Reconstruction From Noisy Substrings

Hengjia Wei, Moshe Schwartz, *Fellow, IEEE*, and Gennian Ge

***Abstract*— This paper studies the problem of encoding messages into sequences which can be uniquely recovered from some noisy observations about their substrings. The observed reads comprise consecutive substrings with some given minimum overlap. This coded reconstruction problem has applications in DNA storage. We consider both single-strand reconstruction codes and multi-strand reconstruction codes, where the message is encoded into a single strand or a set of multiple strands, respectively. Various parameter regimes are studied. New codes are constructed, some of whose rates asymptotically attain the upper bounds.**

***Index Terms*— DNA storage, sequence (string) reconstruction, substitution, substring-distant sequences, robust positioning sequences.**

I. Introduction

SEQUENCE (string) reconstruction refers to a large class of problems of reconstructing a sequence from partial (perhaps noisy) observations of it. Instances of this problem include reconstruction from multiple erroneous copies of the sequence [3], [12], [13], some substrings of the sequence [10], [11], all the length-k subsequences [8], [15], [20], and compositions of the sequence's substrings or prefixes/suffixes [1], [18].

In this paper, we shall consider the problem of encoding messages into sequences which can be uniquely recovered from observations about their substrings. This coding problem is motivated by applications in DNA-based data storage systems, where data are encoded to long DNA sequences. In some DNA sequencing technologies (e.g., shotgun sequencing), a long DNA strand is first replicated multiple times, and these replicas are then fragmented into some short substrings so that they could be read. In order to retrieve the data, the original long sequence should be reconstructed based on the observations about these short substrings.

This coded reconstruction problem has been studied in different models with different assumptions on the substrings. Gabrys and Milenkovic [10] considered the problem of reconstructing a sequence of length n from its L-*multispectrum*, i.e., the multiset of all of its length-L substrings. They constructed two classes of reconstruction codes with redundancies 2 and $O(\log\log n)$ for $L > 2\log n$ and $\log n < L \leqslant 2\log n$, respectively. They also studied the noisy settings in which some substrings/observations may be lost or be corrupted by errors, and constructed codes to combat these effects. Subsequently, Marcovich and Yaakobi [16] followed this noisy setup and provided more code constructions. The constructions in [10] and [16] are based on the so-called (L, d)-*substring distant (SD) sequence*, a sequence in which every two length-L substrings are of Hamming distance at least d apart. When $d = 1$, such sequences are also known as L-*substring unique sequences* or L-*repeat free sequences*. Efficient encoding algorithms can be found in [9] for $L > \log n$. For general d, Marcovich and Yaakobi [16] proposed an encoding algorithm of (L, d)-SD sequences for $L > 2\log n$.

Another model is the *torn-paper channel*, which randomly tears the input sequence into small pieces of different sizes. The output of this channel is a set of substrings of the input sequence with no overlap, and the message which is carried by the input sequence should be recovered from these substrings. This problem has been researched in the probabilistic setting in [17], [19], and [21]. Recently, Bar-Lev et al. [2] considered this problem in the worst-case. They studied both the noiseless setup and the noisy setup, and proposed a couple of index-based constructions to encode messages into sequences each of which can be uniquely recovered from its non-overlapping substrings. Furthermore, motivated by DNA sequencing technologies where multiple strings are sequenced simultaneously, they extended the single-strand reconstruction problem to a multi-strand reconstruction problem, where each message is encoded into a set of multiple strings that need to be reconstructed from the mix of their substrings. They constructed multi-strand reconstruction codes whose rates asymptotically behave like those of single-strand reconstruction codes. Another related paper is by Wang et al. [23], which, unlike [2], does not restrict the length of the torn substrings, but rather their number. For this setting they construct codes that attain the upper bound on the rate up to asymptotically small factors.

In a recent paper, Yehezkeally et al. [25] proposed a general model, which includes the two models above as extreme cases.

Manuscript received 7 December 2023; revised 19 June 2024; accepted 25 August 2024. Date of publication 3 September 2024; date of current version 22 October 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFA0712100; in part by the National Natural Science Foundation of China under Grant 11971325, Grant 12231014, and Grant 12371523; in part by Beijing Scholars Program; in part by the Major Key Project of Peng Cheng Laboratory under Grant PCL2024AS103; and in part by Zhejiang Laboratory BioBit Program under Grant 2022YFB507. *(Corresponding author: Hengjia Wei.)*

Hengjia Wei is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China, also with the Peng Cheng Laboratory, Shenzhen 518055, China, and also with the Pazhou Laboratory (Huangpu), Guangzhou 510555, China (e-mail: hjwei05@gmail.com).

Moshe Schwartz is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada, on leave from the School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 8410501, Israel (e-mail: schwartz.moshe@mcmaster.ca).

Gennian Ge is with the School of Mathematical Sciences, Capital Normal University, Beijing 100048, China (e-mail: gnge@zju.edu.cn).

Communicated by C. Carlet, Associate Editor for Sequences and Cryptography.

Digital Object Identifier 10.1109/TIT.2024.3454119

In this model, the reconstruction is based on the sequence's $(L_{\min}, L_{\text{over}})$-*trace*, which is a multiset of substrings where every substring has length at least $L_{\min}$ and the overlap of every two consecutive substrings has length at least L_{over}. They focused on the noiseless setup, and constructed a class of trace reconstruction codes whose rate can asymptotically achieve the upper bound. They also studied the multi-strand reconstruction problem in the L-multispectrum model, and proposed reconstruction codes whose rates are asymptotically 1.

In this paper, we shall follow the model in [25] and study the coding problem for both single-strand reconstruction and multi-strand reconstruction in the noisy setup. We aim to encode a message into a sequence which can be uniquely recovered from its $(L_{\min}, L_{\text{over}}, e)$-*erroneous trace*, where each substring may suffer from at most e substitution errors, or to encode a message into a set of k sequences which can be recovered from the union of their $(L_{\min}, L_{\text{over}}, e)$-erroneous traces. Our contributions are listed as follows.

1) We first give an algorithm which can encode messages into (L, d)-SD sequences for $L = \lceil a \log n \rceil$ where $a > 1$ is an arbitrary real constant. The rates of the encoded sequences asymptotically approach 1. In contrast, the encoding algorithm in [16] requires a single redundancy bit but works only when $L > 2 \log n$.
2) For single-strand reconstruction, by using the proposed encoding algorithm for SD sequences, we construct two classes of $(L_{\min}, L_{\text{over}}, e)$-trace reconstruction codes whose rates asymptotically achieve the upper bound.
3) For multi-strand reconstruction, we present some upper bounds on the rates of multi-strand $(L_{\min}, L_{\text{over}}, e)$-trace reconstruction codes, as well as some code constructions. In some parameter regimes, our constructions yield codes whose rates asymptotically attain the upper bounds. Interestingly, when $\log k = \kappa n$, $L_{\min} = a \log n$ and $L_{\text{over}} = \gamma L_{\min}$, the maximal rates of multi-strand reconstruction codes not only depend on κ, a, γ, but also depend on the congruence class of n modulo $L_{\min} - L_{\text{over}}$.

II. Preliminaries

For a positive integer $n \in \mathbb{N}$, let $[n]$ denote the set $\{0, 1, 2, \ldots, n-1\}$. Let Σ denote a finite alphabet. Throughout this paper, we always consider the binary case, i.e., $\Sigma = \{0, 1\}$, however, our results can be easily generalized to non-binary cases. We use $\log x$ to denote the logarithm of x to base 2. When generalizing our results to the q-ary alphabet case, it suffices to replace the $\log$ with $\log_q$.

Assume $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1}) \in \Sigma^n$ is a sequence over Σ. We denote its length by $|\mathbf{x}| = n$, and its Hamming weight by $\mathrm{wt}_H(\mathbf{x})$. Given two sequences $\mathbf{x}$ and $\mathbf{y}$ over Σ, we denote their concatenation by $\mathbf{x} \circ \mathbf{y}$. If $\mathbf{x}$ and $\mathbf{y}$ have the same length, we use $d_H(\mathbf{x}, \mathbf{y})$ to denote their Hamming distance.

A *substring* of $\mathbf{x}$ is a sequence of the form $(x_a, x_{a+1}, \ldots, x_b)$, where $0 \leqslant a \leqslant b < |\mathbf{x}|$, and we use $\mathbf{x}[a, b]$ to denote it. We also use $\mathbf{x}_{i+[L]}$, where $i \in [n - L + 1]$, to denote the substring of $\mathbf{x}$ which starts at the position i and has length L, i.e., $\mathbf{x}_{i+[L]} = (x_i, x_{i+1}, \ldots, x_{i+L-1}) = \mathbf{x}[i, i + L - 1]$.

A *code* is simply a set $\mathcal{C} \subseteq \Sigma^n$, whose elements are referred to as *codewords*. We say n is the length of the code. The *rate* of the code is defined as $R(\mathcal{C}) = \frac{1}{n} \log|\mathcal{C}|$, and the *redundancy* of the code is $n - n \cdot R(\mathcal{C})$.

A. Reconstruction From the L-Multispectrum

For a sequence $\mathbf{x} \in \Sigma^n$ and a positive integer $L \leqslant n$, the L-*multispectrum of* $\mathbf{x}$, denoted by $\mathcal{S}_L(\mathbf{x})$, is the *multiset* of all its length-L substrings, namely,

$$\mathcal{S}_L(\mathbf{x}) = \left\{\mathbf{x}_{0+[L]}, \mathbf{x}_{1+[L]}, \ldots, \mathbf{x}_{n-L+[L]}\right\}.$$

Note that in this paper, we use a pair of curly brackets to denote a multiset.

If $\mathbf{x}$ can be uniquely reconstructed from its L-multispectrum, without knowledge of a proper subset of Σ^n that contains $\mathbf{x}$, then we say it is L-*reconstructible*. It was proved in [22] that if all the length-$(L-1)$ substrings of $\mathbf{x}$ are distinct, then $\mathbf{x}$ is L-reconstructible. Such a sequence is referred to as an $(L-1)$-*substring unique sequence*. In the works [9], [10], algorithms were proposed to construct a set of L-substring unique sequences of rate approaching 1, where $L = \lceil a \log n \rceil$ for any constant real number $a > 1$.

In [10], Gabrys and Milenkovic further studied the problem of reconstructing sequences from their noisy multispectra. They first considered the scenario where some substrings are not included in the readout spectrum. For a subset $\hat{\mathcal{S}} \subset \mathcal{S}_L(\mathbf{x})$, if the maximum number of consecutive substrings which are not included in $\hat{\mathcal{S}}$ is G, we say $\hat{\mathcal{S}}$ has *maximal coverage gap* G. A code is called an (L, G)-*reconstruction code* if every codeword $\mathbf{x}$ can be uniquely reconstructed from any subset $\hat{\mathcal{S}} \subset \mathcal{S}_L(\mathbf{x})$ with maximal coverage gap G. Gabrys and Milenkovic proposed a construction for such codes [10] by restricting each codeword $\mathbf{x}$ to be $\hat{L}$-substring unique with $\hat{L} < L - G$ and imposing some constraints on their prefixes.

Gabrys and Milenkovic also researched the scenario where the observations about the substrings suffer from substitution errors. Let $\mathcal{Y} = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{m-1}\}$ be a multiset consisting of m strings of length L. If there is a subset $\hat{\mathcal{S}} = \{\mathbf{x}_{i_0}, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{m-1}}\} \subset \mathcal{S}_L(\mathbf{x})$ with maximal coverage gap G such that $d_H(\mathbf{y}_j, \mathbf{x}_{i_j}) \leqslant e$ for all $j \in [m]$, then we say $\mathcal{Y}$ is an (L, G, e)-*constrained erroneous multispectrum of* $\mathbf{x}$.

Example 1: Consider the string $\mathbf{x} = (1, 1, 0, 1, 0, 0, 1)$ and the multiset $\mathcal{Y} = \{(0, 1, 1, 0), (1, 0, 0, 1)\}$. Then $\mathcal{S}_4(\mathbf{x}) = \{(1, 1, 0, 1), (1, 0, 1, 0), (0, 1, 0, 0), (1, 0, 0, 1)\}$. Thus, $\mathcal{Y}$ is a $(4, 2, 1)$-constrained erroneous multispectrum of $\mathbf{x}$, since the subset $\hat{\mathcal{S}} = \{(0, 1, 0, 0), (1, 0, 0, 1)\} \subset \mathcal{S}_4(\mathbf{x})$ matches $\mathcal{Y}$. Note that the substrings in $\mathcal{Y}$ cover the bit x_4 twice. x_4 was subjected to a substitution error in the substring $(0, 1, \mathbf{1}, 0)$ but not in the substring $(1, \mathbf{0}, 0, 1)$.

We note that an (L, G, e)-constrained erroneous multispectrum $\mathcal{Y}$ can be regarded as a product by the following process: When sequencing the string $\mathbf{x}$, the sliding window gives us

$$\begin{array}{cccccccccc}
x_0 & x_1 & x_2 & x_3 & \cdots & x_{L-1} & & & & \\
 & x_1 & x_2 & x_3 & \cdots & x_{L-1} & x_L & & & \\
 & & x_2 & x_3 & \cdots & x_{L-1} & x_L & x_{L+1} & & \\
 & & & x_3 & \cdots & x_{L-1} & x_L & x_{L+1} & x_{L+2} & \\
 & & & & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}$$

From this, some rows are erased, but nowhere more than G consecutive rows. At most e errors are introduced in each row. The rows are then mixed, and the indentation is for presentation only. This process is called *reliable* if in every column the majority is the original symbol. Furthermore, we say $\mathcal{Y}$ is *reliable* if there is at least one reliable process resulting in it.

