbb{L}, \mathbb{S}\}$; $\delta = s + t$ IF $\mathbb{T} = \mathbb{D}$. IN THE SECOND ROW, r_o DENOTES THE REDUNDANCY OF AN $(s, 0, 0)_{\mathbb{T}}$ -CORRECTING CODE OF \mathbb{X}_M^{2, L_o} , WHILE $r_{\mathbb{T}}$ DENOTES THE REDUNDANCY OF A BLOCK-CODE OF DIMENSION L_o AND LENGTH L THAT CAN CORRECT s ERRORS OF TYPE \mathbb{T}

Channel	Previous Construction	Ref.	Imp. Construction	Ref.
$(s, t, \bullet)_{\mathbb{T}}$	$M \log e + \delta(L - \lceil \log M \rceil)$	[10, Const. 1]		
	δL	[10, Const. 2]*	$\delta L + 4\delta \log \log M$	Cor. 9
	$\frac{1-c}{2} M^c \log M + \delta M^{1-c}(L - \log M)$	[10, Const. 3]		
$(s, M-s, \varepsilon)_{\mathbb{S}}$	$Mr_{\mathbb{S}} + r_o$	[10, Const. 4]	$M\varepsilon(\lceil \log L \rceil + 1) + sL + 4s \log \log M$	Cor. 11
$(s, M-s, \varepsilon)_{\mathbb{D}}$	$Mr_{\mathbb{D}} + r_o$	[10, Const. 4]	$4M\varepsilon \log L + sL + o(M \log L)$	Cor. 12
$(0, t, \varepsilon)_{\mathbb{S}}$	$M \log e + 4t \log M + 2t\varepsilon \log L$	[9, Thm. 2]	$(8t+2) \log M + 2t\varepsilon \lceil \log L \rceil + 8t \log \log M$	Cor. 17
$(0, 1, \varepsilon)_{\mathbb{D}}$	$\log L$ for $\varepsilon = 1$	[10, Const. 5]	$4\varepsilon \log L$ for any given ε	Thm. 14*

A summary of the bounds and constructions appearing in this paper, and a comparison with previous results, is given in Table I and Table II.

The remainder of the paper is organized as follows. In Section II we provide the notation and definitions used throughout the paper. In Section III we consider channels with a fixed number of lost sequences, and a fixed number of erroneous sequences. Section IV studies codes for a channel with no sequence loss. We then study codes when the errors are of limited magnitude in Section V.

II. PRELIMINARIES

For a positive integer $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, 2, \dots, n\}$. For $q \in \mathbb{N}$, we use Σ_q to denote a finite alphabet with q elements, \mathbb{Z}_q to denote the cyclic group of integers with addition modulo q , and \mathbb{F}_q to denote the finite field of size q . Throughout the paper, we denote the base- q logarithm of a real number $a \in \mathbb{R}$ by $\log_q a$, and we omit the subscript if $q = 2$.

For a sequence $\mathbf{a} = (a_1, \dots, a_n) \in \Sigma_q^n$, let $\mathbf{a}[i]$ denote the i th symbol a_i and $\mathbf{a}[i, j]$ denote the subword of \mathbf{a} starting at position i and ending at position j . We use $|\mathbf{a}|$ to denote the length of \mathbf{a} . For two sequences \mathbf{a} and \mathbf{b} , we use (\mathbf{a}, \mathbf{b}) to denote the concatenation of \mathbf{a} and \mathbf{b} . Fix an ordering of the sequences of Σ_q^n . Then every size- M subset $S \subseteq \Sigma_q^n$ can be represented by a binary vector $\mathbb{1}(S)$, termed the *characteristic vector*, of length q^n and weight M , where each non-zero entry indicates that the corresponding element is contained in S .

A. The DNA Storage Channel

In a DNA-based data storage system, data is stored as an (unordered) set

$$S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\} \subseteq \Sigma_q^L$$

of M distinct sequences \mathbf{x}_i , $i \in [M]$. In practice, the length of the sequences L is in the order of a few hundreds, while M is significantly larger. A summary of typical values of L and M can be found in [10, Table I]. In general, we assume that $M = q^{\beta L}$ for some $0 < \beta < 1$. For the sake of simplicity, we further assume βL , i.e., $\log_q M$, is an integer. Otherwise, a floor or ceiling is to be used in certain places, making notation cumbersome and changing nothing in the asymptotic analysis.

We study the $(s, t, \varepsilon)_{\mathbb{T}}$ -DNA storage channel model defined in [10]. In this channel, the sequences in S are drawn arbitrarily and sequenced, possibly with symbol errors, and we have the following assumptions on the errors:

- 1) the maximum number of sequences never drawn is s ;
- 2) the maximum number of sequences with errors is t ;
- 3) each sequence suffers at most ε errors of type \mathbb{T} .

Note that erroneous sequences are not necessarily distinct from each other or from the correct sequences, and that would result in sequence losses. Thus, the output of the channel is a subset S' of at least $M - s - t$ sequences of S with t (or fewer) sequences each suffering ε (or fewer) errors of type \mathbb{T} .

In [10], the authors mainly discuss the following types of errors: substitutions (\mathbb{S}), deletions (\mathbb{D}), and a combination of substitutions, deletions and insertions (\mathbb{L}), that is, $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}, \mathbb{L}\}$. In this paper, apart from the errors mentioned

above, we also discuss *limited-magnitude errors* (LM), the model of which will be described and explained in the next subsection.

Denote

$$\mathcal{X}_M^{q,L} \triangleq \{S \subseteq \Sigma_q^L \mid |S| = M\}.$$

For each $S \in \mathcal{X}_M^{q,L}$, the *error ball* $B_{s,t,\varepsilon}^{\mathbb{T}}(S)$ is defined to be the set of all possible received S' with S being the input of the $(s, t, \varepsilon)_{\mathbb{T}}$ -DNA storage channel. We say a subset $\mathcal{S} \subseteq \mathcal{X}_M^{q,L}$ is an $(s, t, \varepsilon)_{\mathbb{T}}$ -*correcting code* if for any distinct $S_1, S_2 \in \mathcal{S}$, it always holds that

$$B_{s,t,\varepsilon}^{\mathbb{T}}(S_1) \cap B_{s,t,\varepsilon}^{\mathbb{T}}(S_2) = \emptyset.$$

When $\varepsilon = L$, such a code is also called an $(s, t, \bullet)_{\mathbb{T}}$ -*correcting code*. The *redundancy* of the code \mathcal{S} is defined to be

$$\log_q |\mathcal{X}_M^{q,L}| - \log_q |\mathcal{S}| = \log_q \binom{q^L}{M} - \log_q |\mathcal{S}|.$$

In Section III, we study $(s, t, \bullet)_{\mathbb{T}}$ -correcting codes with $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}, \mathbb{L}\}$, and in Section IV we study $(0, t, \varepsilon)_{\mathbb{T}}$ -correcting codes with $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}\}$. Our results are presented in the binary case, and they can be easily generalized to the quaternary case, i.e., $\Sigma = \{A, T, C, G\}$. Table I summarizes the lower bounds and upper bounds on the redundancy of the optimal codes, while Table II summarizes our explicit code constructions. These two tables also include the corresponding results from [9], [10] for comparison. From Table I, we have the following observations:

- 1) For the redundancy of the optimal $(s, t, \bullet)_{\mathbb{T}}$ -correcting code, the lower bound almost attains the upper bound.
- 2) For the redundancy of the optimal $(0, t, \varepsilon)_{\mathbb{T}}$ -correcting code, the lower bound is nearly half as much as the upper bound.
- 3) For the sets of parameters (s, t, \bullet) or $(0, 1, \varepsilon)$, correcting deletions requires fewer redundancy bits than correcting substitutions.

B. Limited-Magnitude Error Model

Recently, a new inexpensive enzymatic method of DNA synthesis was proposed in [8]. Unlike other synthesis methods that focus on the synthesis of a precise DNA sequence, this method focuses on the synthesis of runs of homopolymeric bases. Specifically, the synthesis process proceeds in rounds. Assume at the beginning of the round, the current string is $\mathbf{u} \in \Sigma^*$. A letter $a \in \Sigma$ is chosen, which differs from the last letter of \mathbf{u} . A chemical reaction is then allowed to occur for a duration of $T \in \mathbb{N}$ time units. The resulting string at the end of the round is $(\mathbf{u}, \underbrace{a, a, \dots, a}_{\ell})$, where ℓ is a random variable

whose distribution depends on the new letter being appended, the last letter of the string at the beginning of the round, and the duration of the chemical reaction.

For the sake of simplicity, in this paper, we consider the binary case and assume that the last letter of the initiator is 0. Since long runs may affect the DNA molecule's stability, the encoder refrains from using runs that are too long. Let q denote the length of the longest run used by the encoder¹. Thus, every binary sequence produced by n rounds of synthesis process

¹This is different from Section II.A, where q is the alphabet size.

can be represented by a sequence $\mathbf{r} = (r_1, r_2, \dots, r_n)$ of \mathbb{Z}_q^n , where r_i represents the length of the run appended in the i th round.

Based on this enzymatic method of DNA synthesis, a new method of encoding information in DNA strands is proposed [5]. In this method, the data is encoded to a set of M sequences \mathbf{r}_i of \mathbb{Z}_q^L . Then binary sequences \mathbf{u}_i are produced by L rounds of synthesis process described above so that by controlling the chemical reaction, the run lengths of \mathbf{u}_i are the components of \mathbf{r}_i . In the system, what we store are these sequences \mathbf{u}_i , whereas the data is represented by the run-lengths of these sequences, i.e., $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_M\}$.

The chemical reaction may end up shorter or longer than planned, usually by a limited amount, due to variability in the molecule-synthesis process. Consequently, the sequence of the run lengths of \mathbf{u}_i is $\mathbf{r}_i + \mathbf{e}$, where $\mathbf{e} = (e_1, e_2, \dots, e_L) \in [-k_-, k_+]^L$ for some non-negative integers k_+, k_- . We say ε errors that are (k_+, k_-) -*limited-magnitude errors* (LM) occurred, if exactly ε of the entries of \mathbf{e} are non-zero. This kind of errors can also be found in other applications, like high-density recording [7], [11] and flash memories [1], and the conventional coding problem to protect against such errors has been extensively researched, e.g., see [18] and the references therein.