Example 2: Consider the string $\mathbf{x} = (0, 1, 0, 1, 0, 1, 1)$ and the multiset $\mathcal{Y} = \{(1, 0, 1, 0), (0, 1, 1, 1), (1, 0, 1, 1)\}$. There are two different processes to produce $\mathcal{Y}$:

Process 1.

$$\begin{array}{rcl} 0101 & \longrightarrow & \text{erased} \\ 1010 & \longrightarrow & 1010 \\ 01\mathbf{0}1 & \longrightarrow & 0111 \\ 1011 & \longrightarrow & 1011 \end{array}$$

Process 2.

$$\begin{array}{rcl} 01\mathbf{0}1 & \longrightarrow & 0111 \\ 1010 & \longrightarrow & 1010 \\ 0101 & \longrightarrow & \text{erased} \\ 1011 & \longrightarrow & 1011 \end{array}$$

The first process is reliable, while the second one is not (since after the third sliding window erasure, the correct value for x_2 has no majority). Hence, $\mathcal{Y}$ is a reliable $(4, 1, 1)$-constrained erroneous multispectrum.

A code is called an (L, G, e)-*reconstruction code* if every codeword can be uniquely reconstructed from any of its reliable (L, G, e)-constrained erroneous multispectra.[1] Gabrys and Milenkovic constructed an (L, G, e)-reconstruction code of redundancy $O(\log\log n)$ for $L = 6\log n + O(\log\log n)$. Their construction is based on (L, d)-substring distant sequences, whose definition is presented as follows.

Definition 3: A sequence $\mathbf{w} \in \Sigma^n$ is called (L, d)-*substring distant (SD)* if the minimum Hamming distance of its L-multispectrum is at least d, that is, $d_H(\mathbf{w}_{i+[L]}, \mathbf{w}_{j+[L]}) \geqslant d$ for any $0 \leqslant i < j \leqslant n - L$.

Example 4: Consider the sequence $\mathbf{w} = (1, 1, 0, 1, 1, 1, 0)$. Then

$$\mathcal{S}_4(\mathbf{w}) = \{(1, 1, 0, 1), (1, 0, 1, 1), (0, 1, 1, 1), (1, 1, 1, 0)\}.$$

The distance between any two substrings of $\mathcal{S}_4(\mathbf{w})$ is 2, and so, $\mathbf{w}$ is a $(4, 2)$-SD sequence.

Remark: We observe that an (L, d)-substring distant sequence is also (L', d)-substring distant, for any $L' \geqslant L$. Thus, we may equivalently say that $\mathbf{w} \in \Sigma^n$ is (L, d)-*substring distant (SD)* if $d_H(\mathbf{w}_{i+[L']}, \mathbf{w}_{j+[L']}) \geqslant d$ for any integer $L' \geqslant L$ and $0 \leqslant i < j \leqslant n - L'$. This equivalent definition allows L to be a non-integral rational number, which we shall conveniently use in the future.

In [16], Marcovich and Yaakobi followed the noisy setup of Gabrys and Milenkovic. They studied the case of $G = 0$, i.e., no substring losses. Instead of reconstructing $\mathbf{x}$ from a reliable erroneous multispectrum, they aimed to reconstruct from an $(L, 0, e)$-erroneous multispectrum $\mathcal{Y}$, the so-called *maximum reconstructible-string*, i.e., a string of length n that takes at every position i the majority value of the occurrences of x_i in $\mathcal{Y}$. A sequence $\mathbf{x}$ is called $(L, 0, e)$-*reconstructible*[2] if one can always reconstruct a *unique* maximum reconstructible-string from any of its $(L, 0, e)$-erroneous multispectra. Obviously, if $\mathcal{Y}$ is reliable, then the maximum reconstructible-string is equal to $\mathbf{x}$. In [16], it is assumed that the number of erroneous substrings, t, is less than $L/2$. Then all entries of $\mathbf{x}$ besides the first and last $2t$ entries cannot appear in $\mathcal{Y}$ erroneously more times than they appear correctly.

Proposition 5 ([16, Theorem 16]): If $\mathbf{x}$ is $(L-1, 4e+1)$-SD, then it is $(L, 0, e)$-reconstructible.

The proof of the proposition is given by an explicit reconstruction algorithm, see [16, Algorithm 3], which takes a noisy L-multispectrum of $\mathbf{x}$ as input and computes the distance between the length-$(L-1)$ suffix of a string and the length-$(L-1)$ prefix of another string in the input set. Note that for any two strings that come from consecutive length-L substrings of $\mathbf{x}$, the distance between their suffix and prefix is at most $2e$, since they share the same length-$(L-1)$ substring of $\mathbf{x}$. Hence, in order to distinguish consecutive strings and non-consecutive strings, $\mathbf{x}$ should be $(L-1, 4e+1)$-SD.

For positive integers n, d, L with $d \leqslant L < n$, we use $\mathcal{Z}_n(L, d)$ to denote the set of (L, d)-SD sequences of Σ^n. For fixed d and $a > 1$, Marcovich and Yaakobi showed that the asymptotic rate of the set $\mathcal{Z}(a\log n, d)$ is 1, by using the Lovász Local Lemma. Note that when $a < 1$, even a single $(a\log n)$-substring unique sequence of length n does not exist.

Theorem 6 ([16, Theorem 19]): For fixed d and $a > 1$,

$$\lim_{n\to\infty} \frac{\log|\mathcal{Z}_n(a\log n, d)|}{n} = 1.$$

Marcovich and Yaakobi also presented a deterministic algorithm which uses a single redundancy bit to encode $(a\log n, d)$-SD sequences for $a > 2$.

Theorem 7 ([16, Algorithm 4 and Theorem 25]): Let $d > 0$ be a fixed integer. There is an encoding algorithm which uses a single redundancy bit to encode (L, d)-SD sequences of length n, for

$$L = 2\log n + 2(d - 1 + \epsilon)\log\log n,$$

where $\epsilon > 0$ is a small constant number and n is sufficiently large.

In Section III, we shall present an algorithm which can encode $(a\log n, d)$-SD sequences of length n for any $a > 1$, while its redundancy is $o(n)$. According to Proposition 5, this implies an $(L, 0, e)$-reconstructible code whose rate approaches 1, for $L = \lceil a\log n\rceil + 1$ and $e = \lfloor\frac{d-1}{4}\rfloor$.

B. Reconstruction From an $(L_{\min}, L_{\text{over}})$-Trace

In [25], Yehezkeally et al. studied an extension of the problem of reconstructing from substrings. Let $\mathbf{x} \in \Sigma^n$ be a sequence. A *substring trace* of $\mathbf{x}$ is a multiset of substrings

[1] We emphasize that the multispectrum $\mathcal{Y} = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{m-1}\}$ is just a multiset, and the order/index i of each $\mathbf{y}_i$ cannot be directly read when reconstructing.

[2] The notion here is a bit different from that in [16], where Marcovich and Yaakobi further assumed that there are at most t substrings in $\mathcal{Y}$ each of which is affected by at most e errors and referred to it as a (t, e)-erroneous multispectrum. They proposed two constructions for reconstructible codes: one is independent of t and thus can combat any number of erroneous substrings, while the other one depends on t. In this paper, we focus on reconstructible codes which are independent of t.

$\{\mathbf{x}_{i_0+[L_0]}, \mathbf{x}_{i_1+[L_1]}, \ldots, \mathbf{x}_{i_{m-1}+[L_{m-1}]}\}$ for some positive integer m, where $i_0 < i_1 < \cdots < i_{m-1}$. If $i_0 = 0$, $i_{j+1} \leqslant i_j + L_j$ for all $j < m-1$, and $i_{m-1} + L_{m-1} = n$, then the substring trace is called *complete*. Let $L_{\min}$ and L_{over} be two positive integers such that $L_{\text{over}} < L_{\min} < n$. An $(L_{\min}, L_{\text{over}})$-*trace* is a complete trace such that:

1) every substring has length at least $L_{\min}$, i.e., $L_i \geqslant L_{\min}$ for all $i \in [m]$;
2) the overlap of every two consecutive substrings has length at least L_{over}, i.e., $i_j + L_j - i_{j+1} \geqslant L_{\text{over}}$ for all $j \in [m-1]$.

For a sequence $\mathbf{x}$, let $\mathcal{T}^{L_{\text{over}}}_{L_{\min}}(\mathbf{x})$ denote the set of all $(L_{\min}, L_{\text{over}})$-traces of $\mathbf{x}$. A code $\mathcal{C}$ is referred to as an $(L_{\min}, L_{\text{over}})$-*trace reconstruction code* if $\mathcal{T}^{L_{\text{over}}}_{L_{\min}}(\mathbf{x}) \cap \mathcal{T}^{L_{\text{over}}}_{L_{\min}}(\mathbf{x}') = \emptyset$ for all $\mathbf{x} \neq \mathbf{x}' \in \mathcal{C}$.

Proposition 8 ([25, Lemma 1]): Let $\mathbf{x}$ be an L_{over}-substring unique sequence. Then $\mathbf{x}$ can be uniquely reconstructed from any of its $(L_{\min}, L_{\text{over}})$-traces.

By refining the constructions of substring unique sequences, Yehezkeally et al. obtained the following result.

Theorem 9 ([25, Corollary 6]): There is an $(L_{\min}, L_{\text{over}})$-trace reconstruction code of Σ^n whose rate approaches 1, for $L_{\text{over}} \geqslant \lceil \log n \rceil + 3\lceil \log \log n \rceil + 12$ and sufficiently large n.

They also studied the other parameter regimes.

Lemma 10 ([25, Lemma 8]): If $L_{\min} = a \log n + O(1)$ and $L_{\text{over}} = \gamma L_{\min} + O(1)$ for some $a > 1$ and $0 \leqslant \gamma \leqslant \frac{1}{a}$, then for any $(L_{\min}, L_{\text{over}})$-trace reconstruction code $\mathcal{C} \subseteq \Sigma^n$, its rate $R(\mathcal{C})$ must satisfy

$$R(\mathcal{C}) \leqslant \frac{1 - 1/a}{1 - \gamma} + O\left(\frac{\log \log n}{\log n}\right).$$

Theorem 11 ([25, Theorem 15]): Let $L_{\min} = a \log n$ and $L_{\text{over}} = \gamma L_{\min}$ for some $a > 1$ and $0 \leqslant \gamma \leqslant \frac{1}{a}$. If n is sufficiently large, then there is an $(L_{\min}, L_{\text{over}})$-trace reconstruction code $\mathcal{C} \subseteq \Sigma^n$ with rate

$$R(\mathcal{C}) \geqslant \frac{1 - 1/a}{1 - \gamma} - \frac{(\log n)^{\epsilon}}{a\sqrt{\log n}} - O\left(\frac{1}{\sqrt{\log n}}\right),$$

where $\epsilon > 0$ is a small number which is independent of n.

In this paper, we shall study the problem of reconstructing sequences from their noisy substring traces. Let $\mathcal{Y} = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{m-1}\}$ be a multiset of sequences over Σ, and let $L_j = |\mathbf{y}_j|$ for $j \in [m]$. We say $\mathcal{Y}$ is an $(L_{\min}, L_{\text{over}}, e)$-*erroneous trace of* $\mathbf{x}$ if there exists an $(L_{\min}, L_{\text{over}})$-trace $\{\mathbf{x}_{i_0+[L_0]}, \mathbf{x}_{i_1+[L_1]}, \ldots, \mathbf{x}_{i_{m-1}+[L_{m-1}]}\}$ such that $d_H(\mathbf{y}_j, \mathbf{x}_{i_j+[L_j]}) \leqslant e$ for all $j \in [m]$. Namely, each string $\mathbf{y}_j$ in $\mathcal{Y}$ is an erroneous copy of the substring $\mathbf{x}_{i_j+[L_j]}$ in $\mathbf{x}$ with at most e errors. The index i_j is referred to as the *location of* $\mathbf{y}_j$ *in* $\mathbf{x}$. We note that the location might not be unique since there might be another process resulting in the same $\mathcal{Y}$. For a sequence $\mathbf{x}$ and any of its $(L_{\min}, L_{\text{over}}, e)$-erroneous traces $\mathcal{Y}$, if one can always determine a *unique* location for every $\mathbf{y}_i \in \mathcal{Y}$ in $\mathbf{x}$, then we say $\mathbf{x}$ is $(L_{\min}, L_{\text{over}}, e)$-*trace maximal reconstructible*. Once all the locations of $\mathbf{y}_j$'s are identified, by taking at every position i the majority value of the occurrences of x_i in $\mathcal{Y}$, we can obtain a string which is referred to as the *maximum reconstructible-string* of $\mathcal{Y}$, denoted by $M(\mathcal{Y})$. Since $\mathcal{Y}$ is complete, the length of $M(\mathcal{Y})$ is n. Furthermore, if $\mathcal{Y}$ is reliable[3], then $\mathbf{x} = M(\mathcal{Y})$, i.e., the $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstructible sequence $\mathbf{x}$ can be uniquely reconstructed as long as $\mathcal{Y}$ is reliable.

A code is called an $(L_{\min}, L_{\text{over}}, e)$-*trace maximal reconstruction* code if every codeword $\mathbf{x}$ is $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstructible[4]. In Section IV, we will give two constructions for $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction codes where the number of errors e is fixed. Our results are akin to Theorem 9 and Theorem 11. In particular, when $L_{\text{over}} = a \log n$ for some $a > 1$, we construct a class of $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction codes whose rates approach 1. When $L_{\min} = a \log n$ and $L_{\text{over}} = \gamma L_{\min}$ for some $a > 1$ and $0 \leqslant \gamma \leqslant \frac{1}{a}$, the proposed $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction codes have rates close to $\frac{1-1/a}{1-\gamma}$. These results are summarized in Table I. Our constructions are based on robust positioning sequences and window-weight limited sequences, which are reviewed in Section II-D.