In this paper, we consider coding over sets in the presence of limited-magnitude errors. In this model, the codeword is still a subset $S = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_M\} \subseteq \mathbb{Z}_q^L$. However, each sequence \mathbf{r}_i in S represents the run-lengths of a sequence \mathbf{u}_i produced by L rounds of synthesis process. We note that these synthesized sequences \mathbf{u}_i 's have the same number of runs, i.e. L , but may have various lengths. With a codeword $S \subseteq \mathbb{Z}_q^L$ as input, the $(s, t, \varepsilon, k_+, k_-)_{\text{LM}}$ -DNA storage channel outputs a subsets S' of S with s (or fewer) sequences lost and t (or fewer) sequences being corrupted by at most ε (k_+, k_-) -limited-magnitude errors. The corresponding error-correcting code is called an $(s, t, \varepsilon, k_+, k_-)_{\text{LM}}$ -*correcting code*. In Section V, we propose a construction for such codes, which is based on $(0, t, \varepsilon)_{\mathbb{S}}$ -correcting codes. Some bounds on the redundancy are also derived. As before, the *redundancy* of a code $\mathcal{S} \subseteq \{S \subseteq \mathbb{Z}_q^L \mid |S| = M\}$ is defined to be

$$\log_q \binom{q^L}{M} - \log_q |\mathcal{S}|,$$

where $\binom{q^L}{M}$ is the maximum number of messages encoded by a set of M sequences that are synthesized by L rounds of process. We emphasize that in this model the channel receives as input M vectors of length L each, representing synthesis instructions for L rounds. The redundancy is measured in this space. However, inside the channel, these synthesis instructions are turned into DNA sequences. These may be of different lengths for two reasons: first, the sum of run-lengths may not be equal in all the vectors. Second, the noisy synthesis process may result in different run-lengths from those intended.

C. Some Useful Codes

Our constructions use the well-known Reed-Solomon codes and BCH codes as input (e.g., see [12]). In addition, we also require the following codes.

Lemma 1 ([14, Theorem 1]): For any sequence $\mathbf{c} \in \{0, 1\}^n$ and a fixed positive integer ε , there exists a hash function $\text{Hash}_\varepsilon : \{0, 1\}^n \rightarrow \{0, 1\}^{h_\varepsilon}$ with $h_\varepsilon = 4\varepsilon \log n + o(\log n)$, computable in $O(n^{2\varepsilon+1})$ time, such that

$$\{(\mathbf{c}, \text{Hash}_\varepsilon(\mathbf{c})) \mid \mathbf{c} \in \{0, 1\}^n\}$$

forms an ε -deletion-correcting code. The decoding complexity of the code is $O(n^{\varepsilon+1})$.

Lemma 2: Let ℓ, δ be positive integers such that $\delta \leq \frac{2^\ell - 1}{\ell + 1}$. Then there is a map $\text{enc} : \mathbb{F}_2^{2^\ell} \rightarrow \mathbb{F}_2^r$ with $r \leq \delta\ell + \delta$ such that the set

$$\{(\mathbf{m}, \text{enc}(\mathbf{m})) \mid \mathbf{m} \in \mathbb{F}_2^{2^\ell}\}$$

is a code of minimum Hamming distance $2\delta + 1$.

Proof: Let $n = 2^{\ell+1} - 1$. Since $\delta \leq \frac{2^\ell - 1}{\ell + 1}$, there is a binary $[n, n - \delta(\ell + 1), 2\delta + 1]$ BCH code. We may shorten this code and rearrange its coordinates to obtain a systematic $[2^\ell + \delta(\ell + 1), 2^\ell, 2\delta + 1]$ code, and then the conclusion follows. ■

III. $(s, t, \bullet)_\mathbb{T}$ -CORRECTING CODES

In this section, we study $(s, t, \bullet)_\mathbb{T}$ -correcting codes, where $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}, \mathbb{L}\}$. We give an improved lower bound on the redundancy of such codes, which asymptotically agrees with the upper bound in [10, Theorem 13] up to low-order terms. Then we give an explicit construction of codes whose redundancy is close to this bound.

A. Bounds Based on Constant-Weight Codes

Fix an ordering of the vectors of $\{0, 1\}^L$. For a subset S of $\{0, 1\}^L$, its characteristic vector, denoted $\mathbb{1}(S)$, is a binary vector of length 2^L where each symbol ‘1’ indicates that a specific vector is contained in the set S . In this way, a code $\mathcal{S} \subseteq \mathcal{X}_M^{2^L}$ can be represented by a binary constant-weight code

$$\mathcal{C}(\mathcal{S}) \triangleq \{\mathbb{1}(S) \mid S \in \mathcal{S}\},$$

where all the codewords have weight M .

The following result establishes the equivalence of an $(s, t, \bullet)_\mathbb{S}$ -correcting code and a constant-weight code of certain minimum distance.

Proposition 3: Let s and t be positive integers such that $s + t \leq M$. A code $\mathcal{S} \subseteq \mathcal{X}_M^{2^L}$ is an $(s, t, \bullet)_\mathbb{S}$ -correcting code if and only if the corresponding constant-weight code $\mathcal{C}(\mathcal{S})$ has minimum Hamming distance at least $2(s + t) + 2$.

Proof: Denote $\delta \triangleq s + 2t$. We first show that if $\mathcal{C}(\mathcal{S})$ has minimum distance $\geq 2\delta + 2$ then \mathcal{S} is an $(s, t, \bullet)_\mathbb{S}$ -correcting code. Note that for a codeword $S \in \mathcal{S}$, the s deletions and t substitutions of sequences in S correspond to at most $s + t$ asymmetric $1 \rightarrow 0$ errors and t asymmetric $0 \rightarrow 1$ errors in $\mathbb{1}(S)$. Thus, if $\mathcal{C}(\mathcal{S})$ has minimum Hamming distance $2\delta + 2$, we can correct these δ substitution errors in $\mathbb{1}(S)$, and then recover S . That is, \mathcal{S} is an $(s, t, \bullet)_\mathbb{S}$ -correcting code

In the other direction, if $\mathcal{C}(\mathcal{S})$ has minimum Hamming distance less than $2\delta + 2$, then it is at most 2δ since the distance between two sequences of the same weight is even. Let $\mathbb{1}(S)$ and $\mathbb{1}(S')$ be two codewords in $\mathcal{C}(\mathcal{S})$ with distance at most 2δ . Necessarily, S and S' share at least $M - \delta$ sequences. W.l.o.g., we may assume that

$$S = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{M-\delta}, \mathbf{a}_1, \dots, \mathbf{a}_s, \mathbf{b}_1, \dots, \mathbf{b}_t\} \in \mathcal{S}$$

and

$$S' = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{M-\delta}, \mathbf{a}'_1, \dots, \mathbf{a}'_s, \mathbf{b}'_1, \dots, \mathbf{b}'_t\} \in \mathcal{S}.$$

Note that in the $(s, t, \bullet)_\mathbb{S}$ -DNA storage channel, the result of an erroneous sequence could be any sequence of $\{0, 1\}^L$. Hence, after going through the channel, both S and S' can generate the set

$$\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{M-\delta}, \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_t, \mathbf{b}'_{t+1}, \dots, \mathbf{b}'_2\}.$$

This implies that \mathcal{S} is not an $(s, t, \bullet)_\mathbb{S}$ -correcting code. ■

The following upper bound on the size of a constant-weight code can be found in [4].

Lemma 4: Let \mathcal{C} be a binary constant-weight code of length n , weight w and minimum distance 2τ . Then

$$|\mathcal{C}| \leq \frac{\binom{n}{w-\tau+1}}{\binom{n}{w}}.$$

Corollary 5: Let s and t be positive integers such that $s + t \leq M$. For any $(s, t, \bullet)_\mathbb{S}$ -correcting code \mathcal{S} , the code size satisfies

$$|\mathcal{S}| \leq \frac{\binom{2^L}{M-s-2t}}{\binom{2^L}{M}}.$$

In particular, if both s and t are fixed, the redundancy of an $(s, t, \bullet)_\mathbb{S}$ -correcting code is at least

$$(s + 2t)L - \log((s + 2t)!) - o(1).$$

Proof: The first bound is obtained directly from Proposition 3 and Lemma 4. If s and t are both fixed, then δ is fixed, and the redundancy satisfies

$$\begin{aligned} & \log \binom{2^L}{M} - \log |\mathcal{S}| \\ & \geq \log \frac{\binom{2^L}{M} \binom{M}{M-\delta}}{\binom{2^L}{M-\delta}} \\ & = \log \frac{(2^L - M + 1)(2^L - M + 2) \cdots (2^L - M + \delta)}{\delta!} \\ & \geq \delta \log(2^L - M + 1) - \log(\delta!) \\ & = \delta L - \delta \log \frac{2^L}{2^L - M + 1} - \log(\delta!) \\ & = \delta L - \delta (\log e) \ln \left(1 + \frac{M - 1}{2^L - M + 1} \right) - \log(\delta!) \\ & \stackrel{(a)}{\geq} \delta L - \frac{(M - 1)\delta \log e}{2^L - M + 1} - \log(\delta!) \\ & \stackrel{(b)}{=} \delta L - \log(\delta!) - o(1), \end{aligned}$$

where (a) holds as $\ln(1 + x) \leq x$ for all $x > -1$, and (b) holds as $M = 2^{\beta L}$ for some constant $\beta < 1$. ■

Remark: Note that an $(s, t, \bullet)_\mathbb{L}$ -correcting code is also an $(s, t, \bullet)_\mathbb{S}$ -correcting code. According to Corollary 5, the redundancy of an $(s, t, \bullet)_\mathbb{L}$ -correcting code is at least

$$(s + 2t)L - \log((s + 2t)!) - o(1),$$

which improves upon the bound $(s + t)L + t \log M + o(1)$ in [10, Corollary 1] since $L > \log M$. Moreover, this new bound is asymptotically tight since there exist $(s, t, \bullet)_\mathbb{L}$ -correcting

codes of redundancy $(s + 2t)L$ [10, Construction 2 and Theorem 13].

Using the same argument, we have the following results for codes that can correct deletions.

Proposition 6: Let s and t be positive integers such that $s + t \leq M$. A code $\mathcal{S} \subseteq \mathcal{X}_M^{2,L}$ is an $(s, t, \bullet)_{\mathbb{D}}$ -correcting code if and only if the corresponding constant-weight code $\mathcal{C}(\mathcal{S})$ has minimum Hamming distance $2(s + t) + 2$.

Corollary 7: Let s and t be positive integers such that $s + t \leq M$. For any $(s, t, \bullet)_{\mathbb{D}}$ -correcting code \mathcal{S} , the code size satisfies

$$|\mathcal{S}| \leq \frac{\binom{2^L}{M-s-t}}{\binom{M}{M-s-t}}.$$

In particular, if both s and t are fixed, the redundancy of an $(s, t, \bullet)_{\mathbb{D}}$ -correcting code is at least

$$(s + t)L - \log((s + t)!) - o(1).$$

Note that the $(s, t, \bullet)_{\mathbb{D}}$ -correcting code in [10, Table 2] has redundancy $(s + t)L$. This redundancy almost meets the lower bound in Corollary 7, and is strictly less than the minimum redundancy $(s + 2t)L - \log((s + 2t)!) - o(1)$ required for correcting substitutions.

B. Explicit Code Constructions

For $(s, t, \bullet)_{\mathbb{T}}$ -correcting codes, three constructions can be found in [10]. In particular, [10, Construction 1] and [10, Construction 3] can produce codes with redundancy $\Theta(M)$ and $\Theta(M^c \log M)$ for some real constant $c > 0$, respectively, while [10, Construction 2] requires δL bits of redundancy, where

$$\delta = \begin{cases} s + 2t, & \text{if } \mathbb{T} \in \{\mathbb{S}, \mathbb{L}\}, \\ s + t, & \text{if } \mathbb{T} = \mathbb{D}. \end{cases} \quad (1)$$

Noting that $L = \beta^{-1} \log M$, the latter construction is much better than the former two. However, efficient encoding² for [10, Construction 2] is unknown.