As mentioned above, in order to recover $\mathbf{x}$, we assume that $\mathcal{Y}$ is reliable. This assumption is quite strong, since there is a lack of control over the coverage. This is a feature found in previous works [10], [16]. In this paper, we also consider the case when the majority vote does not correct all the errors, i.e., the case when the maximum reconstructible-string $M(\mathcal{Y})$ is different from $\mathbf{x}$. Let $\mathbf{x}$ be an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstructible string. For an $(L_{\min}, L_{\text{over}}, e)$-erroneous trace $\mathcal{Y}$ of $\mathbf{x}$, if $d_H(M(\mathcal{Y}), \mathbf{x}) \leqslant \tau$, then $\mathcal{Y}$ is referred to as an $(L_{\min}, L_{\text{over}}, e, \tau)$-*erroneous trace*. An $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code is called an $(L_{\min}, L_{\text{over}}, e, \tau)$-*trace reconstruction code* if every codeword $\mathbf{x}$ can be reconstructed from any of its $(L_{\min}, L_{\text{over}}, e, \tau)$-erroneous traces. In Section IV, we will modify our code construction to obtain $(L_{\min}, L_{\text{over}}, e, \tau)$-trace reconstruction codes. We note that a similar problem has been addressed in the scenario where the DNA strands suffer from substitution errors *before* sequencing. Bar-Lev et al. [2] studied this problem in *adversarial torn-paper channel*, where there is no overlap between any two adjacent substrings. Yehezkeally and Polyanskii studied a similar problem for the $(L+1, L)$-trace reconstruction [26]. They introduced the notion of a (t, L)-*resilient repeat-free sequence*, which retains L-substring uniqueness even in the presence of up to t substitution errors. Additionally, they proposed an algorithm for directly encoding such sequences. Interestingly, [26, Lemma 6] shows that an $(L, 2t+1)$-SD sequence is (t, L)-resilient repeat free.

C. Multi-Strand Reconstruction

Motivated by DNA sequencing technologies where multiple DNA strands are sequenced simultaneously, the reconstruction problem has been extended to the multi-strand case in [25] and [2], i.e., reconstructing a *multiset* of k sequences of length n from the union of their traces.

[3]Note that the $\mathcal{Y}$ in Example 2 is reliable, but not complete.

[4]Unlike the noiseless case, in an $(L_{\min}, L_{\text{over}}, e)$-trace reconstruction code it might be possible that two codewords share a common $(L_{\min}, L_{\text{over}}, e)$-erroneous trace. Nevertheless, they cannot have a common reliable trace.

TABLE I
LOWER AND UPPER BOUNDS ON THE CODE RATE OF SINGLE-STRAND $(L_{\min}, L_{\text{over}}, e)$-TRACE MAXIMAL RECONSTRUCTION CODES OF Σ^n

Parameter regimes	Lower bound	Ref.	Upper bound	Ref.
$L_{\text{over}} = \lceil \log n \rceil + (6d+7)\lceil \log \lceil \log n \rceil \rceil + d\lceil \log d \rceil + 5d$ where $d = 4e+1$	$1 - o(1)$	Corollary 22	1	
$L_{\min} = \lceil a \log(n) \rceil$, $L_{\text{over}} = \lceil \gamma L_{\min} \rceil$ where $a > 1$ and $0 \leqslant a\gamma \leqslant 1$	$\frac{1-1/a}{1-\gamma} - o(1)$	Theorem 28 & Theorem 31	$\frac{1-1/a}{1-\gamma} + o(1)$	Lemma 10

Define

$$\mathcal{X}_{n,k} \triangleq \{\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{k-1}\} \ : \ \mathbf{x}_i \in \Sigma^n \text{ for all } i \in [k]\}.$$

Then $|\mathcal{X}_{n,k}| = \binom{k+2^n-1}{k}$. The *rate* of a multi-strand code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ is defined as

$$R(\mathcal{C}) \triangleq \frac{\log|\mathcal{C}|}{\log|\mathcal{X}_{n,k}|}.$$

For a multiset $\mathcal{S} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{k-1}\} \in \mathcal{X}_{n,k}$, its $(L_{\min}, L_{\text{over}})$-*trace* is a (multiset) union $\mathcal{Y} = \bigcup_{i=0}^{k-1} \mathcal{Y}_i$, where each $\mathcal{Y}_i$ is an $(L_{\min}, L_{\text{over}})$-trace of $\mathbf{x}_i$. A code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ is referred to as a *multi-strand* $(L_{\min}, L_{\text{over}})$-*trace reconstruction code* if every codeword can be reconstructed from its $(L_{\min}, L_{\text{over}})$-trace. Two classes of multi-strand trace reconstruction codes whose rates asymptotically attain the upper bound have been constructed in [2] and [25], for $L_{\text{over}} = 0$ or $L_{\text{over}} = L_{\min} - 1$, respectively.

Theorem 12 ([2, Theorem 12]): Suppose that $\log k = o(n)$ and $L_{\min} = a \log(nk)$ with $a > 1$. Then there is a class of multi-strand $(L_{\min}, 0)$-trace reconstruction codes of rate $1 - 1/a - o(1)$.

Theorem 13 ([25, Corollary 23]): Suppose that $\limsup_{n\to\infty} \log k/n < 1$ and $L_{\min} \geqslant \log(nk) + 3\log\log(nk) + 12$. Then there is a class of multi-strand $(L_{\min}, L_{\min} - 1)$-trace reconstruction codes of rate $1 - o(1)$.

In this paper, we will also study the problem of reconstructing multiple strands from their noisy traces. For a multiset $\mathcal{S} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{k-1}\} \in \mathcal{X}_{n,k}$, its $(L_{\min}, L_{\text{over}}, e)$-*erroneous trace* is a (multiset) union $\mathcal{Y} = \bigcup_{i=0}^{k-1} \mathcal{Y}_i$, where each $\mathcal{Y}_i$ is an $(L_{\min}, L_{\text{over}}, e)$-erroneous trace of $\mathbf{x}_i$. We aim to reconstruct $\mathcal{S}$ from its $(L_{\min}, L_{\text{over}}, e)$-erroneous trace. If for any $(L_{\min}, L_{\text{over}}, e)$-erroneous trace $\mathcal{Y}$ of $\mathcal{S}$ and any $\mathbf{y} \in \mathcal{Y}$, it is possible to determine a unique index i such that $\mathbf{y} \in \mathcal{Y}_i$ as well as a unique location of $\mathbf{y}$ in $\mathbf{x}_i$, then we say $\mathcal{S}$ is $(L_{\min}, L_{\text{over}}, e)$-*trace maximal reconstructible*. A code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ is called a *multi-strand* $(L_{\min}, L_{\text{over}}, e)$-*trace maximal reconstruction code* if each of its codewords is $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstructible.

Following the research in [25], we assume that $\limsup_{n\to\infty} \log k/n < 1$, which is of great interest in applications. In Section V, we shall present some upper bounds on the rate of multi-strand trace reconstruction codes and propose some codes whose rates asymptotically attain these bounds. Our results are summarized in Table II and Table III. Among others, when $\log k = \kappa n$ with $0 < \kappa < 1$ and $L_{\min} = \lceil a \log(nk) \rceil$ with $a > 1$, we obtain a class of multi-strand $(L_{\min}, 0, e)$-trace maximal reconstruction codes of rate $\frac{1-1/a}{1-\kappa} + \frac{L^*}{a(1-\kappa)n} - o(1)$, where $L^* \equiv n \pmod{L_{\min}}$. Note that $L^* \in [L_{\min}]$ and $L_{\min} = a\log(nk) = \Theta(n)$. The term $\frac{L^*}{n}$ could be a non-vanishing number, depending on the congruence class of n modulo $L_{\min}$. In contrast, when $\log k = o(n)$, the rate of the multi-strand $(L_{\min}, 0)$-trace maximal reconstruction codes in Theorem 40 is $1 - 1/a - o(1)$, which is the same as that of single-strand reconstruction codes.

D. Robust Positioning Sequences

An (L, d)-substring distant sequence $\mathbf{x}$ is also known as an (L, d)-*robust positioning sequence*, since the contents of any length-L substring can locate the substring's position in $\mathbf{x}$, even if they are corrupted by at most $\lfloor (d-1)/2 \rfloor$ errors. In the context of robust positioning sequences, given L and d, it is of interest to construct a (single) long (L, d)-robust positioning sequence with efficient locating algorithm. This problem, as well as its 2-dimensional extension, has been discussed in [4], [5], [6], [7], and [24]. Among others, Chee et al. [6] constructed a class of (L, d)-robust positioning sequences of length $2^L/(cL^{3d+6.5})$ for some constant number $c > 0$. Their construction was refined in [24] to obtain sequences of length $2^L/(cL^{\lceil (d-1)/2 \rceil + 8})$, whose redundancy[5], $(\lceil (d-1)/2 \rceil + 8)\log L + O(1)$, is close to the lower bound $\lfloor (d-1)/2 \rfloor \log L + O(1)$. The constructions in [6] and [24] require the following notions.

Theorem 14 (d-Auto-Cyclic Sequences [14]): Let $\ell = d\lceil \log d \rceil + 2d$. Set $\mathbf{u}$ to be the sequence

$$\mathbf{u} = 1^d \circ \mathbf{u}_0 \circ \mathbf{u}_1 \circ \cdots \circ \mathbf{u}_{\lceil \log d \rceil}, \text{ where } \mathbf{u}_i = ((1^{2^i} \circ 0^{2^i})^d)[0, d-1].$$

Then for all $1 \leqslant i \leqslant d$, we have that

$$d_H(\mathbf{u}, 0^i \circ \mathbf{u}[0, \ell - i - 1]) \geqslant d,$$

and $\mathbf{u}$ is called a *d-auto-cyclic sequence*.

Definition 15: Let n, L, d be positive integers such that $d < L < n$. We say a sequence $\mathbf{x} \in \Sigma^n$ satisfies the (L, d)-*window weight limited (WWL)* constraint, and is called an (L, d)-*WWL sequence*, if $\text{wt}_H(\mathbf{x}_{i+[L]}) \geqslant d$ for any $i \in [n - L + 1]$.

Proposition 16 ([6, Construction 1 and Theorem 3.7]): Given L and d, choose K such that $\ell < K$ and $K + \ell < L$, where $\ell = d\lceil \log d \rceil + 2d$. Let $\mathbf{u}$ be a d-auto-cyclic vector of length ℓ from Theorem 14 and set $L_p = K + \ell$. Let $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{M-1}$ be a collection of length-$(L - L_p)$ binary vectors satisfying the following conditions:

[5]The redundancy of an (L, d)-robust positioning sequence of length N is defined as $L - \log N$.

TABLE II

LOWER AND UPPER BOUNDS ON THE CODE RATE OF MULTI-STRAND $(L_{\min}, L_{\text{over}}, e)$-TRACE MAXIMAL RECONSTRUCTION CODES OF $\mathfrak{X}_{n,k}$, WHERE $\log k = o(n)$

Parameter regimes	Lower bound	Ref.	Upper bound	Ref.
$L_{\text{over}} = \log(nk) + (24e+13)\log\log(nk) + O(1)$	$1 - o(1)$	Theorem 34	1	
$L_{\min} = \lceil a\log(nk)\rceil$, $L_{\text{over}} = \lceil \gamma L_{\min}\rceil$ where $a > 1$ and $0 \leqslant a\gamma \leqslant 1$	$\frac{1-1/a}{1-\gamma} - o(1)$	Theorem 40	$\frac{1-1/a}{1-\gamma} + o(1)$	Lemma 36
$L_{\min} \leqslant \log(nk) + o(\log(nk))$			$o(1)$	Corollary 37

TABLE III

LOWER AND UPPER BOUNDS ON THE CODE RATE OF MULTI-STRAND $(L_{\min}, L_{\text{over}}, e)$-TRACE MAXIMAL RECONSTRUCTION CODES OF $\mathfrak{X}_{n,k}$, WHERE $\log k = \kappa n + o(n)$ AND $L^* = (n - L_{\text{over}}) \bmod (L_{\min} - L_{\text{over}})$. TO SAVE SPACE, SOME TERMS OF $o(1)$ ARE OMITTED

Parameter regimes	Lower bound	Ref.	Upper bound	Ref.
$L_{\text{over}} = \log(nk) + (24e+13)\log\log(nk) + O(1)$	$1 - o(1)$	Theorem 34	1	
$L_{\min} = \lceil a\log(nk)\rceil$, $L_{\text{over}} = \lceil \gamma L_{\min}\rceil$ where $a > 1$ and $0 \leqslant a\gamma \leqslant 1$	$\frac{1-a\gamma\kappa}{1-\kappa}\left(\frac{1-1/a}{1-\gamma}\right)$	Theorem 40	$\frac{1-a\gamma\kappa}{1-\kappa}\left(\frac{1-1/a}{1-\gamma}\right) + \frac{1/a-\gamma}{(1-\gamma)(1-\kappa)}\frac{L^*}{n}$	Lem. 36
$L_{\text{over}} = 0$, $L_{\min} = \lceil a\log(nk)\rceil$, $a > 1$, and $L^* \leqslant L_{\min} - (1+\epsilon)\log(nk)$	$\frac{1-1/a}{1-\kappa} + \frac{L^*}{a(1-\kappa)n}$	Theorem 42	$\frac{1-1/a}{1-\kappa} + \frac{L^*}{a(1-\kappa)n}$	Lem. 36
$L_{\min} = \log(nk) + o(\log(nk))$ and $L_{\min} - L_{\text{over}} = \Theta(\log(nk))$			$o(1)$	Lem. 38
$L_{\min} = \lceil a\log(nk)\rceil$ with $a < 1$			$o(1)$	Lem. 39

(P1) $\mathbf{s}_i$ is a (K,d)-WWL vector for $i \in [M]$;
(P2) $\mathbf{s}_{i+1}[0, j-1] \circ \mathbf{s}_i[j, L-L_p-1]$ is a (K,d)-WWL vector for $i \in [M-1]$ and $j \in [L-L_p-1]$; and
(P3) the concatenation $\mathbf{s}_0 \circ \mathbf{s}_1 \circ \mathbf{s}_2 \circ \cdots \circ \mathbf{s}_{M-1}$ is an $(L-L_p, d)$-modular robust positioning sequence[6].

Then the sequence

$$\mathbf{s} \triangleq 0^K \circ \mathbf{u} \circ \mathbf{s}_0 \circ 0^K \circ \mathbf{u} \circ \mathbf{s}_1 \circ \cdots \circ 0^K \circ \mathbf{u} \circ \mathbf{s}_{M-1}$$

is an (L,d)-robust positioning (substring distant) sequence.

Theorem 17 ([6, Construction 1A and Corollary 3.12]): Given d and L, set $K = 3\lceil (3\log L)/2\rceil = \frac{9}{2}\log L + O(1)$. There is an explicit construction of sequences $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{M-1}$ of length $L - K - \ell$, where $\log M = L - 3d\log L - 7.5\log L - O(1)$, such that the conditions (P1)–(P3) in Proposition 16 are satisfied.

Remark: We note that for each $i \in [M]$, the concatenation $0^K \circ \mathbf{u} \circ \mathbf{s}_i$ is an (L_p, d)-WWL sequence, since the length-d prefix of $\mathbf{u}$ is 1^d and $\mathbf{s}_i$ is (K,d)-WWL.

The sequence $\mathbf{s}$ in Proposition 16 features an efficient locating algorithm, see [6, Algorithm 3.1]. In Section IV and Section V, we will give several constructions of reconstruction codes. These constructions rely on robust positioning sequences, leveraging the aforementioned locating algorithm to reconstruct codewords from their traces.

[6]A sequence $\mathbf{w}$ is an $(L-L_p, d)$-modular robust positioning sequence if $d_H(\mathbf{w}_{i+[L-L_p]}, \mathbf{w}_{j+[L-L_p]}) \geqslant d$ for any $i \equiv j \pmod{L-L_p}$ and $i \neq j$.

III. ENCODING OF $(a\log n, d)$-SUBSTRING DISTANT SEQUENCES FOR $a > 1$

In this section we shall present an encoding method which can generate a set of $(a\log n, d)$-SD sequences of length n (with $a > 1$, a real number) whose rate asymptotically approaches 1. We shall, in fact, construct (L,d)-SD sequences with $L = \log n + (6d+7)\log\log n + O(1)$, but using the remark following Definition 3, we shall find it more convenient to denote these sequences as $(a\log n, d)$-SD.

We first require some notations. For a sequence $\mathbf{w} \in \Sigma^n$, we say that (i,j) (where $0 \leqslant i < j \leqslant n-L$) is an (L,ρ)-*close window pair* in $\mathbf{w}$ if $d_H(\mathbf{w}_{i+[L]}, \mathbf{w}_{j+[L]}) \leqslant \rho$. Moreover, (i,j) is called *primal*, if for any other (L,ρ)-close window pair (i',j') in $\mathbf{w}$ we have $j \leqslant j'$. Let $\mathbf{x}, \mathbf{x}' \in \Sigma^L$ be two sequences with $d_H(\mathbf{x}, \mathbf{x}') \leqslant \rho$ for some integer $\rho \leqslant L$. Let $p_1, p_2, \ldots, p_{d_H(\mathbf{x},\mathbf{x}')}$ denote the indices of the entries where $\mathbf{x}$ and $\mathbf{x}'$ do not agree. For every $1 \leqslant i \leqslant \rho$ let

$$\mathbf{b}_i = \begin{cases} b(p_i) & \text{if } i \leqslant d_H(\mathbf{x}, \mathbf{x}'), \\ 0^{\lceil \log(L+1)\rceil} & \text{otherwise,} \end{cases} \tag{1}$$

where $b(i)$ is the binary representation of i with $\lceil \log(L+1)\rceil$ symbols. Let

$$\text{EncDist}_{L,\rho}(\mathbf{x}, \mathbf{x}') \triangleq \mathbf{b}_1 \circ \mathbf{b}_2 \circ \cdots \circ \mathbf{b}_\rho.$$

Then $\text{EncDist}_{n,\rho}(\mathbf{x}, \mathbf{x}')$ encodes the difference between $\mathbf{x}$ and $\mathbf{x}'$, and its length is $\rho\lceil \log(L+1)\rceil$. We note that it requires at least $\log(\sum_{i=1}^{\rho}\binom{L}{i})$ bits to encode the difference between $\mathbf{x}$ and $\mathbf{x}'$ as $d_H(\mathbf{x}, \mathbf{x}') \leqslant \rho$. Thus when ρ is fixed, the length of EncDist is close to the lower bound.

Given a fixed d and a sufficiently large n, we are going to present an encoding algorithm which can encode (L,d)-SD sequences of length n. Our method involves concatenation of a sequence $\bar{\mathbf{w}}$, which is both (L_1,d)-SD and (K_1,d)-WWL, and a sequence $\bar{\mathbf{s}}$, which is obtained by invoking Theorem 17 with parameters "$K=K_2$" and "$L=L_2$". The parameters L_1,K_1,L_2,K_2 are defined as follows:

$$\begin{aligned}
L_1 &\triangleq \lceil \log n\rceil + (2d-1)\lceil \log\lceil \log n\rceil\rceil + 6d + \lceil \log(d+1)\rceil,\\
K_1 &\triangleq d\lceil \log\lceil \log n\rceil\rceil + d,\\
L_2 &\triangleq \lceil \log n\rceil + (3d+7)\lceil \log\lceil \log n\rceil\rceil,\\
K_2 &\triangleq 3\left\lceil \frac{3}{2}\log L_2\right\rceil.
\end{aligned}$$

The parameter L for the desired (L,d)-SD sequence is determined by these parameters:

$$L \triangleq \max\{L_1+K_2+K_{\max}+\ell, L_2+2K_1+K_{\max}+\ell\},$$

where $K_{\max}\triangleq \max\{K_1,K_2\}$, and $\ell \triangleq d\lceil \log d\rceil + 2d$ as per Proposition 16 and Theorem 17. In the end of this section, we will show that $L=\log n+(6d+7)\log\log n+O(1)$ when $d\geqslant 5$ is fixed and n is sufficiently large.

Our encoder resembles the encoding algorithms in [10] and [9] and consists of the following three parts:

1) We first use the encoder presented in [14] to encode a message sequence $\mathbf{m}\in\Sigma^{n'}$ into a $(d\lceil\log\lceil\log(n)\rceil\rceil,d)$-WWL sequence $\mathbf{w}$ of length $n-K_1-K_2$. According to [14, Corollary 20], this encoder, denoted by $\mathcal{E}_1$, requires approximately $2d\cdot 2^{\mathcal{F}(n-K_1-K_2,d)-d\lceil\log\lceil\log n\rceil\rceil}$ redundancy symbols, where
$$\mathcal{F}(n,d)=\lceil\log n\rceil+(d-1)(\lceil\log\lceil\log n\rceil\rceil+C)+2$$
for some constant C. Hence,
$$\begin{aligned}
n' &= n-K_1-K_2-2d\cdot 2^{\mathcal{F}(n-K_1-K_2,d)-d\lceil\log\lceil\log n\rceil\rceil}\\
&\geqslant n-K_1-K_2-\\
&\quad 2d\cdot 2^{\lceil\log(n-K_1-K_2)\rceil-\lceil\log\lceil\log(n-K_1-K_2)\rceil\rceil+(d-1)C+2}\\
&\geqslant n-K_1-K_2-d\cdot 2^{(d-1)C+4}\cdot\frac{n-K_1-K_2}{\log(n-K_1-K_2)}\\
&= n-K_1-K_2-\Theta(n/\log n), \qquad (2)
\end{aligned}$$
where the last equality holds as $K_1+K_2=O(\log\log n)$. We note that the complexity of $\mathcal{E}_1$ is $O(n)$, see [14, Lemma 19] and the discussion after it.
2) Then we encode the $(d\lceil\log\lceil\log n\rceil\rceil,d)$-WWL sequence $\mathbf{w}$ into an (L_1,d)-SD sequence $\bar{\mathbf{w}}$ by eliminating the pairs of substrings of small distance and attaching some information about their positions and difference. This encoder, denoted by $\mathcal{E}_2$, is presented in Algorithm 1, and it can additionally guarantee the output sequence is (K_1,d)-WWL.
3) As an output of Algorithm 1, the sequence $\bar{\mathbf{w}}$ is usually shorter than the sequence $\mathbf{w}$. Thus, we need an expansion step to increase the sequence length while keeping the substring-distant property. Let $\mathbf{s}_0,\mathbf{s}_1,\ldots,\mathbf{s}_{M-1}$ be a collection of (K_2,d)-WWL sequences of length L_2-L_p as in Theorem 17. Set
$$\bar{\mathbf{s}}\triangleq 0^{K_{\max}}\circ\mathbf{u}\circ\mathbf{s}_0\circ 0^{K_{\max}}\circ\mathbf{u}\circ\mathbf{s}_1\circ\cdots\circ 0^{K_{\max}}\circ\mathbf{u}\circ\mathbf{s}_{M-1},$$
where $\mathbf{u}$ is the d-auto-cyclic vector of length ℓ from Theorem 14. Finally, let
$$\hat{\mathbf{w}}\triangleq\mathcal{E}_3(\bar{\mathbf{w}})\triangleq(\bar{\mathbf{w}}\circ 0^{K_2}\circ\bar{\mathbf{s}})[0,n-1].$$
We shall show $\hat{\mathbf{w}}$ is the required (L,d)-SD sequence of length n.

We first describe the encoding presented in Algorithm 1. This procedure encodes a $(d\lceil\log\lceil\log n\rceil\rceil,d)$-WWL sequence $\mathbf{w}$ into a sequence $\bar{\mathbf{w}}$ that is simultaneously (L_1,d)-SD and (K_1,d)-WWL. Initiate $\bar{\mathbf{w}}=\mathbf{w}$. If there are no $(L_1,d-1)$-close window pairs in $\bar{\mathbf{w}}$, then the algorithm returns $\bar{\mathbf{w}}$ as the output. We observe that since $\mathbf{w}$ is $(d\lceil\log\lceil\log n\rceil\rceil,d)$-WWL and $K_1\geqslant d\lceil\log\lceil\log n\rceil\rceil$, then $\mathbf{w}$ is also (K_1,d)-WWL.

Otherwise, we choose a primal $(L_1,d-1)$-close window pair, say (i,j). We replace the substring $\bar{\mathbf{w}}_{j+[L_1]}$ with the sequence
$$1^d\circ 0^{d\lceil\log\lceil\log n\rceil\rceil}\circ 1^d\circ B(i)\circ 1^d\circ \mathrm{EncDist}_{L_1,d-1}(\bar{\mathbf{w}}_{i+[L_1]},\bar{\mathbf{w}}_{j+[L_1]})\circ 0^{\lceil\log(d+1)\rceil}\circ 1^d, \quad (3)$$
where $B(i):[n]\longrightarrow\Sigma^{\lceil\log n\rceil+d}$ is the encoding function in [14, Algorithm 2], which can encode integers in $[n]$ into $(d\lceil\log\lceil\log n\rceil\rceil,d)$-WWL sequences in $O(n)$ time. We will show later that the length of this sequence is at most L_1-1. We note that this sequence is (K_1,d)-WWL and contains the information about the position i and the difference between $\bar{\mathbf{w}}_{i+[L_1]}$ and $\bar{\mathbf{w}}_{j+[L_1]}$. Moreover, the substring $0^{d\lceil\log\lceil\log n\rceil\rceil}$ serves as a marker which indicates the position j of the removed substring $\bar{\mathbf{w}}_{j+[L_1]}$.

We shall repeat this procedure until there are no $(L_1,d-1)$-close window pairs in $\bar{\mathbf{w}}$. But in order to ensure that $\mathbf{w}$ can be recovered from the output of the algorithm, we need more tricks. We note that in [10] the inserted sequences always start with a marker $0^{2\log\log n}$ and end with a symbol '1'. This pattern together with the rule that only the primal pairs can be chosen and replaced guarantees that after each replacement the latest inserted substring always starts with the rightmost $0^{2\log\log n}$ in $\bar{\mathbf{w}}$. Due to this property, we have a decoding algorithm which can recover $\mathbf{w}$ from $\bar{\mathbf{w}}$: Let $\bar{\mathbf{w}}^{(k)}$ denote the sequence $\bar{\mathbf{w}}$ after the k-th replacement. One can search for the rightmost $0^{2\log\log n}$ in $\bar{\mathbf{w}}^{(k)}$ to find the position j of the inserted substring in the k-th replacement. By replacing the inserted substring with the removed substring, one can recover $\bar{\mathbf{w}}^{(k-1)}$ from $\bar{\mathbf{w}}^{(k)}$. Doing this iteratively, one can eventually recover $\mathbf{w}$ from $\bar{\mathbf{w}}$.

In our encoding, the inserted substring should always contain 1^d as both prefix and suffix to maintain the property of being (K_1,d)-WWL. We have to modify the substring $0^{\lceil\log(d+1)\rceil}$ in (3) to ensure the latest inserted substring always starts with the rightmost $1^d\circ 0^{d\lceil\log\lceil\log n\rceil\rceil}$ in $\bar{\mathbf{w}}$. Let j_p and j be the positions of the removed substrings in the previous replacement and in the current replacement, respectively. Since we only choose the primal pairs, necessarily, $j>j_p-L_1$. If $j>j_p-L_1+d$, then we still replace the substring $\bar{\mathbf{w}}_{j+[L_1]}$ with the sequence in (3), since the marker $0^{d\lceil\log\lceil\log n\rceil\rceil}$ which

is inserted in the previous replacement will be destroyed by the suffix 1^d of this inserted sequence. If $j_p - L_1 < j \leqslant j_p - L_1 + d$, we first set $\bar{\mathbf{w}}[j_p + d]$ to be '1' to destroy the previous marker $0^{d\lceil \log \lceil \log n \rceil \rceil}$. Then we replace $\bar{\mathbf{w}}_{j+[L_1]}$ with the sequence

$$1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil} \circ 1^d \circ B(i) \circ 1^d \circ \mathrm{EncDist}_{L_1,d-1}(\bar{\mathbf{w}}_{i+[L_1]}, \bar{\mathbf{w}}_{j+[L_1]}) \circ b(j - j_p + L_1) \circ 1^d, \tag{4}$$

where $b(j-j_p+L_1)$ is the binary encoding of $j-j_p+L_1$ with $\lceil \log(d+1) \rceil$ symbols, since $1 \leqslant j - j_p + L_1 \leqslant d$.