In this section, we propose an explicit construction of $(s, t, \bullet)_{\mathbb{T}}$ -correcting codes with redundancy at most $\delta L + O(\log \log M)$, that can be encoded efficiently when δ is fixed. Our method modifies [10, Construction 1], where the code contains the codewords $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\} \subseteq \mathbb{F}_2^L$ such that

- 1) $\mathbf{x}_i = (I(i), \mathbf{u}_i)$ for $1 \leq i \leq M$, where $I(i)$ is the binary representation of $i - 1$;
- 2) if each \mathbf{u}_i is regarded as an element of $\mathbb{F}_{2^{L-\log M}}$, the sequence $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M)$ belongs to a given $[M, M - \delta, \delta + 1]$ MDS code over $\mathbb{F}_{2^{L-\log M}}$, where $\delta \triangleq s + 2t$.

In our construction, instead of using the binary representations $I(i)$ to index the sequences in the codeword S , we use sequences of length L' with $L' > \log M$ to index those sequences. Specifically, let $\log M < L' < L$, and let \mathcal{A} be the collection of all the subsets of $\mathbb{F}_2^{L'}$ of size M . Each set $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M\} \in \mathcal{A}$ is regarded as a set of

addresses.³ For each codeword S of the proposed DNA-storage code, we associate an address set $A \in \mathcal{A}$ to S and use the addresses \mathbf{a}_i 's to index the sequences in S . It is worth noting that, in our construction, different codewords may be associated with different address sets, while in [10, Construction 1] all the codewords use the same set of addresses, i.e., $\{I(i) \mid 1 \leq i \leq M\}$.

Besides \mathcal{A} , our construction also requires the following codes:

- A binary systematic $[2^{L'} + \delta(L' + 1), 2^{L'}, 2\delta + 1]$ code $\mathcal{C}_{\mathcal{A}}$ from Lemma 2. For each $A \in \mathcal{A}$, let $\text{enc}_{\mathcal{A}}(A)$ be the vector of $\mathbb{F}_2^{\delta(L'+1)}$ such that $(\mathbb{1}(A), \text{enc}_{\mathcal{A}}(A)) \in \mathcal{C}_{\mathcal{A}}$.
- A hash function $\text{Hash}_{\delta} : \{0, 1\}^{\delta(L'+1)} \rightarrow \{0, 1\}^h$ from Lemma 1, where $h = 4\delta \log(L') + o(\log L')$.
- An $[M, M - \delta, \delta + 1]$ MDS code \mathcal{B} over $\mathbb{F}_{2^{L-L'}}$. Such a code exists whenever $L - L' \geq \log M$.

Theorem 8: Let $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}, \mathbb{L}\}$. Given s, t, M and L , let δ be defined as in (1) and let L' be a positive integer such that $\log M < \min\{L', L - L'\}$. Suppose that $\delta(L' + 1) + h \leq M - \delta$, and assume \mathcal{A} , $\mathcal{C}_{\mathcal{A}}$, Hash_{δ} , and \mathcal{B} , are as above.

Denote by \mathcal{S} the collection of the sets $\{\mathbf{x}_i = (\mathbf{a}_i, \mathbf{u}_i) \mid 1 \leq i \leq M\} \subseteq \mathbb{F}_2^L$ that satisfy all of the following conditions:

- 1) $A \triangleq \{\mathbf{a}_1, \dots, \mathbf{a}_M\} \in \mathcal{A}$ (indexed lexicographically).
- 2)

$$\begin{aligned} & (\mathbf{u}_1[1], \mathbf{u}_2[1], \dots, \mathbf{u}_{\delta(L'+1)+h}[1]) \\ & = (\text{enc}_{\mathcal{A}}(A), \text{Hash}_{\delta}(\text{enc}_{\mathcal{A}}(A))). \end{aligned}$$

- 3) $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M) \in \mathcal{B}$, where \mathbf{u}_i is treated as an element of $\mathbb{F}_{2^{L-L'}}$.

Then the code \mathcal{S} is an $(s, t, \bullet)_{\mathbb{T}}$ -correcting code of size

$$|\mathcal{S}| = \binom{2^{L'}}{M} 2^{(L-L')(M-\delta)-\delta(L'+1)-h}.$$

Proof: We first check the size of \mathcal{S} , i.e., the number of possible choices of $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ that satisfy all the conditions above. The construction is depicted in Fig. 1. From 1), there are $\binom{2^{L'}}{M}$ choices of $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M\}$. Given A , according to 2), there are $2^{L-L'-1}$ choices of \mathbf{u}_i for each $1 \leq i \leq \delta(L' + 1) + h$, and $2^{L-L'}$ choices of \mathbf{u}_i for each $\delta(L' + 1) + h + 1 \leq i \leq M - \delta$. Finally, for $M - \delta + 1 \leq i \leq M$, according to 3) these \mathbf{u}_i 's are determined by $(\mathbf{u}_1, \dots, \mathbf{u}_{M-\delta})$ since the code \mathcal{B} has dimension $M - \delta$. Thus, the size of \mathcal{S} stated in the theorem is correct.

Now, we show that \mathcal{S} is an $(s, t, \bullet)_{\mathbb{T}}$ -correcting code by describing a decoding procedure. Suppose that the input of the channel is a set $S = \{\mathbf{x}_i = (\mathbf{a}_i, \mathbf{u}_i) \mid 1 \leq i \leq M\}$ and the output is a set $S' = \{\mathbf{x}'_i = (\mathbf{a}'_i, \mathbf{u}'_i) \mid 1 \leq i \leq M - s'\}$, where $0 \leq s' \leq s$ and the sequences in \mathcal{S} and \mathcal{S}' are enumerated in a descending lexicographic order. Our decoding has the following steps:

Step 1: Let $\mathbf{c}' \triangleq (\mathbf{u}'_1[1], \mathbf{u}'_2[1], \dots, \mathbf{u}'_{\delta L'+h-s}[1])$. Then \mathbf{c}' can be obtained from $\mathbf{c} \triangleq (\mathbf{u}_1[1], \mathbf{u}_2[1], \dots, \mathbf{u}_{\delta(L'+1)+h}[1])$ by deleting s' elements and inserting t' elements for some s' and t' . In the following, we give an upper bound on $s' + t'$. We treat the channel as having two stages. In the first stage, only

²By "efficient", we mean the complexity of the encoding is in $\text{poly}(M)$ time.

³Throughout this paper we keep enumerating the sequences in the address set A in a descending lexicographic order.

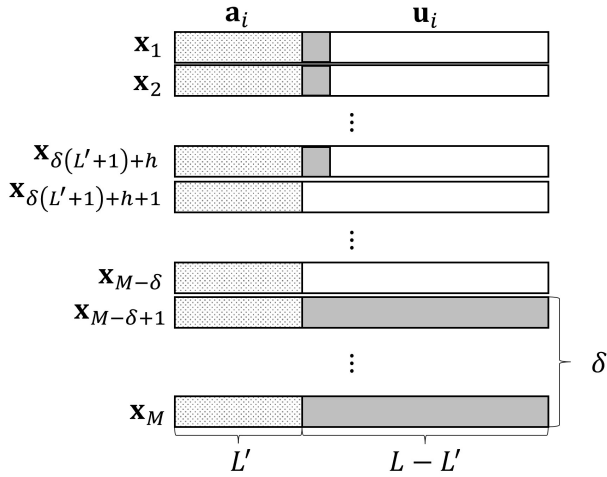


Fig. 1. The construction in Theorem 8. The dotted parts represent the addresses, the blank parts represent the information bits, and the gray parts represent the check bits of the codeword.

s sequences are lost but no surviving sequence is corrupted. Hence, in this stage there are s deletions in \mathbf{c} . In the second stage, at most t sequences are corrupted. Note that each erroneous sequence results in either one substitution or one deletion and one insertion. In total, we have that $s' + t' \leq s + 2t$. If $\mathbb{T} = \mathbb{S}$ or \mathbb{L} , then $s' + t' \leq s + 2t = \delta$. Due to the error-correcting capability of the deletion code in Lemma 1, we are able to recover \mathbf{c} from \mathbf{c}' ,⁴ and so also $\text{enc}_{\mathcal{A}}(A)$. If $\mathbb{T} = \mathbb{D}$, we further delete $\mathbf{u}'_{\ell}[1]$ from \mathbf{c}' if the length of \mathbf{x}'_{ℓ} is less than L and denote the resulting sequence as \mathbf{c}'' . Note that in this case each erroneous sequence only results in one deletion. Thus, \mathbf{c}'' can be obtained from \mathbf{c} by deleting at most $s + t = \delta$ elements. Hence, we can recover \mathbf{c} from \mathbf{c}'' , and so also $\text{enc}_{\mathcal{A}}(A)$.

Step 2: Let

$$A' \triangleq \{\mathbf{a}' \mid \mathbf{x}' = (\mathbf{a}', \mathbf{u}') \in S', |\mathbf{x}'| = L\}.$$

Here we ignore the multiplicity of the elements in A' , namely, A' is a simple set. Since at most s sequences of S are lost and t sequences are erroneous, the characteristic vector $\mathbb{1}(A')$ can be treated as an erroneous version of $\mathbb{1}(A)$ with at most $s + 2t$ substitution errors if $\mathbb{T} = \mathbb{S}$ or \mathbb{L} , or with at most $s + t$ substitutions if $\mathbb{T} = \mathbb{D}$. Thus, we can run the decoding algorithm of $\mathcal{C}_{\mathcal{A}}$ on the concatenation of $\mathbb{1}(A')$ and $\text{enc}_{\mathcal{A}}(A)$ to recover $\mathbb{1}(A)$, and so, the address set $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M\}$.

Step 3: For each $1 \leq i \leq M$, if there is a unique sequence $\mathbf{x}' = (\mathbf{a}', \mathbf{u}') \in S'$ with $|\mathbf{x}'| = L$ such that $\mathbf{a}' = \mathbf{a}_i$, let $\mathbf{u}'_i = \mathbf{u}'$; otherwise, let \mathbf{u}'_i be the empty string (representing an erasure). Consider the sequence $(\mathbf{u}'_1, \mathbf{u}'_2, \dots, \mathbf{u}'_M)$ over $\mathbb{F}_2^{L-L'}$. Using the same argument as that in the proof of [10, Construction 1], one can show that this sequence can be obtained from $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M)$ by at most s' erasures and t' substitutions with $s' + 2t' \leq \delta$ if $\mathbb{T} \in \{\mathbb{S}, \mathbb{L}\}$, or by at most $s + t$ erasures if $\mathbb{T} = \mathbb{D}$. So, we may run the decoding algorithm of \mathcal{B} on $(\mathbf{u}'_1, \mathbf{u}'_2, \dots, \mathbf{u}'_M)$ to recover $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M)$. ■

⁴We note that the decoding in Lemma 1 is designed to correct deletions, while we need to correct both deletions and insertions. However, since the number of codewords is $2^{\delta(L'+1)}$, if $L' = O(\log M)$, we still can decode in $\text{poly}(M)$ time.