Note that the substring $B(i)$ and the substring $\mathrm{EncDist}_{L_1,d-1}(\bar{\mathbf{w}}_{i,L_1}, \bar{\mathbf{w}}_{j,L_1})$ have length $\lceil \log n \rceil + d$ and length at most $(d-1)(\lceil \log \lceil \log n \rceil \rceil + 1)$, respectively. It follows that in the loop we replace substrings of length L_1 with substrings of length at most

$$\begin{aligned} &4d + d\lceil \log \lceil \log n \rceil \rceil + (\lceil \log n \rceil + d) \\ &\quad + (d-1)\lceil \log(L_1+1) \rceil + \lceil \log(d+1) \rceil \\ &\leqslant 4d + d\lceil \log \lceil \log n \rceil \rceil + (\lceil \log n \rceil + d) \\ &\quad + (d-1)(\lceil \log \lceil \log n \rceil \rceil + 1) + \lceil \log(d+1) \rceil \\ &= L_1 - 1, \end{aligned}$$

where the first inequality is obtained by noting that for all sufficiently large n we have $L_1 + 1 \leqslant 2\lceil \log n \rceil$. Hence, the loop will execute at most $|\mathbf{w}| - L_1 + 1$ times and the algorithm will terminate eventually.

Algorithm 1 Primal Pair Elimination Encoder $\mathcal{E}_2$ for Generating (L_1, d)-SD Sequences

Input: a $(d\lceil \log \lceil \log n \rceil \rceil, d)$-WWL sequence $\mathbf{w} \in \Sigma^{n-K_1-K_2}$
Output: a sequence $\bar{\mathbf{w}} \in \Sigma^{\leqslant n-K_1-K_2}$

Set $\bar{\mathbf{w}} = \mathbf{w}$ and $j_p = 0$
while there are two length-L_1 substrings in $\bar{\mathbf{w}}$ whose Hamming distance is at most $d-1$ **do**
  Suppose (i, j) is a primal $(L_1, d-1)$-close window pair in $\bar{\mathbf{w}}$ (then necessarily $j > j_p - L_1$)
  if $j > j_p - L_1 + d$ **then**
    Remove the substring of length L_1 starting at position j and replace it with the sequence

$$1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil} \circ 1^d \circ B(i) \circ 1^d \circ \mathrm{EncDist}_{L_1,d-1}(\bar{\mathbf{w}}_{i+[L_1]}, \bar{\mathbf{w}}_{j+[L_1]}) \circ 0^{\lceil \log(d+1) \rceil} \circ 1^d$$

  else
    Set $\bar{\mathbf{w}}[j_p + d]$ to be '1'
    Remove the substring of length L_1 starting at position j and replace it with the sequence

$$1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil} \circ 1^d \circ B(i) \circ 1^d \circ \mathrm{EncDist}_{L_1,d-1}(\bar{\mathbf{w}}_{i+[L_1]}, \bar{\mathbf{w}}_{j+[L_1]}) \circ b(j - j_p + L_1) \circ 1^d$$

  end if
  $j_p \leftarrow j$
end while
return $\bar{\mathbf{w}}$

Lemma 18: The output sequence $\bar{\mathbf{w}}$ is (K_1, d)-WWL and (L_1, d)-SD, and the input sequence $\mathbf{w}$ can be recovered from $\bar{\mathbf{w}}$, for all sufficiently large n.

Proof: The while loop ensures that the output $\bar{\mathbf{w}}$ of Algorithm 1 is an (L_1, d)-SD sequence. Moreover, since $\mathbf{w}$ is $(d\lceil \log \lceil \log n \rceil \rceil, d)$-WWL and $K_1 = d\lceil \log \lceil \log n \rceil \rceil + d$, one can tediously verify that for all large enough n, $\bar{\mathbf{w}}$ is (K_1, d)-WWL. In particular, even if $\mathrm{EncDist}_{L_1,d-1}(\bar{\mathbf{w}}_{i+[L_1]}, \bar{\mathbf{w}}_{j+[L_1]})$ is all zeros, for all large enough n

$$K_1 - \left| \mathrm{EncDist}_{L_1,d-1}(\bar{\mathbf{w}}_{i+[L_1]}, \bar{\mathbf{w}}_{j+[L_1]}) \circ 0^{\lceil \log(d+1) \rceil} \right| \geqslant d,$$

and a substring of length K_1 containing $\mathrm{EncDist}_{L_1,d-1}(\bar{\mathbf{w}}_{i+[L_1]}, \bar{\mathbf{w}}_{j+[L_1]}) \circ 0^{\lceil \log(d+1) \rceil}$ must also contain at least d of the surrounding 1's.

Next, we show after each replacement the latest inserted substring always starts with the rightmost $1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil}$. Let $\bar{\mathbf{w}}^{(k)}$ be the sequence $\bar{\mathbf{w}}$ after the k-th replacement. We prove this by induction. When $k = 1$, since $\mathbf{w} = \bar{\mathbf{w}}^{(0)}$ is $(d\lceil \log \lceil \log n \rceil \rceil, d)$-WWL, the marker $1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil}$ appears exactly once in $\bar{\mathbf{w}}^{(1)}$, and so the claim holds. Now, in the k-th replacement, j denotes the position of the substring removed in this replacement, while j_p denotes the position of the substring removed in the $(k-1)$-th replacement. According to the inductive assumption, the rightmost $1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil}$ in $\bar{\mathbf{w}}^{(k-1)}$ starts at the position j_p. If $j \geqslant j_p$, then the rightmost $1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil}$ in $\bar{\mathbf{w}}^{(k)}$ is $\bar{\mathbf{w}}^{(k)}_{j+[d\lceil \log \lceil \log n \rceil \rceil + d]}$. If $j_p - L_1 + d < j < j_p$, the overlap of $\bar{\mathbf{w}}^{(k-1)}_{j+[L_1]}$ and $\bar{\mathbf{w}}^{(k-1)}_{j_p+[L_1]}$ has length greater than d. Since the sequence which is inserted in the k-th replacement ends with a symbol '1', it can destroy the marker in $\bar{\mathbf{w}}^{(k-1)}_{j_p+[L_1]}$. If $j_p - L_1 < j \leqslant j_p - L_1 + d$, we set $\bar{\mathbf{w}}^{(k)}[j_p + d]$ to be '1' to destroy the marker in $\bar{\mathbf{w}}^{(k-1)}_{j_p+[L_1]}$. In all cases, the rightmost $1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil}$ in $\bar{\mathbf{w}}^{(k)}$ is always $\bar{\mathbf{w}}^{(k)}_{j+[d\lceil \log \lceil \log n \rceil \rceil + d]}$.

Now, given the sequence $\bar{\mathbf{w}}^{(k)}$, we first search for the rightmost $1^d \circ 0^{d\lceil \log \lceil \log n \rceil \rceil}$ in $\bar{\mathbf{w}}^{(k)}$ to determine the position j. Then from the substring $\bar{\mathbf{w}}^{(k)}_{j+[L_1-1]}$ we can decode i, the difference between $\bar{\mathbf{w}}^{(k-1)}_{i+[L_1]}$ and $\bar{\mathbf{w}}^{(k-1)}_{j+[L_1]}$, and $b(j - j_p + L_1)$. Note that $\bar{\mathbf{w}}^{(k-1)}_{i+[\min\{L_1, j-i\}]} = \bar{\mathbf{w}}^{(k)}_{i+[\min\{L_1, j-i\}]}$. So we can recover $\bar{\mathbf{w}}^{(k-1)}_{j+[L_1]}$. We remove $\bar{\mathbf{w}}^{(k)}_{j+[L_1-1]}$ from $\bar{\mathbf{w}}^{(k)}$ and replace it with $\bar{\mathbf{w}}^{(k-1)}_{j+[L_1]}$. If $b(j - j_p + L_1) \neq 0^{\lceil \log(d+1) \rceil}$, we further set the symbol in the position $j_p + d$ to be '0'. In this way, we recover the sequence $\bar{\mathbf{w}}^{(k-1)}$. We repeat this procedure until there is no substring $0^{d \log \log n}$. Then the resulting sequence is the required $\mathbf{w}$. ∎

Now, we need to extend the sequence $\bar{\mathbf{w}}$ to a long sequence of length n while keeping the property of being (L, d)-SD.

Lemma 19: Assume n is sufficiently large. Let $\bar{\mathbf{w}}$ be an output of Algorithm 1. Recall that $K_2 = 3\lceil \frac{3}{2} \log L_2 \rceil$. By invoking Theorem 17 with parameters "$K = K_2$" and "$L = L_2$", we get a collection of (K_2, d)-WWL sequences $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{M-1}$ of length $L_2 - L_p$, where $L_p = K_2 + d\lceil \log d \rceil + 2d$. Let

$$\bar{\mathbf{s}} \triangleq 0^{K_{\max}} \circ \mathbf{u} \circ \mathbf{s}_0 \circ 0^{K_{\max}} \circ \mathbf{u} \circ \mathbf{s}_1 \circ \cdots \circ 0^{K_{\max}} \circ \mathbf{u} \circ \mathbf{s}_{M-1},$$

where $K_{\max} = \max\{K_1, K_2\}$. Set

$$\hat{\mathbf{w}} = \mathcal{E}_3(\bar{\mathbf{w}}) \triangleq (\bar{\mathbf{w}} \circ 0^{K_2} \circ \bar{\mathbf{s}})[0, n-1].$$

Then $\hat{\mathbf{w}}$ is a (K, d)-WWL and (L, d)-SD sequence where $K = 2(K_1 + K_2)$ and $L = \max\{L_1 + K_2 + K_{\max} + \ell, L_2 + 2K_1 + K_{\max} + \ell\}$. Moreover, $\bar{\mathbf{w}}$ can be recovered from $\hat{\mathbf{w}}$.

Proof: We first prove that $\bar{\mathbf{s}}$ is a $(K_{\max} + K_2, d)$-WWL and $(L_2 + K_{\max} - K_2, d)$-SD sequence of length at least n. According to the construction, the length of $\bar{\mathbf{s}}$ is $M(L_2 + K_{\max} - K_2) \geqslant ML_2$. Recall that $\log M = L_2 - 3d \log L_2 - 7.5 \log L_2 - O(1)$ and $L_2 = \lceil \log n \rceil + (3d+7)\lceil \log \lceil \log n \rceil \rceil$. Then

$$ML_2 = 2^{L_2 - 3d \log L_2 - 6.5 \log L_2 - O(1)} = \frac{2^{L_2}}{2^{O(1)} L_2^{3d+6.5}} \geqslant \frac{n(\log n)^{3d+7}}{2^{O(1)}(\log n + (3d+6.5)\log\log n)^{3d+6.5}} > n. \quad (5)$$

Hence, $\bar{\mathbf{s}}$ has length at least n. Note that each $\mathbf{s}_i$ is a (K_2, d)-WWL sequence and the length-d prefix of $\mathbf{u}$ is 1^d. It follows that $\bar{\mathbf{s}}$ is a $(K_{\max}+K_2, d)$-WWL sequence. Moreover, note that the sequences $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{M-1}$ satisfy the conditions (P1)-(P3) with "$K = K_2$". If $K_2 \geqslant K_1$ (namely, $K_{\max} = K_2$), then by Proposition 16, the sequence $\bar{\mathbf{s}}$ is an (L_2, d)-SD sequence, hence also an $(L_2 + K_{\max} - K_2, d)$-SD sequence. If $K_2 < K_1$, since the property of being (K_2, d)-WWL implies the property of being $(K_{\max}, d)$-WWL, the sequences $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{M-1}$ also satisfy the conditions (P1)-(P3) with "$K = K_{\max}$"[7]. Again, by Proposition 16, the sequence $\bar{\mathbf{s}}$ is an $(L_2 + K_{\max} - K_2, d)$-SD sequence.

We have shown that $\bar{\mathbf{s}}$ is a $(K_{\max} + K_2, d)$-WWL sequence in the above paragraph and $\bar{\mathbf{w}}$ is a (K_1, d)-WWL sequence in Lemma 18. By using the fact that $K_1 > d$ and that the $\mathbf{u}$ substring of $\bar{\mathbf{s}}$ starts with 1^d, it follows that the sequence $\hat{\mathbf{w}} = (\bar{\mathbf{w}} \circ 0^{K_2} \circ \bar{\mathbf{s}})[0, n-1]$ is $(K_1 + K_2 + K_{\max}, d)$-WWL. Since $2(K_1+K_2) \geqslant K_1+K_2+K_{\max}$, it is also (K, d)-WWL, where $K = 2(K_1+K_2)$ as stated in this lemma. Now, we shall show that it is also (L, d)-SD. For any two substrings $\hat{\mathbf{w}}_{i+[L]}$ and $\hat{\mathbf{w}}_{j+[L]}$ with $i, j \in [n-L+1]$ and $i < j$, we consider the following cases:

Case 1: $i < j \leqslant |\bar{\mathbf{w}}| - L_1$. Then

$$d_H(\hat{\mathbf{w}}_{i+[L]}, \hat{\mathbf{w}}_{j+[L]}) \geqslant d_H(\bar{\mathbf{w}}_{i+[L_1]}, \bar{\mathbf{w}}_{j+[L_1]}) \geqslant d,$$

where the first inequality holds since $L \geqslant L_1$ and the second inequality holds since $\bar{\mathbf{w}}$ is an (L_1, d)-SD sequence.