We now turn to show that the redundancy of the code constructed in Theorem 8 is $\delta L + O(\log \log M)$.

Corollary 9: Let $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}, \mathbb{L}\}$. Given s, t, M and L , let δ be defined as in (1). Assume that $M \geq 2\delta \log M + 6\delta \log \log M$ and $L > 3 \log M$, then there is an $(s, t, \bullet)_{\mathbb{T}}$ -correcting code of $\mathcal{X}_M^{2,L}$ with redundancy at most

$$\delta L + 4\delta \log \log M + o(\log \log M).$$

Proof: Applying Theorem 8, we get an $(s, t, \bullet)_{\mathbb{T}}$ -correcting code \mathcal{S} with redundancy

$$\begin{aligned} & \log \binom{2^L}{M} - \log |\mathcal{S}| \\ &= \log \binom{2^L}{M} - \log \binom{2^{L'}}{M} \\ & \quad - (L - L')(M - \delta) + \delta(L' + 1) + h \\ & \leq M(L - L') + M \log \frac{2^{L'}}{2^{L'} - M} \\ & \quad - (L - L')(M - \delta) + \delta(L' + 1) + h \\ & \leq \delta(L - L') + \delta(L' + 1) + h + \log e \frac{M^2}{2^{L'} - M} \\ & = \delta L + \delta + h + \log e \frac{M^2}{2^{L'} - M}, \\ & = \delta L + \delta + 4\delta \log(L') + \log e \frac{M^2}{2^{L'} - M} + o(\log L'). \end{aligned}$$

Note that if $L' \leq c \log M$ for some constant $0 < c < 2$, then $\frac{M^2}{2^{L'} - M} = \Omega(M^{2-c})$. Thus, we require that $L' \geq 2 \log M$. On the other hand, $\log(L')$ is increasing with L' and $L > L' + \log M$. So we choose $L' = 2 \log M$ and then, the redundancy is $\delta L + 4\delta \log \log M + o(\log \log M)$. ■

Next, we analyze the complexity of the encoding and the decoding of the codes in Theorem 8. Assume that the message is

$$(\mathbf{a}, \mathbf{m}) \in \left[0, \binom{2^{L'}}{M} - 1\right] \times \mathbb{F}_2^{(L-L')(M-\delta) - \delta(L'+1) - h}.$$

Let us first examine the encoding process. First, we encode the integer a of $\left[0, \binom{2^{L'}}{M} - 1\right]$ to a subset $A = \{\mathbf{a}_1, \dots, \mathbf{a}_M\} \in \mathcal{A}$. This can be done by a greedy algorithm in $O(M2^{L'}L')$ time. We run the encoding of $\mathcal{C}_{\mathcal{A}}$ on $\mathbb{1}(A)$ to obtain $\text{enc}_{\mathcal{A}}(A)$. Since $\mathcal{C}_{\mathcal{A}}$ has dimension $2^{L'}$ and redundancy $\delta(L' + 1)$, the time complexity is $O(2^{L'}\delta(L' + 1))$. Then, we compute $\text{Hash}_{\delta}(\text{enc}_{\mathcal{A}}(A))$, which can be done in $O((\delta L')^{2\delta+1})$ time. We write $\text{enc}_{\mathcal{A}}(A)$, $\text{Hash}_{\delta}(\text{enc}_{\mathcal{A}}(A))$ and the word \mathbf{m} of $\mathbb{F}_2^{(L-L')(M-\delta) - \delta(L'+1) - h}$ onto \mathbf{u}_i , where $1 \leq i \leq M - \delta$. Finally, we use the encoding of the code \mathcal{B} to determine \mathbf{u}_i , where $M - \delta + 1 \leq i \leq M$. The time complexity of this step is $O(\delta(M - \delta)(L - L'))$. In summary, taking $L' = 2 \log M$, and since δ is a constant, the total time complexity of the encoding is $O(M^3 \log M)$ operations over \mathbb{F}_2 .

For the decoding, in Step 1 we first sort the sequences in the received codeword S' and then decode $\text{enc}_{\mathcal{A}}(A)$ from \mathbf{c}' . This may be done in $O(M^2 \log M)$ time. Steps 2 and 3 simply run the decoding of the codes $\mathcal{C}_{\mathcal{A}}$ and \mathcal{B} . Hence, the complexity of the whole decoding is $O(M^3)$.

C. Application to $(s, M - s, \varepsilon)_{\mathbb{T}}$ -Correcting Codes

In [10], Lenz et al. gave a concatenation method to construct $(s, M - s, \varepsilon)_{\mathbb{T}}$ -correcting codes. This construction uses an $(s, 0, 0)_{\mathbb{T}}$ -correcting code as the inner code. Lenz et al. suggested to use their Constructions 1, 2 or 3 to obtain the required $(s, 0, 0)_{\mathbb{T}}$ -correcting code. Now we can use the code from Theorem 8 as the inner code to construct the $(s, M - s, \varepsilon)_{\mathbb{T}}$ -correcting code and the whole construction is still explicit.

Lemma 10: [10, Construction 4 and Lemma 8] Let $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}\}$. Let $\mathcal{S}_o \subseteq \mathcal{X}_M^{2, L_o}$ be an $(s, 0, 0)_{\mathbb{T}}$ -correcting code and \mathcal{C}_i be a block-code of dimension L_o and length L that can correct ε errors of type \mathbb{T} . Let $\text{enc}_i(\cdot) : \{0, 1\}^{L_o} \rightarrow \{0, 1\}^L$ be an encoder of the code \mathcal{C}_i . Define

$$\mathcal{S} \triangleq \left\{ S \in \mathcal{X}_M^{2, L} \mid S = \bigcup_{\mathbf{x}_o \in \mathcal{S}_o} \{\text{enc}_i(\mathbf{x}_o)\}, S_o \in \mathcal{S}_o \right\}.$$

Then \mathcal{S} is an $(s, M - s, \varepsilon)_{\mathbb{T}}$ -correcting code.

Let $r_o \triangleq \log \binom{2^{L_o}}{M} - \log |\mathcal{S}_o|$ and $r_i \triangleq L - L_o$ be the redundancy of the outer code and the inner code, respectively. When $L_o \geq 2 \log M$, the redundancy of the code \mathcal{S} in the construction above can be bounded as follows:

$$\begin{aligned} & \log \binom{2^L}{M} - \log |\mathcal{S}| \\ &= \log \binom{2^L}{M} - \log \binom{2^{L_o}}{M} + \log \binom{2^{L_o}}{M} - \log |\mathcal{S}_o| \\ &\leq M(L - L_o) + M \log \frac{2^{L_o}}{2^{L_o} - M} + \log \binom{2^{L_o}}{M} - \log |\mathcal{S}_o| \\ &\leq M r_i + r_o + 2 \log e. \end{aligned}$$

Using the code from Lemmas 1 or 2 as the inner code and the code from Corollary 9 as the outer code, we obtain the following results.

Corollary 11: For any positive integers s, M, L_o with $L_o > 3 \log M$ and $M \geq 2s \log M + 5s \log \log M$, and a fixed positive integer ε , there is an $(s, M - s, \varepsilon)_{\mathbb{S}}$ -correcting code $\mathcal{S} \subseteq \mathcal{X}_M^{2, L}$ with $L = L_o + \varepsilon(\lceil \log L_o \rceil + 1)$ and redundancy at most

$$M\varepsilon(\lceil \log L_o \rceil + 1) + sL_o + 4s \log \log M + o(\log \log M).$$

Corollary 12: For any positive integers s, M, L_o with $L_o > 3 \log M$ and $M \geq 2s \log M + 5s \log \log M$, and a fixed positive integer ε , there is an $(s, M - s, \varepsilon)_{\mathbb{D}}$ -correcting code $\mathcal{S} \subseteq \mathcal{X}_M^{2, L}$ with $L = L_o + 4\varepsilon \log L_o + o(\log L_o)$, and redundancy at most

$$4M\varepsilon \log L_o + sL_o + o(M \log L_o).$$

IV. $(0, t, \varepsilon)_{\mathbb{T}}$ -CORRECTING CODES

In this section, we study channels that have no sequence loss, namely, $(0, t, \varepsilon)_{\mathbb{T}}$ -correcting codes with $\mathbb{T} \in \{\mathbb{S}, \mathbb{D}\}$. We improve the lower bound on the redundancy of optimal $(0, t, \varepsilon)_{\mathbb{D}}$ -correcting codes and propose new constructions for $(0, 1, \varepsilon)_{\mathbb{D}}$ -correcting codes and $(0, t, \varepsilon)_{\mathbb{S}}$ -correcting codes.

A. $(0, t, \varepsilon)_{\mathbb{D}}$ -Correcting Codes

Let $s = 0$ and t and ε be fixed. Lenz et al. [10] showed the following two lower bounds on the number of redundancy bits that are required to correct substitutions and deletions, respectively.

Lemma 13 ([10, Theorem 7 and Theorem 9]): For fixed positive integers t and ε , the redundancy of a $(0, t, \varepsilon)_{\mathbb{S}}$ -correcting code is at least

$$t \log M + t\varepsilon \log L + o(1),$$

while the redundancy of a $(0, t, \varepsilon)_{\mathbb{D}}$ -correcting code is at least

$$t\varepsilon \log L + o(1).$$

Since $M = 2^{\beta L}$, it follows that $t\varepsilon \log L = O(\log \log M)$. Thus Lemma 13 implies that in the $(0, t, \varepsilon)_{\mathbb{T}}$ -DNA storage channel, correcting deletions may require fewer redundancy bits than correcting substitutions. When $t = \varepsilon = 1$, Lenz et al. demonstrated this by constructing a class of $(0, 1, 1)_{\mathbb{D}}$ -correcting codes of redundancy $\log(L + 1)$. Their method utilized the fact that one can directly identify the unique erroneous sequence with deletions. We generalize their method to $\varepsilon > 1$ and obtain the following result.

Theorem 14: Let M, L , and ε , be positive integers. Let $\text{Hash}_\varepsilon : \{0, 1\}^L \rightarrow \{0, 1\}^{h_\varepsilon}$ be the hash function defined in Lemma 1, where $h_\varepsilon = 4\varepsilon \log L + o(\log L)$. For any $\mathbf{a} \in \{0, 1\}^{h_\varepsilon}$, define

$$\mathcal{S}_\mathbf{a} \triangleq \left\{ S \in \mathcal{X}_M^{2, L} \mid \sum_{\mathbf{x} \in S} \text{Hash}_\varepsilon(\mathbf{x}) = \mathbf{a} \right\}$$

Then $\mathcal{S}_\mathbf{a}$ is a $(0, 1, \varepsilon)_{\mathbb{D}}$ -correcting code. Furthermore, there is at least one choice of $\mathbf{a} \in \{0, 1\}^{h_\varepsilon}$ such that the code $\mathcal{S}_\mathbf{a}$ has redundancy at most

$$4\varepsilon \log L + o(\log L).$$

Proof: Suppose that the input of the channel is S and the output is S' . W.l.o.g., assume that the deletions occur in the sequence $\mathbf{x}_o \in S$ and result in a sequence $\mathbf{x}'_o \in S'$. We can identify the $M - 1$ error-free sequences from S' as they have length L and the erroneous sequence \mathbf{x}'_o has length less than L . So we have that

$$\text{Hash}_\varepsilon(\mathbf{x}_o) = \mathbf{a} - \sum_{\mathbf{x} \in S', |\mathbf{x}|=L} \text{Hash}_\varepsilon(\mathbf{x}).$$

We then recover the sequence \mathbf{x}_o from \mathbf{x}'_o by running the decoding algorithm mentioned in Lemma 1. ■

As a consequence, we observe that as long as $t = 1$, correcting deletions indeed requires fewer redundancy bits than correcting substitutions. When $t \geq 2$, however, the lower bound for deletions in Lemma 13 is not tight. We can improve it exponentially, from $\Omega(\log \log M)$ to $\lfloor t/2 \rfloor \log M$.

Theorem 15: Let M, L, t, ε be positive integers with t and ε fixed. Assume that $L > 3 \log M + \varepsilon$. Then the redundancy of a $(0, t, \varepsilon)_{\mathbb{D}}$ -correcting code is at least

$$\lfloor t/2 \rfloor \log M + \lfloor t/2 \rfloor \varepsilon - O(1).$$

Proof: For each $S \in \mathcal{X}_M^{2, L}$, we index the sequences $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$ in S such that they are in a descending

lexicographic order. Denote $L_\varepsilon \triangleq L - \varepsilon$. Let $S|_{L_\varepsilon}$ be the multiset projection of S onto the first L_ε bits, i.e., the multiset

$$S|_{L_\varepsilon} \triangleq \{\mathbf{x}_1[1, L_\varepsilon], \mathbf{x}_2[1, L_\varepsilon], \dots, \mathbf{x}_M[1, L_\varepsilon]\}.$$

Partition $\mathcal{X}_M^{2,L}$ into equivalence classes $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_m$ such that S and S' are in the same subset if and only if their multiset projections $S|_{L_\varepsilon}$ and $S'|_{L_\varepsilon}$ are the same. Each S contains M distinct sequences, so in each projection, each sequence of length L_ε occurs at most 2^ε times. Thus, the number of equivalence classes m is exactly the number of ways to throw M indistinguishable balls into 2^{L_ε} distinguishable urns, each of capacity limited to 2^ε balls. This number is known to be (e.g., see [2, Ex. 6, p. 360])

$$m = \sum_{j=0}^{2^{L_\varepsilon}} (-1)^j \binom{2^{L_\varepsilon}}{j} \binom{2^{L_\varepsilon} + M - j(2^\varepsilon + 1) - 1}{2^{L_\varepsilon}}.$$

This expression for m , however, is inconvenient to work with, so now we give an upper bound on m . W.l.o.g., we assume that for $1 \leq i \leq m_1$, where $m_1 \leq m$, the multiset projection in each \mathcal{D}_i contains M different sequences of length L_ε , and for $m_1 < i \leq m$, the multiset projection contains fewer than M distinct sequences. For $1 \leq i \leq m_1$, since the projection in each \mathcal{D}_i has M different sequences, \mathcal{D}_i has size exactly $(2^\varepsilon)^M$, and so

$$m_1 = \binom{2^{L_\varepsilon}}{M} \leq \frac{\binom{2^{L_\varepsilon}}{M}}{2^{\varepsilon M}}. \quad (2)$$

The number of equivalence classes with repetitions is

$$m - m_1 \leq \sum_{K=1}^{M-1} \binom{2^{L_\varepsilon}}{K} K^{M-K},$$

where in this expression, K counts the number of distinct sequences in the multiset, $\binom{2^{L_\varepsilon}}{K}$ gives the number of choices of these distinct sequences, and K^{M-K} counts how the remaining $M - K$ sequences as repetitions of the K distinct ones (we ignore the 2^ε upper limit on repetition). Since $L > 3 \log M + \varepsilon$, when $K \leq M - 2$, we have

$$\frac{\binom{2^{L_\varepsilon}}{K} K^{M-K}}{\binom{2^{L_\varepsilon}}{K+1} (K+1)^{M-K-1}} = \frac{(K+1)^2}{2^{L_\varepsilon} - K} \left(\frac{K}{K+1} \right)^{M-K} < 1.$$

It follows that $\binom{2^{L_\varepsilon}}{K} K^{M-K}$ is increasing in K . Hence,

$$m - m_1 = \sum_{K=1}^{M-1} \binom{2^{L_\varepsilon}}{K} K^{M-K} \leq \binom{2^{L_\varepsilon}}{M-1} M^2. \quad (3)$$

We show that the number in (2) is larger than that in (3):

$$\begin{aligned} & \frac{\binom{2^{L_\varepsilon}}{M}}{2^{\varepsilon M}} \bigg/ \left(\binom{2^{L_\varepsilon}}{M-1} M^2 \right) \\ &= \frac{(2^L - M + 1)(2^L - M + 2)(2^L - M + 3) \cdots 2^L}{M(2^{L_\varepsilon} - M + 2)(2^{L_\varepsilon} - M + 3) \cdots 2^{L_\varepsilon}} \\ & \quad \cdot \frac{1}{2^{\varepsilon M} M^2} \\ & \geq \frac{(2^{L_\varepsilon} - M + 1)(2^{L_\varepsilon} - M + 2)(2^{L_\varepsilon} - M + 3) \cdots 2^{L_\varepsilon}}{M^3 (2^{L_\varepsilon} - M + 2)(2^{L_\varepsilon} - M + 3) \cdots 2^{L_\varepsilon}} \\ &= \frac{2^{L_\varepsilon} - M + 1}{M^3} \geq 2^{L_\varepsilon - 1 - 3 \log M} \geq 1. \end{aligned}$$

Hence,

$$m \leq \frac{\binom{2^L}{M}}{2^{\varepsilon M - 1}}.$$

Now, let \mathcal{S} be a $(0, t, \varepsilon)_{\mathbb{D}}$ -correcting code. According to the pigeonhole principle, there is one \mathcal{D}_{i_0} , where $1 \leq i_0 \leq m$, such that $\mathcal{S} \cap \mathcal{D}_{i_0}$ has size at least $\frac{|\mathcal{S}|}{m}$. Denote $\mathcal{S}^* \triangleq \mathcal{S} \cap \mathcal{D}_{i_0}$. So

$$|\mathcal{S}^*| \geq \frac{|\mathcal{S}|}{m} \geq \frac{|\mathcal{S}|}{\binom{2^L}{M} / 2^{\varepsilon M - 1}}. \quad (4)$$

Let $\Sigma \triangleq \{0, 1\}^\varepsilon$ and

$\mathcal{C} \triangleq$

$$\left\{ (\mathbf{x}_1[L_\varepsilon + 1, L], \mathbf{x}_2[L_\varepsilon + 1, L], \dots, \mathbf{x}_M[L_\varepsilon + 1, L]) \in \Sigma^M \mid \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\} \in \mathcal{S}^* \right\}.$$

We point out that while $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\} \in \mathcal{S}^*$ is a set, at this point we use the lexicographic ordering to assign the indices, resulting in a single vector, $(\mathbf{x}_1[L_\varepsilon + 1, L], \mathbf{x}_2[L_\varepsilon + 1, L], \dots, \mathbf{x}_M[L_\varepsilon + 1, L]) \in \Sigma^M$.

We contend that $\mathcal{C} \subseteq \Sigma^M$ is a code of minimum Hamming distance at least $t + 1$; otherwise, if there are two codewords in \mathcal{C} that have a Hamming distance of at most t , then the two corresponding codewords in \mathcal{S}^* would be confusable in the $(0, t, \varepsilon)_{\mathbb{D}}$ -DNA storage channel by deleting the length- ε suffixes corresponding to the positions in which the codewords in \mathcal{C} differ. Hence, by using the Hamming bound on $|\mathcal{C}|$, which is the same as $|\mathcal{S}^*|$, we have that

$$|\mathcal{S}^*| \leq \frac{2^{\varepsilon M}}{\sum_{i=0}^{\lfloor t/2 \rfloor} \binom{M}{i} (2^\varepsilon - 1)^i}. \quad (5)$$

Combining (4) and (5), we have that

$$|\mathcal{S}| \leq \frac{2 \binom{2^L}{M}}{\sum_{i=0}^{\lfloor t/2 \rfloor} \binom{M}{i} (2^\varepsilon - 1)^i}.$$

Hence

$$\begin{aligned} \log \binom{2^L}{M} - \log |\mathcal{S}| & \geq \log \left(\sum_{i=0}^{\lfloor t/2 \rfloor} \binom{M}{i} (2^\varepsilon - 1)^i \right) - 1 \\ &= \lfloor t/2 \rfloor \log M + \lfloor t/2 \rfloor \varepsilon - O(1). \end{aligned}$$

Remark: When $t \geq 2$, it is still unclear whether there are $(0, t, \varepsilon)_{\mathbb{D}}$ -correcting codes of redundancy less than the lower bound $t \log M + t \varepsilon \log L + o(1)$ in Lemma 13 for substitutions. The Gilbert-Varshamov bound shows that the redundancy of optimal $(0, t, \varepsilon)_{\mathbb{D}}$ -correcting codes is at most $t \log M + 2t \varepsilon \log(L/2)$, see [10, Thm. 4]. This upper bound is nearly twice the improved lower bound for deletions in Theorem 15, but is still a bit larger than the lower bound for substitutions. ■

B. $(0, t, \varepsilon)_{\mathbb{S}}$ -Correcting Codes

Next, we consider the problem of finding explicit constructions for $(0, t, \varepsilon)_{\mathbb{S}}$ -correcting codes. A related problem is studied in [9]. The input of that channel is a set S of M indexed sequences of \mathbb{F}_2^L , i.e., $S = \{(I(i), \mathbf{u}_i) \mid 1 \leq i \leq M\} \subseteq \mathbb{F}_2^L$, and at the decoder no sequences of S are lost, and at most t

sequences are erroneous where each $I(i)$ suffers at most ε_1 substitution errors and each \mathbf{u}_i at most ε_2 substitution errors. The construction proposed in [9] requires $M \log e + 4t \log M + 2t\varepsilon_2 \log L$ bits of redundancy.⁵

Our construction involves the following codes:

- A code

$$\mathcal{A} \triangleq \{ \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M\} \subseteq \mathbb{F}_2^{L'} \mid \mathbf{a}_1 = \mathbf{1}, d_H(\mathbf{a}_i, \mathbf{a}_j) \geq 2\varepsilon + 1 \text{ for all } i \neq j \}, \quad (6)$$

where $\log M < L' < L$. The idea behind this code comes from [9] and the cardinality analysis may be found in [13]. [16] shows that such a code can be constructed using an algorithm which is similar to the Gilbert-Varshamov bound so that

$$|\mathcal{A}| \geq \frac{\prod_{i=2}^M (2^{L'} - (i-1)Q)}{(M-1)!},$$

where $Q = \sum_{i=0}^{2\varepsilon} \binom{L'}{i}$ is the size of a Hamming ball of radius 2ε in $\mathbb{F}_2^{L'}$. [15] proposes an efficient encoding algorithm in $\text{poly}(M, L, \varepsilon)$ time.