Case 2: $i \leqslant |\bar{\mathbf{w}}| - L_1$ and $|\bar{\mathbf{w}}| - L_1 + 1 \leqslant j \leqslant |\bar{\mathbf{w}}|$. Since $L - L_1 \geqslant K_2 + K_{\max} + \ell$, where ℓ is the length of $\mathbf{u}$, then $\hat{\mathbf{w}}_{j+[L]}$ must contain $0^{K_2+K_{\max}} \circ \mathbf{u}$ as a substring. Assume that $\hat{\mathbf{w}}_{j+\delta+[K_2+K_{\max}+\ell]} = 0^{K_2+K_{\max}} \circ \mathbf{u}$ for some $\delta \in [L_1]$. If $j - i \leqslant d$, then

$$\begin{aligned} &d_H(\hat{\mathbf{w}}_{i+[L]}, \hat{\mathbf{w}}_{j+[L]}) \\ &\geqslant d_H(\hat{\mathbf{w}}_{i+\delta+K_2+K_{\max}+[\ell]}, \hat{\mathbf{w}}_{j+\delta+K_2+K_{\max}+[\ell]}) \\ &= d_H(0^{j-i} \circ \mathbf{u}[0, \ell-(j-i)-1], \mathbf{u}) \geqslant d, \end{aligned}$$

where the last inequality follows from the definition of a d-auto-cyclic sequence. If $d < j - i \leqslant K_2 + K_{\max}$, since the prefix of $\mathbf{u}$ is 1^d, then

$$\begin{aligned} &d_H(\hat{\mathbf{w}}_{i+[L]}, \hat{\mathbf{w}}_{j+[L]}) \\ &\geqslant d_H(\hat{\mathbf{w}}_{i+\delta+K_2+K_{\max}+[d]}, \hat{\mathbf{w}}_{j+\delta+K_2+K_{\max}+[d]}) \\ &= d_H(0^d, 1^d) = d. \end{aligned}$$

If $j - i > K_2 + K_{\max}$, then $i + \delta + K_2 + K_{\max} < j + \delta$, and so, $\hat{\mathbf{w}}_{i+\delta+[K_2+K_{\max}]}$ is a substring of $\bar{\mathbf{w}}$. Hence,

$$\begin{aligned} &d_H(\hat{\mathbf{w}}_{i+[L]}, \hat{\mathbf{w}}_{j+[L]}) \\ &\geqslant d_H(\hat{\mathbf{w}}_{i+\delta+[K_2+K_{\max}]}, \hat{\mathbf{w}}_{j+\delta+[K_2+K_{\max}]}) \\ &= d_H(\hat{\mathbf{w}}_{i+\delta+[K_2+K_{\max}]}, 0^{K_2+K_{\max}}) \geqslant d, \end{aligned}$$

where the last inequality holds since $\bar{\mathbf{w}}$ is a (K_1, d)-WWL sequence.

Case 3 and Case 4, which now follow, together cover the case of $i \leqslant |\bar{\mathbf{w}}| - L_1$ and $j > |\bar{\mathbf{w}}|$ and the case of $|\bar{\mathbf{w}}| - L_1 < i < |\bar{\mathbf{w}}|$ and $i < j$,

Case 3: $i \leqslant |\bar{\mathbf{w}}| - (L_2 + 2K_1 - K_2)$ $(\leqslant |\bar{\mathbf{w}}| - L_1)$ and $j > |\bar{\mathbf{w}}|$. Denote $L' \triangleq (L_2 - K_2) + 2K_1$. Then $L \geqslant L'$. Note that $\hat{\mathbf{w}}_{j+[L']}$ always contains 0^{K_1} as a substring, and $\hat{\mathbf{w}}_{i+[L']}$ is a substring of $\bar{\mathbf{w}}$, which is (K_1, d)-WWL. Hence,

$$d_H(\hat{\mathbf{w}}_{i+[L]}, \hat{\mathbf{w}}_{j+[L]}) \geqslant d_H(\hat{\mathbf{w}}_{i+[L']}, \hat{\mathbf{w}}_{j+[L']}) \geqslant d.$$

Case 4: $|\bar{\mathbf{w}}| - (L_2 + 2K_1 - K_2) + 1 \leqslant i < |\bar{\mathbf{w}}|$ and $i < j$. Since $L \geqslant (L_2 + 2K_1 - K_2) + K_2 + K_{\max} + \ell$, $\hat{\mathbf{w}}_{i+[L]}$ must contain $0^{K_2+K_{\max}} \circ \mathbf{u}$ as a substring. If $j - i \leqslant K_2 + K_{\max}$, then $\hat{\mathbf{w}}_{j+[L]}$ must contain $\mathbf{u}$ as a substring, and so, with the same argument as that in Case 2, one can show that $d_H(\hat{\mathbf{w}}_{i+[L]}, \hat{\mathbf{w}}_{j+[L]}) \geqslant d$. If $j - i > K_2 + K_{\max}$, assume that $\hat{\mathbf{w}}_{i+\delta'+[K_2+K_{\max}]}$ is the all-zero substring of length $K_2 + K_{\max}$. Then $j + \delta' > i + \delta' + K_2 + K_{\max}$. It follows that $\hat{\mathbf{w}}_{j+\delta'+[K_2+K_{\max}]}$ is a substring of $\bar{\mathbf{s}}$, which is $(K_2+K_{\max}, d)$-WWL. Hence,

$$\begin{aligned} &d_H(\hat{\mathbf{w}}_{i+[L]}, \hat{\mathbf{w}}_{j+[L]}) \\ &\geqslant d_H(\hat{\mathbf{w}}_{i+\delta'+[K_2+K_{\max}]}, \hat{\mathbf{w}}_{j+\delta'+[K_2+K_{\max}]}) \\ &\geqslant d. \end{aligned}$$

Case 5: $|\bar{\mathbf{w}}| \leqslant i < j$. Then

$$\begin{aligned} &d_H(\hat{\mathbf{w}}_{i+[L]}, \hat{\mathbf{w}}_{j+[L]}) \\ &\geqslant d_H(\hat{\mathbf{w}}_{i+K_2+[L-K_2]}, \hat{\mathbf{w}}_{j+K_2+[L-K_2]}) \\ &= d_H(\bar{\mathbf{s}}_{i-|\bar{\mathbf{w}}|+[L-K_2]}, \bar{\mathbf{s}}_{j-|\bar{\mathbf{w}}|+[L-K_2]}) \geqslant d, \end{aligned}$$

where the second inequality holds since $L - K_2 \geqslant L_2 + K_{\max} - K_2$ and $\bar{\mathbf{s}}$ is $(L_2 + K_{\max} - K_2, d)$-SD.

Finally, note that in the sequence $\hat{\mathbf{w}}$ there is exactly one run of '0' which has length at least $K_2 + K_{\max}$. So we can search for the rightmost $0^{K_2+K_{\max}}$ in $\hat{\mathbf{w}}$ and remove this substring as well as the suffix after it to recover the sequence $\bar{\mathbf{w}}$. ■

Theorem 20: Let $\mathcal{E}_{\mathtt{SD}}(\cdot) \triangleq \mathcal{E}_3(\mathcal{E}_2(\mathcal{E}_1(\cdot)))$. Then, for fixed d and sufficiently large n, $\mathcal{E}_{\mathtt{SD}} : \Sigma^{n'} \to \Sigma^n$ is invertible and can encode sequences of $\Sigma^{n'}$ into (K, d)-WWL and (L, d)-SD sequences where $K = (2d+9)\log\log n + O(1)$ and

$$L = \lceil \log n \rceil + (6d+7)\lceil \log \lceil \log n \rceil \rceil + d\lceil \log d \rceil + 5d$$

[7]In this case, we take "$L = L_2 + K_{\max} - K_2$", "$K = K_{\max}$", "$L_p = K + \ell$", and so, "$L - L_p = L_2 - K_2 - \ell$", which is equal to the length of the $\mathbf{s}_i$'s.

when $d \geqslant 5$, or

$$L \leqslant \lceil \log n \rceil + (5d + 11.5)\lceil \log \lceil \log n \rceil \rceil + d\lceil \log d \rceil + 4d + 7.5,$$

otherwise.

Moreover, $n - n' = \Theta(n/\log n)$, and so, we have that

$$\lim_{n\to\infty} \frac{n'}{n} = 1.$$

Proof: The statement about $\mathcal{E}_{\mathtt{SD}}$ follows from Lemma 18 and Lemma 19. Recall that the encoder $\mathcal{E}_1$ requires $\Theta(n/\log n)$ redundancies (see (2)) and $K_1 + K_2 = \Theta(\log\log n)$. Hence,

$$n - n' = K_1 + K_2 + \Theta(n/\log n) = \Theta(n/\log n).$$

It remains to estimate the value of L. Since d is fixed and n is sufficiently large, $L_1 + K_2 < L_2 + 2K_1$. It follows that $L = \max\{L_1 + K_2 + K_{\max} + \ell, L_2 + 2K_1 + K_{\max} + \ell\} = L_2 + 2K_1 + K_{\max} + \ell$. Note that $K_1 > K_2$ if and only if $d \geqslant 5$. Hence,

$$L = \begin{cases} L_2 + 3K_1 + \ell & \text{if } d \geqslant 5, \\ L_2 + 2K_1 + K_2 + \ell & \text{otherwise.} \end{cases}$$

Substituting the values of K_1, L_2 and ℓ, and noting that

$$\begin{aligned} K_2 &= 3\lceil 1.5 \log L_2 \rceil \leqslant 3\lceil 1.5 \log \lceil \log n \rceil + 1.5 \rceil \\ &\leqslant 4.5 \lceil \log \lceil \log n \rceil \rceil + 7.5, \end{aligned}$$

the conclusion then follows. ∎

IV. Generalized Reconstruction From Noisy Substring Trace

In this section, we are going to give constructions of $(L_{\min}, L_{\text{over}}, e)$-trace (maximal) reconstruction codes. Our first result generalizes Proposition 5 and Proposition 8, which shows that the property of being (L_{over}, d)-substring distant implies the property of being $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstructible.

Proposition 21: Suppose that $L_{\min} > L_{\text{over}}$. If a sequence $\mathbf{x} \in \Sigma^n$ is $(L_{\text{over}}, 4e + 1)$-substring distant, then $\mathbf{x}$ is $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstructible.

Proof: Let $\mathcal{Y} = \{\mathbf{y}^{(0)}, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(m-1)}\}$ be an $(L_{\min}, L_{\text{over}}, e)$-erroneous trace of $\mathbf{x}$ where the location of each $\mathbf{y}^{(j)}$ in $\mathbf{x}$ is i_j. Since $\mathbf{x}$ is $(L_{\text{over}}, 4e + 1)$-substring distant, for any two substrings $\mathbf{y}^{(j)}$ and $\mathbf{y}^{(j')}$ and their any two subsubstrings $\mathbf{y}^{(j)}_{k+[L_{\text{over}}]}$ and $\mathbf{y}^{(j')}_{k'+[L_{\text{over}}]}$, we have that

$$d_H\left(\mathbf{y}^{(j)}_{k+[L_{\text{over}}]}, \mathbf{y}^{(j')}_{k'+[L_{\text{over}}]}\right) \begin{cases} \geqslant 2e+1 & \text{if } i_j + k \neq i_{j'} + k', \\ \leqslant 2e & \text{if } i_j + k = i_{j'} + k'. \end{cases}$$

Therefore, $\mathbf{y}^{(0)}$ can be identified as the unique substring $\mathbf{y} \in \mathcal{Y}$ whose length-L_{over} prefix is of Hamming distance at least $2e+1$ from every length-L_{over} subsubstring of any other $\mathbf{y}' \in \mathcal{Y}\backslash\{\mathbf{y}\}$. Denote the length-$L_{\text{over}}$ suffix of $\mathbf{y}^{(0)}$ as $\mathbf{s}_0$. Then we can identify the substrings $\mathbf{y}$'s in $\mathcal{Y}$ which overlap $\mathbf{y}^{(0)}$ at least L_{over} positions, since each of them contains a unique length-L_{over} subsubstring $\mathbf{w}$ whose distance from $\mathbf{s}_0$ is at most $2e$. Furthermore, the locations of these substrings in $\mathbf{x}$ can be determined by aligning the subsubstring $\mathbf{w}$ and the suffix $\mathbf{s}_0$. Assume that there are m' such substrings. Then we have identified the substrings $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(m')} \in \mathcal{Y}$. Next, we consider the length-L_{over} suffix of $\mathbf{y}^{(m')}$ and we can identify all the substrings in $\mathcal{Y}$ which overlap $\mathbf{y}^{(m')}$ at least L_{over} positions. We repeat the procedure above. Finally, we can determine the location of every substring $\mathbf{y} \in \mathcal{Y}$ in $\mathbf{x}$. ∎

Combining Theorem 20 and Proposition 21, we have the following result.

Corollary 22: Suppose that $L_{\text{over}} = \lceil \log n \rceil + (24e + 13)\lceil \log \lceil \log n \rceil \rceil + (4e + 1)\lceil \log(4e + 1) \rceil + 20e + 5$ and $L_{\min} > L_{\text{over}}$. If n is sufficiently large, then there is an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code of Σ^n whose rate is $1 - o(1)$.

Now, we consider another parameter regime. Suppose that

$$\begin{aligned} L_{\min} &= \lceil a \log n \rceil, \\ L_{\text{over}} &= \lceil \gamma L_{\min} \rceil, \end{aligned}$$

where $a > 1$ and $0 < a\gamma \leqslant 1$ are real constants. We are going to construct an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code whose rate approaches $\frac{1-1/a}{1-\gamma}$. The basic idea of our code construction is similar to the one in [16] for the noiseless scenario: A message $\mathbf{m}$ is encoded into a codeword $\mathbf{w} = \mathbf{w}_0 \circ \mathbf{w}_1 \circ \cdots \circ \mathbf{w}_{2^I - 1}$ such that

(i) the index i can be decoded from any length-$L_{\min}$ substring of $\mathbf{w}_i$ or $\mathbf{w}_i \circ \mathbf{w}_{i+1}$ even if the substring is corrupted by at most e errors;
(ii) $\mathbf{w}_i$ can be reconstructed from any of its $(L_{\min}, L_{\text{over}}, e)$-erroneous traces.

To this end, our construction leverages the map $\mathcal{E}_{\mathtt{SD}}$ in Section III, which can encode WWL and SD sequences, as well as some coded indices $\mathbf{c}_i$'s. Roughly speaking, our coded indices are obtained by applying Theorem 17 with $L = I + r_I$ and $d = d_I$ so that the concatenation $\mathbf{c}_0 \circ \mathbf{c}_1 \circ \cdots \circ \mathbf{c}_{2^I-1}$ is an $(I + r_I, d_I)$-SD sequence. The parameters I, r_I and d_I are defined as follows:

$$\begin{aligned} I &\triangleq \left\lceil \frac{1-\gamma a}{1-\gamma} \log n + (\log n)^{0.5+\epsilon} \right\rceil, \\ r_I &\triangleq \lceil (3d_I + 8) \log I \rceil, \\ d_I &\triangleq 2e + 1, \end{aligned}$$

where $0 < \epsilon < 0.5$ is an arbitrary fixed number which is independent of n. In our construction, we require that some segments of each $\mathbf{w}_i$ are (K, d_I)-WWL, where $K \triangleq \lceil \sqrt{\log n} \rceil$. Then, we are able to use coded indices / robust positioning sequences to decode the index i from any length-$L_{\min}$ substring of $\mathbf{w}_i$ or $\mathbf{w}_i \circ \mathbf{w}_{i+1}$, even if e errors occur. To reconstruct $\mathbf{w}_i$ from its any $(L_{\min}, L_{\text{over}}, e)$-erroneous trace, we use Proposition 21 and require that some subsequence $\mathbf{v}_i$ of $\mathbf{w}_i$ is substring distant, where the distance between any two substrings of $\mathbf{v}_i$ should be at least

$$d_S \triangleq 4e + 1.$$

The subsequence $\mathbf{v}_i$ is obtained by applying the map $\mathcal{E}_{\mathtt{SD}}$ and the sequence $\mathbf{w}_i$ is obtained by partitioning $\mathbf{c}_i$ into small segments and interspersing them among $\mathbf{v}_i$.

Construction A (Index Construction): Let I, r_I and d_I be defined as above. Since e, a, γ, ϵ are constants, and $n \to \infty$, we have

$$\begin{aligned}&(I + r_I) - (3d_I + 7.5)\log(I + r_I) - O(1)\\ &= I + 0.5\log I - O(1) > I.\end{aligned}$$

Applying Theorem 17 with $L = I + r_I$ and $d = d_I$, there is an explicit construction of sequences $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{2^I-1} \in \Sigma^{I+r_I}$, which are actually $0^K \circ \mathbf{u} \circ \mathbf{s}_i$ in Theorem 17, such that the concatenation

$$\mathbf{c} \triangleq \mathbf{c}_0 \circ \mathbf{c}_1 \circ \cdots \circ \mathbf{c}_{2^I-1}$$

is an $(I + r_I, d_I)$-SD sequence. Moreover, according to the remark following Theorem 17, each $\mathbf{c}_i$ is $(3\lceil \frac{3}{2}\log(I + r_I)\rceil + \ell_{d_I}, d_I)$-WWL where

$$\ell_{d_I} \triangleq d_I\lceil \log d_I\rceil + 2d_I$$

is the length of the d_I-auto-cyclic sequence $\mathbf{u}$. Denote

$$K \triangleq \left\lceil \sqrt{\log n}\right\rceil,$$
$$F \triangleq \left\lceil \frac{I + r_I}{K}\right\rceil.$$

For each $i \in [2^I]$, we partition the sequence $\mathbf{c}_i$ into segments $\mathbf{c}_i^{(0)}, \mathbf{c}_i^{(1)}, \ldots, \mathbf{c}_i^{(F-1)}$, each of length $\lceil \frac{I+r_I}{F}\rceil$ or $\lfloor \frac{I+r_I}{F}\rfloor$. ■

In the following, we first consider the case of $L_{\min} \mid n$ and give the code construction. Then we will show how to modify this construction to settle the other cases.

A. The Case of $L_{\min} \mid n$

Let us define

$$r \triangleq I + r_I + K + \ell_{d_I} + d_I,$$

$$L \triangleq \left\lceil \left(L_{\text{over}} - K - \ell_{d_I} - d_I - 2\left\lceil \frac{I + r_I}{F}\right\rceil\right)\frac{L_{\min} - r}{L_{\min} - r + I + r_I}\right\rceil.$$

We note that by our choice of parameters, $L_{\min} > r$ for all sufficiently large n. Assume that $L_{\min} \mid n$ and denote $n_L \triangleq \frac{n}{L_{\min}}$. For each $i \in [2^I]$, let

$$N_i \triangleq \begin{cases}\lceil n_L/2^I\rceil(L_{\min} - r) & \text{if } i < n_L \bmod 2^I,\\ \lfloor n_L/2^I\rfloor(L_{\min} - r) & \text{otherwise.}\end{cases} \tag{6}$$

Then $\sum_{i\in[2^I]} N_i = n_L(L_{\min} - r)$.

Lemma 23: Let K, L, N_i be defined as above, and assume n is large enough. Then for each $i \in [2^I]$ there is an integer $m(N_i)$ with $N_i - m(N_i) = \Theta(N_i/\log N_i)$ and an invertible map $\mathcal{E}_{\mathsf{SD}}^{(i)} : \Sigma^{m(N_i)} \to \Sigma^{N_i}$ which can encode sequences of $\Sigma^{m(N_i)}$ into $(\lfloor K/4\rfloor, d_S)$-WWL and (L, d_S)-SD sequences.

Proof: We shall apply Theorem 20 to prove this lemma. To this end, we first need to verify that N_i can be arbitrarily large. As noted before, $L_{\min} - r > 0$. Additionally, $n_L = \Theta(n/\log n)$, and $2^I = n^{\frac{1-\gamma a}{1-\gamma}(1+o(1))}$ and by our choice of parameters, $\frac{1-\gamma a}{1-\gamma} < 1$ is a constant. Hence, $N_i \to \infty$ as $n \to \infty$.

Next, we need to verify that $\lfloor K/4\rfloor$ and L satisfy the two conditions in Theorem 20. Regarding the value of K, we need to show that $\lfloor K/4\rfloor \geqslant (2d_S+9)\log\log N_i + O(1)$. Noting that $r_I = \lceil (3d_I + 8)\log I\rceil = O(\log\log n)$ and $K = O(\sqrt{\log n})$, we have that

$$\begin{aligned}&1 - \frac{r}{L_{\min}}\\ &= 1 - \frac{I + r_I + K + \ell_{d_I} + d_I}{L_{\min}}\\ &= 1 - \frac{(\frac{1-\gamma a}{1-\gamma})\log n + (\log n)^{0.5+\epsilon} + O(\sqrt{\log n})}{a\log n + O(1)}\\ &= 1 - \left(\frac{1/a - \gamma}{1-\gamma} + \frac{1}{a(\log n)^{0.5-\epsilon}} + O\left(\frac{1}{\sqrt{\log n}}\right)\right)\\ &\qquad \times \frac{a\log n}{a\log n + O(1)}\\ &= 1 - \left(\frac{1/a - \gamma}{1-\gamma} + \frac{1}{a(\log n)^{0.5-\epsilon}} + O\left(\frac{1}{\sqrt{\log n}}\right)\right)\\ &\qquad \times \left(1 - O\left(\frac{1}{\log n}\right)\right)\\ &= \frac{1 - 1/a}{1-\gamma} - \frac{1}{a(\log n)^{0.5-\epsilon}} - O\left(\frac{1}{\sqrt{\log n}}\right).\end{aligned}$$

It follows that

$$\begin{aligned}\log N_i &= \log\left(\frac{n_L}{2^I}(L_{\min} - r)\right) \pm O(1)\\ &= \log\left(\frac{n}{2^I}\left(1 - \frac{r}{L_{\min}}\right)\right) \pm O(1)\\ &= \log n - I \pm O(1)\\ &= \frac{\gamma a - \gamma}{1-\gamma}\log n - (\log n)^{0.5+\epsilon} \pm O(1).\end{aligned}$$

Since $K = \lceil \sqrt{\log n}\rceil$, we have that $\lfloor K/4\rfloor$ is substantially larger than $(2d_S + 9)\log\log N_i + O(1)$.

Now, we verify the condition on L, namely that $L \geqslant \log N_i + (6d_S + 7)\log\log N_i + O(1)$. Note that

$$\begin{aligned}\frac{I + r_I}{L_{\min} - r} &= \frac{I + O(\log\log n)}{L_{\min} - I - O(\sqrt{\log n})}\\ &= \frac{I}{L_{\min} - I}\cdot\frac{1 + O(\log\log n/\log n)}{1 - O(1/\sqrt{\log n})}\\ &= \frac{I}{L_{\min} - I}\left(1 + O\left(\frac{1}{\sqrt{\log n}}\right)\right)\\ &= \frac{I}{L_{\min} - I} + O\left(\frac{1}{\sqrt{\log n}}\right),\end{aligned}$$

and

$$\left\lceil \frac{I + r_I}{F}\right\rceil = \left\lceil \frac{I + r_I}{\lceil (I + r_I)/K\rceil}\right\rceil \leqslant \frac{I + r_I}{(I + r_I)/K} + 1 = K + 1.$$

Hence, we have that

$$\begin{aligned}L &\geqslant \left(L_{\text{over}} - K - \ell_{d_I} - d_I - 2\left\lceil \frac{I + r_I}{F}\right\rceil\right)\\ &\qquad \cdot \frac{L_{\min} - r}{L_{\min} - r + I + r_I}\\ &\geqslant \frac{L_{\text{over}} - 3K - \ell_{d_I} - d_I - 2}{1 + (I + r_I)/(L_{\min} - r)}\end{aligned}$$

$$
\begin{aligned}
&= \frac{L_{\text{over}} - O(\sqrt{\log n})}{\frac{L_{\min}}{L_{\min}-I} + O\big(1/\sqrt{\log n}\big)} \\
&= \frac{L_{\text{over}}(L_{\min}-I)}{L_{\min}} \cdot \frac{1 - O(1/\sqrt{\log n})}{1 + O(1/\sqrt{\log n})} \\
&\geqslant \gamma\left(a \log n - \frac{1-\gamma a}{1-\gamma}\log n - (\log n)^{0.5+\epsilon} - 1\right) \\
&\quad \cdot \left(1 - O\left(\frac{1}{\sqrt{\log n}}\right)\right) \\
&= \frac{\gamma a - \gamma}{1-\gamma}\log n - \gamma(\log n)^{0.5+\epsilon} - O(\sqrt{\log n}).
\end{aligned}
$$

It follows that

$$
\begin{aligned}
L - \log N_i &= (1-\gamma)(\log n)^{0.5+\epsilon} - O(\sqrt{\log n}) \\
&= \omega(\log\log N_i).
\end{aligned}
$$

We can conclude that L is substantially larger than $\log N_i + (6d_S+7)\log\log N_i + O(1)$. ∎

Now, we present our code construction.

Construction B: Let $m(N_i)$'s be defined as in Lemma 23. We now describe a mapping from $\Sigma^{\sum_{i\in[2^I]} m(N_i)}$ to Σ^n. For any message $\mathbf{m} \in \Sigma^{\sum_{i\in[2^I]} m(N_i)}$, partition $\mathbf{m}$ into 2^I substrings:

$$\mathbf{m} = \mathbf{m}_0 \circ \mathbf{m}_1 \circ \cdots \circ \mathbf{m}_{2^I-1},$$

where each $\mathbf{m}_i$ has length $m(N_i)$. For each $i \in [2^I]$, let

$$\mathbf{v}_i = \mathcal{E}^{(i)}_{\mathtt{SD}}(\mathbf{m}_i) \in \Sigma^{N_i},$$

where $\mathcal{E}^{(i)}_{\mathtt{SD}}$ is the map mentioned in Lemma 23. We partition each $\mathbf{v}_i$ into substrings of length $L_{\min}-r$:

$$\mathbf{v}_i = \begin{cases} \mathbf{v}_{i,0} \circ \mathbf{v}_{i,1} \circ \cdots \circ \mathbf{v}_{i,\lceil n_L/2^I\rceil - 1} & \text{if } i < n_L \bmod 2^I, \\ \mathbf{v}_{i,0} \circ \mathbf{v}_{i,1} \circ \cdots \circ \mathbf{v}_{i,\lfloor n_L/2^I\rfloor - 1} & \text{otherwise.} \end{cases}$$

Then the total number of $\mathbf{v}_{i,j}$'s is n_L. We further partition each $\mathbf{v}_{i,j}$ into F segments of lengths $\lceil (L_{\min}-r)/F\rceil$ or $\lfloor (L_{\min}-r)/F\rfloor$:

$$\mathbf{v}_{i,j} = \mathbf{v}^{(0)}_{i,j} \circ \mathbf{v}^{(1)}_{i,j} \circ \cdots \circ \mathbf{v}^{(F-1)}_{i,j}.$$

Recall $\mathbf{c}^{(m)}_i$ from the index construction, Construction A. Let

$$\mathbf{w}_{i,j} \triangleq \begin{cases} 0^{d_I} \circ \mathbf{v}^{(0)}_{i,j} \circ \mathbf{c}^{(0)}_i \circ \cdots \circ \mathbf{v}^{(F-1)}_{i,j} \circ \mathbf{c}^{(F-1)}_i & \text{if } j = 0, \\ 1^{d_I} \circ \mathbf{v}^{(0)}_{i,j} \circ \mathbf{c}^{(0)}_i \circ \cdots \circ \mathbf{v}^{(F-1)}_{i,j} \circ \mathbf{c}^{(F-1)}_i & \text{otherwise.} \end{cases}$$

Finally, if $i < n_L \bmod 2^I$, let

$$\mathbf{w}_i = \mathbf{p} \circ \mathbf{w}_{i,0} \circ \mathbf{p} \circ \mathbf{w}_{i,1} \circ \cdots \circ \mathbf{p} \circ \mathbf{w}_{i,\lceil n_L/2^I\rceil - 1},$$

otherwise, let

$$\mathbf{w}_i = \mathbf{p} \circ \mathbf{w}_{i,0} \circ \mathbf{p} \circ \mathbf{w}_{i,1} \circ \cdots \circ \mathbf{p} \circ \mathbf{w}_{i,\lfloor n_L/2^I\rfloor - 1},$$

where $\mathbf{p} \triangleq 0^K \circ \mathbf{u}$ and $\mathbf{u}$ is the d_I-auto-cyclic sequence in Theorem 14. Denote

$$\mathbf{w} \triangleq \mathbf{w}_0 \circ \mathbf{w}_1 \circ \cdots \circ \mathbf{w}_{2^I-1}.$$

The constructed code, $\mathcal{C}_{\text{Trace}}$, is the image of the mapping described above. ∎

Lemma 24: Let $\mathcal{C}_{\text{Trace}}$ be the code obtained by Construction B. Then $\mathcal{C}_{\text{Trace}} \subseteq \Sigma^n$ and its rate is

$$R(\mathcal{C}_{\text{Trace}}) = \frac{1-1/a}{1-\gamma} - \frac{1}{a(\log n)^{0.5-\epsilon}} - O\left(\frac{1}{\sqrt{\log n}}\right).$$

Proof: In our construction, every sequence $\mathbf{w}_{i,j}$ has length $L_{\min} - r + d_I + |\mathbf{c}_i| = L_{\min} - K - \ell_{d_I}$, and so, the concatenation $\mathbf{p} \circ \mathbf{w}_{i,j}$ has length $L_{\min}$. It follows that the codeword $\mathbf{w}$ has length $n_L L_{\min} = n$. Noting that the map $\mathcal{E}_{\mathtt{SD}}$ is invertible, we can uniquely recover $\mathbf{m}$ from $\mathbf{w}$. Therefore, the code $\mathcal{C}_{\text{Trace}}$ has rate $\sum_{i\in[2^I]} m(N_i)/n$.