- A binary $[2^{L'} + 2t(L' + 1), 2^{L'}, 4t + 1]$ code \mathcal{C}_A from Lemma 2. For each $A \in \mathcal{A}$, let $\text{enc}_A(A)$ be the vector of $\mathbb{F}_2^{2t(L'+1)}$ such that $(\mathbb{1}(A), \text{enc}_A(A)) \in \mathcal{C}_A$.
- A hash function $\text{Hash}_{2t} : \{0, 1\}^{2t(L'+1)} \rightarrow \{0, 1\}^h$ from in Lemma 1, where $h = 8t \log(L') + o(\log L')$.
- A binary $[L - L', L - L' - r, 2\varepsilon + 1]$ code \mathcal{C}_1 with $r \triangleq \varepsilon \lceil \log(L - L') \rceil$. To obtain such a code, we may shorten a binary $[n, n - r, 2\varepsilon + 1]$ BCH code with $n = 2^{\lceil \log(L - L') \rceil} - 1$.
- An $[M, M - \tilde{r}, 2t + 1]$ code \mathcal{C}_2 over \mathbb{F}_{2^r} . Let $q = 2^r$ and $m = \lceil \log(M + 1)/r \rceil$. Then this code can be obtained by shortening a $[q^m - 1, q^m - 1 - \tilde{r}, 2t + 1]$ BCH code over \mathbb{F}_q . Note that

$$\tilde{r} < 2tm = 2t \lceil \log(M + 1)/r \rceil.$$

Theorem 16: Given t, M and L , let L' be a positive integer such that $\log M < \min\{L', L - L'\}$. Let \mathcal{A} , $\text{enc}_A(A)$, Hash_{2t} , \mathcal{C}_1 and \mathcal{C}_2 be defined as above. Let H be the parity check matrix of \mathcal{C}_1 . Suppose that $M \geq 2t(L' + 1) + h$ where $h = 8t \log(L') + o(\log L')$.

Denote by \mathcal{S} the collection of the sets $\{\mathbf{x}_1 = (\mathbf{a}_i, \mathbf{u}_i) \mid 1 \leq i \leq M\} \subseteq \mathbb{F}_2^L$ that satisfy all of the following:

- 1) $A \triangleq \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M\} \in \mathcal{A}$.
- 2)

$$(\mathbf{u}_1[1], \mathbf{u}_2[1], \dots, \mathbf{u}_{2t(L'+1)+h}[1]) = (\text{enc}_A(A), \text{Hash}_{2t}(\text{enc}_A(A))).$$

- 3) $(\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_M) \in \mathcal{C}_2$, where $\mathbf{s}_i \triangleq \mathbf{u}_i H^T$ is regarded as an element of \mathbb{F}_{2^r} .⁶

Then the code \mathcal{S} is a $(0, t, \varepsilon)_{\mathcal{S}}$ -correcting code of size

$$|\mathcal{S}| = |\mathcal{A}| 2^{M(L-L') - 2t(L'+1) - h - r\tilde{r}}.$$

⁵The redundancy in [9], which is defined to be $M(L - \log M) - \log|\mathcal{S}|$, is different from the one defined in this paper and [10].

⁶This kind of coding scheme is known as a tensor product code [19]. See also [9] and the reference therein.

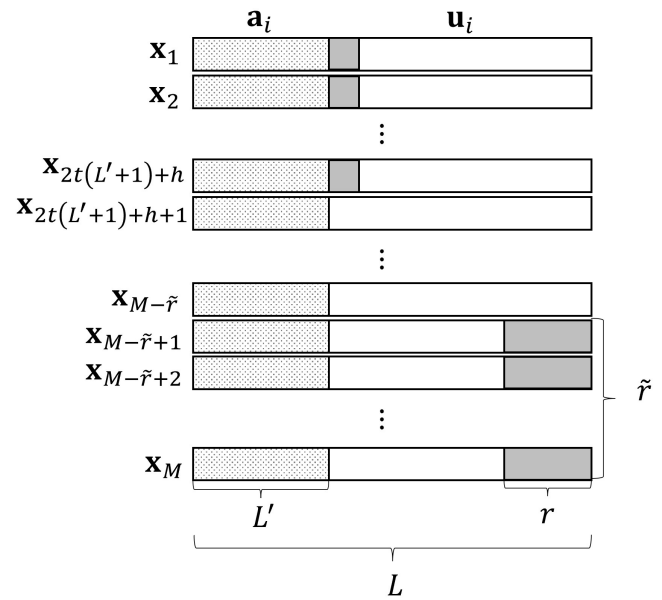


Fig. 2. The construction in Theorem 16. The dotted parts represent the addresses, the blank parts represent the information bits and the gray parts represent the check bits of the codeword.

Proof: We first check the size of \mathcal{S} , i.e., the number of possible choices of $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ satisfying all the conditions above. The construction is depicted in Fig. 2. For 1), there are $|\mathcal{A}|$ choices of $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M\}$. Given A , according to 2), there are $2^{L-L'-1}$ choices of \mathbf{u}_i for each $1 \leq i \leq 2t(L' + 1) + h$, and $2^{L-L'}$ choices of \mathbf{u}_i for each $2t(L' + 1) + h + 1 \leq i \leq M - \tilde{r}$. According to 3), the sequences \mathbf{u}_i , where $1 \leq i \leq M - \tilde{r}$, can determine \mathbf{s}_i for all $1 \leq i \leq M$, as \mathcal{C}_2 has dimension $M - \tilde{r}$. Now, for each $M - \tilde{r} + 1 \leq i \leq M$, given \mathbf{s}_i , there are $2^{L-L'-r}$ choices of \mathbf{u}_i , since \mathcal{C}_1 is a code of dimension $L - L' - r$. Thus, the size of \mathcal{S} is as stated.

Now, we show that \mathcal{S} is a $(0, t, \varepsilon)_{\mathcal{S}}$ -correcting code by providing a decoding procedure. Suppose that the input of the channel is a set $S = \{\mathbf{x}_i = (\mathbf{a}_i, \mathbf{u}_i) \mid 1 \leq i \leq M\}$ and the output is a set $S' = \{\mathbf{x}'_i = (\mathbf{a}'_i, \mathbf{u}'_i) \mid 1 \leq i \leq M\}$, where the sequences in \mathcal{S} and S' are enumerated in a descending lexicographic order. Our decoding has the following steps.

- 1) Let $\mathbf{c}' \triangleq (\mathbf{u}'_1[1], \mathbf{u}'_2[1], \dots, \mathbf{u}'_{2t(L'+1)+h}[1])$. Then \mathbf{c}' can be obtained from $\mathbf{c} = (\mathbf{u}_1[1], \mathbf{u}_2[1], \dots, \mathbf{u}_{2t(L'+1)+h}[1])$ by deleting t' elements and inserting t' elements with $t' \leq t$. Due to the error-correcting capability of the deletion code in Lemma 1, we are able to recover \mathbf{c} from \mathbf{c}' , and so $\text{enc}_A(A)$.
- 2) Denote $A' \triangleq \{\mathbf{a}' \mid (\mathbf{a}', \mathbf{u}') \in S'\}$. Noting that there are at most t sequences of S' suffering errors, the distance between $\mathbb{1}(A)$ and $\mathbb{1}(A')$ is at most $2t$. Thus, we may run the decoding algorithm of \mathcal{C}_A on the concatenation of $\mathbb{1}(A')$ and $\text{enc}_A(A)$ to recover $\mathbb{1}(A)$, and so the set A . Denote the sequences of A as $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M$, in a descending lexicographic order. Since A is a code of minimum distance $2\varepsilon + 1$, for each $1 \leq i \leq M$, there is a unique sequence of S' such that its length- L' prefix is of distance at most ε from \mathbf{a}_i . Denote this sequence as $\mathbf{x}'_i = (\mathbf{a}'_i, \mathbf{u}'_i)$.

- 3) Compute the syndromes $\mathbf{s}'_i = \mathbf{u}'_i H^T$ for $1 \leq i \leq M$. Since there are at most t sequences of S' are erroneous, we can run the decoding algorithm of \mathcal{C}_2 on $(\mathbf{s}'_1, \mathbf{s}'_2, \dots, \mathbf{s}'_M)$ to recover $(\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_M)$.
- 4) For each $1 \leq i \leq M$, choose an arbitrary solution \mathbf{y}_i to $\mathbf{y}_i H^T = \mathbf{s}_i$. Then $(\mathbf{u}_i - \mathbf{y}_i) H^T = \mathbf{u}_i H^T - \mathbf{y}_i H^T = \mathbf{0}$, and so $\mathbf{u}_i - \mathbf{y}_i$ is a codeword of \mathcal{B} . Run the decoding algorithm of \mathcal{C}_1 on $\mathbf{u}'_i - \mathbf{y}_i$ to recover $\mathbf{u}_i - \mathbf{y}_i$, and so $\mathbf{x}_i = (\mathbf{a}_i, \mathbf{u}_i)$. ■

Corollary 17: If $L \geq 4 \log M + 4\varepsilon^2 + 1$, t and ε are fixed positive integers, and M is sufficiently large, then there is a $(0, t, \varepsilon)_{\mathbb{S}}$ -correcting code with redundancy at most

$$(8t + 2) \log M + 2t\varepsilon \lceil \log L \rceil + 8t \log \log M + o(\log \log M).$$

Proof: Let $L' = 3 \log M + 4\varepsilon^2 + 1$. Then $\log M < \min\{L', L - L'\}$ and $M \geq 2t(L' + 1) + h$, and Theorem 16 shows that there is a $(0, t, \varepsilon)_{\mathbb{S}}$ -correcting code \mathcal{S} of redundancy

$$\begin{aligned} & \log \binom{2^L}{M} - \log |\mathcal{S}| \\ & \leq \log \binom{2^L}{M} - \log \frac{\prod_{i=2}^M [2^{L'} - (i-1)Q]}{(M-1)!} - M(L-L') \\ & \quad + 2t(L'+1) + r\tilde{r} + h \\ & \leq ML' - \log M - (M-1) \log (2^{L'} - MQ) \\ & \quad + 2t(L'+1) + h + r\tilde{r} \\ & = L' - \log M + (M-1) \log \left(1 + \frac{MQ}{2^{L'} - MQ} \right) \\ & \quad + 2t(L'+1) + h + r\tilde{r} \\ & \leq L' - \log M + \frac{(M-1)MQ}{2^{L'} - MQ} \log e + 2t(L'+1) + h + r\tilde{r} \\ & \stackrel{(a)}{\leq} L' - \log M + \log e + 2t(L'+1) + h + r\tilde{r} \\ & \leq (8t+2) \log M + 2t\varepsilon \lceil \log L \rceil + 8t \log \log M \\ & \quad + o(\log \log M), \end{aligned}$$

where the inequality (a) holds as $Q = \sum_{i=0}^{2\varepsilon} \binom{L'}{i} < (L')^{2\varepsilon}$, and $M^2(L')^{2\varepsilon} \leq 2^{L'}$ when $L' = 3 \log M + 4\varepsilon^2 + 1$, see [16, Appendix E]. ■

V. CODES CORRECTING LIMITED-MAGNITUDE ERRORS

In this section, we study the limited-magnitude errors, motivated by DNA-storage channels that involve such errors [5]. Unlike substitution errors studied in previous sections, a limited-magnitude error replaces an integer entry $x \in \mathbb{Z}$ by x' such that $x - k_- \leq x' \leq x + k_+$. Thus, we no longer use the binary alphabet, instead, using \mathbb{Z}_q .

First, we derive the following Gilbert-Varshamov bound.

Lemma 18: There is an $(s, t, \varepsilon, k_+, k_-)_{\text{LM}}$ -correcting code $\mathcal{S} \subseteq \mathcal{X}_M^{q,L}$ of size at least

$$\binom{q^L}{M} / \left[\binom{M}{s} \binom{M-s}{t} \binom{M-s+t-1}{t} \binom{q^L}{s} V^{2t} \right],$$

where

$$V = \sum_{i=0}^{\varepsilon} \binom{L}{i} (k_+ + k_-)^i.$$

Proof: Given a set $S \in \mathcal{X}_M^{q,L}$, let $B^{\text{LM}}(S)$ be the set of all possible received S' with S being the input of the $(s, t, \varepsilon, k_+, k_-)_{\text{LM}}$ -DNA storage channel. We shall give an upper bound on the number of sets $\tilde{S} \in \mathcal{X}_M^{q,L}$ such that $B^{\text{LM}}(S) \cap B^{\text{LM}}(\tilde{S}) \neq \emptyset$.

Note that if two balls $B^{\text{LM}}(S)$ and $B^{\text{LM}}(\tilde{S})$ intersect, necessarily in the intersection there is a set S' of size at most $M-s$. The number of $S' \in B^{\text{LM}}(S)$ with $|S'| \leq M-s$ is no more than $\binom{M}{s} \binom{M-s}{t} V^t$, where $V = \sum_{i=0}^{\varepsilon} \binom{L}{i} (k_+ + k_-)^i$. Now, fix a set $S' \in B^{\text{LM}}(S)$, we count the number of $\tilde{S} \in \mathcal{X}_M^{q,L}$ such that $S' \in B^{\text{LM}}(\tilde{S})$. First, there are at most $\binom{q^L}{s}$ choices for the lost sequences. Next, noting that each sequence of S' may come from different sequences of \tilde{S} with errors, the number of the choices for the erroneous sequences of \tilde{S} is at most $\binom{M-s+t-1}{t} V^t$. Hence, the number of \tilde{S} such that $B^{\text{LM}}(S) \cap B^{\text{LM}}(\tilde{S}) \neq \emptyset$ is at most

$$\binom{M}{s} \binom{M-s}{t} \binom{M-s+t-1}{t} \binom{q^L}{s} V^{2t}.$$

The conclusion follows from this estimation and a greedy argument. ■

Corollary 19: When s and t are fixed and $\varepsilon = L$, there is a q -ary $(s, t, \bullet, k_+, k_-)_{\text{LM}}$ -correcting code with redundancy at most

$$(s+2t) \log_q M + 2tL \log_q (k_+ + k_- + 1) + sL + O(1);$$

when s and ε are fixed, and $t = M - s$, there is a q -ary $(s, M-s, \varepsilon, k_+, k_-)_{\text{LM}}$ -correcting code with redundancy at most

$$2M(\varepsilon \log_q L + \varepsilon \log_q (k_+ + k_-) + \log_q 2 - \log_q \varepsilon!) + o(M);$$

when s, t, ε are fixed, there is a q -ary $(s, t, \varepsilon, k_+, k_-)_{\text{LM}}$ -correcting code of redundancy at most

$$(s+2t) \log_q M + sL + 2t\varepsilon \log_q L + 2t\varepsilon \log_q (k_+ + k_-) + O(1).$$

For lower bounds on the redundancy, we follow the approach in [10] and derive the following sphere-packing bounds. The proofs are the same as that of [10, Theorem 7 and Theorem 8], and we omit here.

Lemma 20: For fixed s, t, ε and fixed $0 < \beta < 1$, any $(s, t, \varepsilon, k_+, k_-)_{\text{LM}}$ -correcting code $\mathcal{S} \subseteq \mathcal{X}_M^{q,L}$ with $M = q^{\beta L}$ has redundancy at least

$$\begin{aligned} & sL + t \log_q M + t\varepsilon(\log_q L + \log_q (k_+ + k_-)) \\ & \quad - \log_q (s!t!(\varepsilon!)^t) + o(1). \end{aligned}$$

Lemma 21: For fixed s, ε and fixed $0 < \beta < 1$, any $(s, M-s, \varepsilon, k_+, k_-)_{\text{LM}}$ -correcting code $\mathcal{S} \subseteq \mathcal{X}_M^{q,L}$ with $M = q^{\beta L}$ has redundancy at least

$$M\varepsilon(\log_q L + \log_q ((k_+ + k_-))) + O(M).$$

We can make use of $(0, t, \varepsilon)_{\mathbb{S}}$ -codes which we constructed earlier as a basis for constructing codes for the limited-magnitude error scenario.

Theorem 22: Let $\mathcal{C} \subseteq \mathcal{X}_M^{p,L}$ be a $(0, t, \varepsilon)_{\mathbb{S}}$ -correcting code with $p \geq k_+ + k_- + 1$. Suppose that for each $C \in \mathcal{C}$, the minimum Hamming distance of C is at least $2\varepsilon + 1$. Define

$$\mathcal{S} \triangleq \{ \{\mathbf{x}_1, \dots, \mathbf{x}_M\} \subseteq \mathbb{Z}_q^L \mid \{\mathbf{x}_i \pmod{p} \mid 1 \leq i \leq M\} \in \mathcal{C} \}.$$

Then \mathcal{S} is a $(0, t, \varepsilon, k_+, k_-)_{\text{LM}}$ -correcting code over \mathbb{Z}_q of size $|\mathcal{S}| \geq \lfloor q/p \rfloor^{ML} |\mathcal{C}|$.

Proof: We prove the theorem by describing a decoding procedure. Let $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\} \in \mathcal{S}$ be the input of the channel, and S' be the output. Since each codeword $C \in \mathcal{C}$ has minimum Hamming distance $2\varepsilon + 1$, we can see that S' comprises M distinct sequences. W.l.o.g., we assume that $S' = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\}$ and there is a permutation π (to be determined later) such that for all $1 \leq i \leq M$, \mathbf{x}_i yields $\mathbf{y}_{\pi(i)}$ after passing through the channel.

Let $\chi_i = \mathbf{x}_i \pmod{p}$ and $\psi_i = \mathbf{y}_i \pmod{p}$. Then $\psi_{\pi(i)} - \chi_i = \mathbf{y}_{\pi(i)} - \mathbf{x}_i \pmod{p}$. That implies that $\psi_{\pi(i)}$ is an erroneous version of χ_i with at most ε positions being corrupted by substitution errors. Thus, we may run the decoding algorithm of the $(0, t, \varepsilon)_{\mathbb{S}}$ -correcting code \mathcal{C} on $\{\psi_1, \psi_2, \dots, \psi_M\}$ to recover the set $\{\chi_1, \chi_2, \dots, \chi_M\}$.

Now, for each χ_i , we have $d_H(\chi_i, \psi_{\pi(i)}) \leq \varepsilon$. We claim that $d_H(\chi_i, \psi_j) > \varepsilon$ for all $j \neq \pi(i)$. Otherwise,

$$d_H(\chi_i, \chi_{\pi^{-1}(j)}) \leq d_H(\chi_i, \psi_j) + d_H(\psi_j, \chi_{\pi^{-1}(j)}) \leq 2\varepsilon,$$

which contradicts the assumption that the minimum Hamming distance of C is at least $2\varepsilon + 1$. Therefore, the permutation π can be determined by computing the Hamming distance between χ_i and ψ_j for all $1 \leq i, j \leq M$. Denote $\epsilon_i = \psi_{\pi(i)} - \chi_i \pmod{p}$ and let $\mathbf{e}_i = (e_1^{(i)}, e_2^{(i)}, \dots, e_L^{(i)})$ where

$$e_\ell^{(i)} \triangleq \begin{cases} \epsilon_\ell^{(i)}, & \text{if } 0 \leq \epsilon_\ell^{(i)} \leq k_+; \\ \epsilon_\ell^{(i)} - p, & \text{otherwise.} \end{cases}$$

Then \mathbf{x}_i can be decoded as $\mathbf{x}_i = \mathbf{y}_{\pi(i)} - \mathbf{e}_i$. ■

Let

$$r_p(\mathcal{C}) \triangleq \log_p \binom{p^L}{M} - \log_p |\mathcal{C}|.$$

Then we have that

$$\log_q |\mathcal{C}| = \log_q p \log_p |\mathcal{C}| = \log_q \binom{p^L}{M} - r_p(\mathcal{C}) \log_q p.$$

If $p \mid q$ and $\log_p M < L/2$, then the redundancy of the code \mathcal{S} is

$$\begin{aligned} & \log_q \binom{q^L}{M} - \log_q |\mathcal{S}| \\ &= \log_q \binom{q^L}{M} - ML \log_q \left(\frac{q}{p} \right) - \log_q |\mathcal{C}| \\ &= \log_q \binom{q^L}{M} - \log_q \binom{p^L}{M} - ML \log_q \left(\frac{q}{p} \right) + r_p(\mathcal{C}) \log_q p \\ &\leq M \log_q \frac{p^L}{p^L - M} + r_p(\mathcal{C}) \log_q p \\ &\leq \frac{M^2}{p^L - M} \log_q e + r_p(\mathcal{C}) \log_q p \\ &= r_p(\mathcal{C}) \log_q p + o(1). \end{aligned}$$

We note that the code in Corollary 17 satisfies the condition in Theorem 22, i.e., each codeword has minimum Hamming distance $2\varepsilon + 1$, thus we may use it as the input code and get the following result.

Corollary 23: Let $q > 0$ be an even integer. If $L \geq (2t + 1)(3 \log M + 4\varepsilon^2 + 2) + \varepsilon \lceil \log L \rceil - 1$, and t and ε

are fixed, then there is a q -ary $(0, t, \varepsilon, 1, 0)_{\text{LM}}$ -correcting code with redundancy at most

$$(8t + 2) \log_q M + (2t + 1) \varepsilon \log_q L + O(1).$$

We note that this redundancy is larger than the Gilbert-Varshamov bound $2t \log_q M + 2t\varepsilon \log_q L + O(1)$.

Next, we modify Lemma 10 to construct an $(s, M - s, \varepsilon, k_+, k_-)_{\text{LM}}$ -correcting code.

Lemma 24: Let $\mathcal{S}_o \subseteq \mathcal{X}_M^{2, L_o}$ be an $(s, 0, 0)_{\text{T}}$ -correcting code and let \mathcal{C}_i be a q -ary block-code of size $2_{L_o}^L$ and length L that can correct ε (k_+, k_-) -limited-magnitude errors. Let $\text{enc}_i(\cdot) : \{0, 1\}^{L_o} \rightarrow \mathbb{Z}_q^L$ be an encoder of the inner code \mathcal{C}_i . Define

$$\mathcal{S} \triangleq \left\{ \bigcup_{\mathbf{x}_o \in \mathcal{S}_o} \{\text{enc}_i(\mathbf{x}_o)\} \in \mathcal{X}_M^{q, L} \mid \mathcal{S}_o \in \mathcal{S}_o \right\}.$$

Then \mathcal{S} is an $(s, M - s, \varepsilon, k_+, k_-)_{\text{LM}}$ -correcting code.

Proof: We first use the inner code to correct all the limited-magnitude errors, and then use the outer code to recover all the lost sequences. ■

Let $L_o > 3 \log M$. We may use the code in Corollary 9 as the outer code with redundancy $r_o = sL + O(\log \log M)$. As for the inner code, let p be the smallest prime number such that $p > k_+ + k_-$, and L be the smallest integer such that $\frac{q^L}{p^{L-K}} \geq 2^{L_o}$, where $L - K = \lceil 2\varepsilon(1 - 1/p) \log_p L \rceil$. We take the code in [18, Theorem 5 and Corollary 8] of size $\frac{q^L}{p^{L-K}}$ as the inner code. The redundancy of the resulting code is

$$\begin{aligned} & \log_q \binom{q^L}{M} - \log_q |\mathcal{S}_o| \\ &\leq \log_q \binom{q^L}{M} - \log_q \binom{2^{L_o}}{M} + \log_q \binom{2^{L_o}}{M} - \log_q |\mathcal{S}_o| \\ &\leq M \log_q \frac{q^L}{2^{L_o}} + M \log_q \frac{2^{L_o}}{2^{L_o} - M} + \frac{r_o}{\log q} \\ &\leq M \lceil 2\varepsilon(1 - 1/p) \log_p L \rceil \log_q p + o(M). \end{aligned}$$

Since $2\varepsilon(1 - 1/p) < 2\varepsilon$, this redundancy is usually better than the Gilbert-Varshamov bound in Corollary 19. When $(k_+, k_-) = (1, 0)$ and $p = 2$, it almost meets the sphere-packing bound $M\varepsilon \log_q L + o(M)$.

Finally, we would like to discuss the case where $\varepsilon = L$. We note that by using q -ary deletion-correcting codes, one can easily generalize Theorem 8 to obtain q -ary $(s, t, \bullet)_{\mathbb{S}}$ -correcting codes of redundancy

$$(s + 2t)L + O(\log_q \log_q M).$$

This code can be used as an $(s, t, \bullet, k_+, k_-)_{\text{LM}}$ -correcting code. In contrast, Corollary 19 shows the existence of such a code of redundancy no more than

$$sL + 2tL \log_q(k_+ + k_- + 1) + (s + 2t) \log_q M + O(1).$$

Since $\log_q(k_+ + k_- + 1)$ is less than one, in some cases this Gilbert-Varshamov bound is less than $(s + 2t)L$. It is therefore an interesting question to find explicit constructions of codes with redundancy less than $(s + 2t)L$. Besides, establishing a good lower bound on the redundancy of such codes is also an open problem.

We close this section with a description of the quaternary case. Unlike the binary case, a quaternary sequence produced by n rounds of synthesis process is represented by a sequence

$$\mathbf{d} \circ \mathbf{r} \triangleq ((d_1, r_1), (d_2, r_2), \dots, (d_n, r_n)) \in (\mathbb{Z}_3 \times \mathbb{Z}_q)^n,$$

where the sequence $\mathbf{d} = (d_1, d_2, \dots, d_n)$ represents the difference between the letters appended in the $(i-1)$ th round and i th round, or the difference between the letters in the initiator and in the first round (e.g., ref [8]), and the sequence $\mathbf{r} = (r_1, r_2, \dots, r_n)$ still represents the run-lengths.

Our codeword is a subset $S = \{\mathbf{d}_1 \circ \mathbf{r}_1, \mathbf{d}_2 \circ \mathbf{r}_2, \dots, \mathbf{d}_M \circ \mathbf{r}_M\}$ of $(\mathbb{Z}_3 \times \mathbb{Z}_q)^L$. The $(s, t, \varepsilon, k_+, k_-)_{\text{LM}}$ -DNA storage channel outputs a subset S' of S , with at most s sequences lost and at most t sequences corrupted. In each erroneous sequence $\mathbf{d}_i \circ \mathbf{r}_i$, the sequence \mathbf{d}_i is preserved and at most ε elements of \mathbf{r}_i are corrupted by the (k_+, k_-) -limited-magnitude errors.

In the following we briefly describe how to generalize the constructions in Theorem 22 and Lemma 24 to yield codes of $(\mathbb{Z}_3 \times \mathbb{Z}_q)^L$ with the same error-correcting capability. We do the encoding such that $(\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_M)$ is a codeword of an s -erasure-correcting code over \mathbb{F}_{3^L} , and $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_M\}$ is a codeword of the codes over sets in Theorem 22 or Lemma 24. Note that the constructions ensure that $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_M\}$ is a block-code correcting ε (k_+, k_-) -limited-magnitude errors. Assume that $S' = \{\mathbf{d}'_1 \circ \mathbf{r}'_1, \mathbf{d}'_2 \circ \mathbf{r}'_2, \dots, \mathbf{d}'_{M-s} \circ \mathbf{r}'_{M-s}\}$. We can first recover the set $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_M\}$ from $\{\mathbf{r}'_1, \mathbf{r}'_2, \dots, \mathbf{r}'_{M-s}\}$ by using the decoding schemes in the proofs of Theorem 22 or Lemma 24. Since $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_M\}$ is an (ε, k_+, k_-) -error-correcting code, by comparing the sequences \mathbf{r}'_i with the sequences \mathbf{r}_i , we can determine the ordering of the received sequences $\mathbf{d}'_i \circ \mathbf{r}'_i$. In this way we actually determine the ordering of the $M-s$ surviving sequences \mathbf{d}_i 's. Thus we can use the s -erasure-correcting code to recover the lost s sequences.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and the anonymous reviewers for their insightful comments and suggestions.

REFERENCES

- [1] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for asymmetric limited-magnitude errors with applications to multilevel flash memories," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1582–1595, Apr. 2010.
- [2] C. A. Charalambides, *Enumerative Combinatorics*. Boca Raton, FL, USA: CRC Press, 2002.
- [3] Y. M. Chee, H. M. Kiah, and H. Wei, "Efficient and explicit balanced primer codes," *IEEE Trans. Inf. Theory*, vol. 66, no. 9, pp. 5344–5357, Sep. 2020.
- [4] R. Graham and N. Sloane, "Lower bounds for constant weight codes," *IEEE Trans. Inf. Theory*, vol. IT-26, no. 1, pp. 37–43, Jan. 1980.
- [5] S. Jain, F. Farnoud, M. Schwartz, and J. Bruck, "Coding for optimized writing rate in DNA storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 711–716.
- [6] M. Kovacevic and V. Y. F. Tan, "Codes in the space of multisets—Coding for permutation channels with impairments," *IEEE Trans. Inf. Theory*, vol. 64, no. 7, pp. 5156–5169, Jul. 2018.
- [7] A. V. Kuznetsov and A. J. H. Vinck, "A coding scheme for single peak-shift correction in (d, k) -constrained channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1440–1450, Jul. 1993.

- [8] H. H. Lee, R. Kalhor, N. Goela, J. Bolot, and G. M. Church, "Terminator-free template-independent enzymatic DNA synthesis for digital information storage," *Nature Commun.*, vol. 10, no. 1, pp. 1–12, Dec. 2019.
- [9] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Anchor-based correction of substitutions in indexed sets," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 757–761.
- [10] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Coding over sets for DNA storage," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2331–2351, Apr. 2020.
- [11] V. I. Levenshtein and A. J. H. Vinck, "Perfect (d, k) -codes capable of correcting single peak-shifts," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 656–662, Mar. 1993.
- [12] F. J. MacWilliams and N. J. A. Sloane, *The Theory Error-Correcting Codes*. Haarlem, The Netherlands: North-Holland, 1978.
- [13] T. Shinkar, E. Yaakobi, A. Lenz, and A. Wachter-Zeh, "Clustering-correcting codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 81–85.
- [14] J. Sima, R. Gabrys, and J. Bruck, "Optimal systematic t -deletion correcting codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 769–774.
- [15] J. Sima, N. Raviv, and J. Bruck, "Robust indexing-optimal codes for DNA storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 717–722.
- [16] J. Sima, N. Raviv, and J. Bruck, "On coding over sliced information," *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 2793–2807, May 2021.
- [17] W. Song, K. Cai, and K. A. Schouhamer Immink, "Sequence-subset distance and coding for error control in DNA-based data storage," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, pp. 6048–6065, Oct. 2020.
- [18] H. Wei, X. Wang, and M. Schwartz, "On lattice packings and coverings of asymmetric limited-magnitude balls," *IEEE Trans. Inf. Theory*, vol. 67, no. 8, pp. 5104–5115, Aug. 2021.
- [19] J. K. Wolf, "An introduction to tensor product codes and applications to digital storage systems," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Chengdu, China, Oct. 2006, pp. 6–10.
- [20] S. M. H. T. Yazdi, H. M. Kiah, R. Gabrys, and O. Milenkovic, "Mutually uncorrelated primers for DNA-based data storage," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6283–6296, Sep. 2018.
- [21] S. M. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Rep.*, vol. 7, Jul. 2017, Art. no. 5011.

Hengjia Wei received the Ph.D. degree in applied mathematics from Zhejiang University, Hangzhou, Zhejiang, China, in 2014.

He was a Post-Doctoral Fellow with Capital Normal University, Beijing, China, from 2014 to 2016, and a Research Fellow with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, from 2016 to 2019. He is currently a Post-Doctoral Fellow with the School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Israel. His research interests include combinatorial design theory, coding theory, and their intersections. He received the 2017 Kirkman Medal from the Institute of Combinatorics and its Applications.

Moshe Schwartz (Senior Member, IEEE) received the B.A. (*summa cum laude*), M.Sc., and Ph.D. degrees from the Computer Science Department, Technion—Israel Institute of Technology, Haifa, Israel, in 1997, 1998, and 2004, respectively.

He was a Fulbright Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, University of California San Diego, and a Post-Doctoral Researcher with the Department of Electrical Engineering, California Institute of Technology. While on sabbatical from 2012 to 2014, he was a Visiting Scientist at the Massachusetts Institute of Technology (MIT). He is currently a Professor with the School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Israel. His research interests include algebraic coding, combinatorial structures, and digital sequences.

Prof. Schwartz received the 2009 IEEE Communications Society Best Paper Award in signal processing and coding for data storage and the 2020 NVMW Persistent Impact Prize. He served as an Associate Editor for coding techniques and coding theory for the IEEE TRANSACTIONS ON INFORMATION THEORY during 2014–2021. Since 2021, he has been serving as an Area Editor for coding and decoding for the IEEE TRANSACTIONS ON INFORMATION THEORY and an Editorial Board Member for the *Journal of Combinatorial Theory, Series A*.