We have shown in the proof of Lemma 23 that

$$1 - \frac{r}{L_{\min}} = \frac{1-1/a}{1-\gamma} - \frac{1}{a(\log n)^{0.5-\epsilon}} - O\left(\frac{1}{\sqrt{\log n}}\right),$$

and for each $i \in [2^I]$,

$$\log N_i = \Theta(\log n).$$

Hence,

$$
\begin{aligned}
R(\mathcal{C}_{\text{Trace}}) &= \frac{\sum_{i\in[2^I]} m(N_i)}{n} = \frac{\sum_{i\in[2^I]} N_i - \Theta(N_i/\log N_i)}{n} \\
&= \frac{\sum_{i\in[2^I]} N_i}{n}\left(1 - \Theta\left(\frac{1}{\log n}\right)\right) \\
&= \frac{n_L(L_{\min}-r)}{n}\left(1 - \Theta\left(\frac{1}{\log n}\right)\right) \\
&= \left(1 - \frac{r}{L_{\min}}\right)\left(1 - \Theta\left(\frac{1}{\log n}\right)\right) \\
&= \frac{1-1/a}{1-\gamma} - \frac{1}{a(\log n)^{0.5-\epsilon}} - O\left(\frac{1}{\sqrt{\log n}}\right).
\end{aligned}
$$

∎

In the following, we shall show that the code $\mathcal{C}_{\text{Trace}}$ is an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code.

Lemma 25 (Construction 3 and Lemma 3.6 in [6]): Let $\mathbf{w} = \mathbf{p} \circ \mathbf{w}_{0,0} \circ \mathbf{p} \circ \mathbf{w}_{0,1} \circ \cdots \circ \mathbf{p} \circ \mathbf{w}_{2^I-1,\lfloor n_L/2^I\rfloor - 1}$ be a codeword of $\mathcal{C}_{\text{Trace}}$. Assume that the substrings $\mathbf{w}_{i,j}$'s satisfy the following conditions:

- (P1) $\mathbf{w}_{i,j}$ is a (K, d_I)-WWL sequence for each (i,j); and
- (P2) $\mathbf{w}_{i,j}[0, \mu-1] \circ \mathbf{w}_{i',j'}[\mu, L_{\min}-K-\ell_{d_I}-1]$ is a (K, d_I)-WWL sequence for $(i,j), (i',j')$ such that $(i,j) \neq (i',j')$ and $\mu \in [L_{\min} - K - \ell_{d_I}]$.

Then for every substring $\mathbf{y} = \mathbf{w}_{i_0+[L_{\min}]}$ in $\mathbf{w}$ and each[8] $i \in [L_{\min}]$, the following hold:

- (i) If $i + i_0 \equiv 0 \pmod{L_{\min}}$, then $\mathbf{y}_{i+[K+\ell_{d_I}]} = \mathbf{p}$.
- (ii) If $i + i_0 \not\equiv 0 \pmod{L_{\min}}$, then $d_H(\mathbf{y}_{i+[K+\ell_{d_I}]}, \mathbf{p}) \geqslant d_I$.

Lemma 26: Assume n is sufficiently large. Let $\mathbf{y}$ be an arbitrary length-$L_{\min}$ substring of $\mathbf{w} \in \mathcal{C}_{\text{Trace}}$. Then $\mathbf{y}$ contains a length-$(I + r_I - \mu)$ suffix of a coded index $\mathbf{c}_i$ and a length-μ prefix of either $\mathbf{c}_i$ or $\mathbf{c}_{i+1}$ for some $i \in [2^I]$ and $\mu \in [I + r_I]$. Furthermore, even if $\mathbf{y}$ is corrupted by at most e errors, we can still identify the positions where the said suffix and prefix appear, and so reconstruct them with at most e errors.

[8] If $i \in [L_{\min} - K + \ell_{d_I}, L_{\min} - 1]$, we let $\mathbf{y}_{i+[K+\ell_{d_I}]}$ denote the concatenation $\mathbf{y}[i, L_{\min}-1] \circ \mathbf{y}[0, K + \ell_{d_I} - (L_{\min} - i) - 1]$.

Proof: We note that the length of $\mathbf{p}\circ\mathbf{w}_{i,j}$ is $L_{\min}$, and that $\mathbf{w}$ is a concatenation of such strings. Hence, the first statement follows directly from the code construction. Now, assume that $\mathbf{y}$ is corrupted by at most e errors. We shall use Lemma 25 to identify the location of the marker $\mathbf{p}$ in $\mathbf{y}$. Recall that every $\mathbf{c}_i$ is $(3\lceil\frac{3}{2}\log(I+r_I)\rceil+\ell_{d_I},d_I)$-WWL (see the index construction, Construction A) and every $\mathbf{v}_i$ is $(\lfloor K/4\rfloor,d_S)$-WWL (see Lemma 23). Since $3\lceil\frac{3}{2}\log(I+r_I)\rceil+\ell_{d_I}<\lfloor K/4\rfloor$ and $d_I<d_S$, all the segments $\mathbf{c}_i^{(h)}$'s and $\mathbf{v}_{i,j}^{(h)}$'s are $(\lfloor K/4\rfloor,d_I)$-WWL. Note that the length of each $\mathbf{c}_i^{(h)}$ is at least $\lfloor\frac{I+r_I}{F}\rfloor=\lfloor\frac{I+r_I}{\lceil\frac{I+r_I}{K}\rceil}\rfloor\geqslant\frac{K}{2}$. Hence, $\mathbf{w}_{i,j}$'s satisfy the conditions in Lemma 25. This follows since any substring of length K contains a substring of length $\lfloor K/4\rfloor$ that is fully contained within a segment of the form $\mathbf{c}_i^{(h)}$ or $\mathbf{v}_{i,j}^{(h)}$, thus providing the minimum weight of d_I as claimed.

Since $\mathbf{y}$ suffers from at most e errors and $d_I=2e+1$, by Lemma 25 there is a unique index $i\in[L_{\min}]$ such that

$$d_H(\mathbf{y}_{i+[K+\ell_{d_I}]},\mathbf{p})\leqslant e.$$

Hence, by comparing the distance between the marker $\mathbf{p}$ and each length-$(K+\ell_{d_I})$ substring of $\mathbf{y}$, we can identify the location of the marker in $\mathbf{y}$. Once the marker $\mathbf{p}$ is located, the positions in which the symbols of the coded indices $\mathbf{c}_i^{(h)}$'s appear can also be determined. Then we can reconstruct a prefix $\mathbf{c}_i[\mu,I+r_I-1]$ and a suffix $\mathbf{c}_i[0,\mu-1]$ or $\mathbf{c}_{i+1}[\mu-1]$ for some $\mu\in[I+r_I]$ with at most e errors. ∎

The following lemma ensures that every length-L_{over} substring of $\mathbf{w}$ contains a long-enough substring of the (L,d_S)-SD sequence $\mathbf{v}_i$.

Lemma 27: Assume n is sufficiently large. Let $\mathbf{w}$ be a codeword of $\mathcal{C}_{\text{Trace}}$. Then every length-$L_{\text{over}}$ substring of $\mathbf{w}$ contains at least L consecutive symbols of $\mathbf{v}=\mathbf{v}_0\circ\mathbf{v}_1\circ\cdots\circ\mathbf{v}_{2^I-1}$.

Proof: Note that the concatenation

$$\mathbf{v}_{i,j}^{(0)}\circ\mathbf{c}_i^{(0)}\circ\cdots\circ\mathbf{v}_{i,j}^{(F-1)}\circ\mathbf{c}_i^{(F-1)}$$

consists of $|\mathbf{v}_{i,j}|+|\mathbf{c}_i|=L_{\min}-r+I+r_I$ symbols, out of which $|\mathbf{v}_{i,j}|=L_{\min}-r$ symbols are from $\mathbf{v}$. Then according to the construction, every length-L_{over} substring of $\mathbf{w}$ contains at least

$$\left(L_{\text{over}}-(K+\ell_{d_I})-d_I-2\left\lceil\frac{I+r_I}{F}\right\rceil\right)\frac{L_{\min}-r}{L_{\min}-r+I+r_I}$$

consecutive symbols of $\mathbf{v}$, where $L_{\text{over}}-(K+\ell_{d_I})-d_I-2\lceil\frac{I+r_I}{F}\rceil$ accounts for the worst case where the substring both begins and ends with some segments of the coded indices (of length $\lceil\frac{I+r_I}{F}\rceil$ or $\lfloor\frac{I+r_I}{F}\rfloor$) and contains a copy of $\mathbf{p}\circ 0^{d_I}$ or $\mathbf{p}\circ 1^{d_I}$. ∎

Theorem 28: The code $\mathcal{C}_{\text{Trace}}$ obtained in Construction B is an $(L_{\min},L_{\text{over}},e)$-trace maximal reconstruction code of Σ^n with rate

$$R(\mathcal{C}_{\text{Trace}})=\frac{1-1/a}{1-\gamma}-\frac{1}{a(\log n)^{0.5-\epsilon}}-O\left(\frac{1}{\sqrt{\log n}}\right).$$

Proof: The code rate has been calculated in Lemma 24. Let $\mathbf{w}$ be a codeword of $\mathcal{C}_{\text{Trace}}$ and $\mathcal{Y}$ be an $(L_{\min},L_{\text{over}},e)$-erroneous trace of $\mathbf{w}$. For each $\mathbf{y}$ in $\mathcal{Y}$, since the length of $\mathbf{y}$ is at least $L_{\min}$, according to Lemma 26, we can extract a corrupted copy $\mathbf{c}_{\text{suf}}$ of the length-$(I+r_I-\mu)$ suffix of $\mathbf{c}_i$, and a corrupted copy $\mathbf{c}_{\text{pre}}$ of a length-μ prefix of either $\mathbf{c}_i$ or $\mathbf{c}_{i+1}$, with the total number of errors being no more than e. Consider the following cases.

1) If $\mu=0$, then $\mathbf{c}_{\text{suf}}$ is a corrupted copy of $\mathbf{c}_i$, and so, we can run the locating algorithm of the robust positioning sequence $\mathbf{c}=\mathbf{c}_0\circ\mathbf{c}_1\circ\cdots\circ\mathbf{c}_{2^I-1}$ on the corrupted $\mathbf{c}_{\text{suf}}$ to determine the index i.
2) If $\mu>0$ then $\mathbf{y}$ contains a copy of either $\mathbf{p}\circ 0^{d_I}$ or $\mathbf{p}\circ 1^{d_I}$ with at most e errors. Since $d_I=2e+1$, we can distinguish these two cases.
   a) If $\mathbf{y}$ contains a copy of $\mathbf{p}\circ 0^{d_I}$, then $\mathbf{c}_{\text{pre}}$ is a prefix of $\mathbf{c}_{i+1}$, and so, we run the locating algorithm of $\mathbf{c}$ on $\mathbf{c}_{\text{suf}}\circ\mathbf{c}_{\text{pre}}$ to decode the index i.
   b) If $\mathbf{y}$ contains a copy of $\mathbf{p}\circ 1^{d_I}$, then $\mathbf{c}_{\text{pre}}$ is a prefix of $\mathbf{c}_i$, and so, we run the locating algorithm of $\mathbf{c}$ on $\mathbf{c}_{\text{pre}}\circ\mathbf{c}_{\text{suf}}$ to decode the index i.

The discussion above shows that for every string $\mathbf{y}\in\mathcal{Y}$, we can decode the index i. If $\mathbf{y}$ intersects both $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$, then we can determine its location in $\mathbf{w}$ by identifying the location of the marker $\mathbf{p}$ in $\mathbf{y}$. For the other strings with index i, since $\mathbf{v}_i$ is an $(L,4e+1)$-SD sequence, according to Lemma 27 and Proposition 21, there is a unique way to determine the correct order of these strings and match correctly the suffix and the prefix of consecutive strings. By taking the majority value at every position, we can reconstruct a sequence $\mathbf{w}_i'$. Recalling that $\mathbf{w}_i'$ is reconstructed from the substrings of $\mathcal{Y}$ that have index i and do not intersect with $\mathbf{w}_{i-1}$ and $\mathbf{w}_{i+1}$, it constitutes a substring of $\mathbf{w}_i$, possibly with some errors, missing a prefix and a suffix of $\mathbf{w}_i$. It remains to determine the location of $\mathbf{w}_i'$ in $\mathbf{w}_i$, which can be done as follows.

1) If $\mathbf{w}_i'$ contains a corrupted copy of $\mathbf{p}\circ 0^{d_I}$ with at most e errors, then the location this marker in $\mathbf{w}_i'$ determines the location of $\mathbf{w}_i'$ in $\mathbf{w}_i$, since $\mathbf{w}_i$ only contains one copy of $\mathbf{p}\circ 0^{d_I}$.
2) If $\mathbf{w}_i'$ does not contain any corrupted copy of $\mathbf{p}\circ 0^{d_I}$ with up to e errors, then there is a string $\hat{\mathbf{y}}\in\mathcal{Y}$ which intersects both $\mathbf{w}_{i-1}$ and $\mathbf{w}_i$ and contains $\mathbf{p}\circ 0^{d_I}$ as a substring with at most e errors, since the length of $\mathbf{p}\circ 0^{d_I}$ is less that L_{over}.
   a) If $\hat{\mathbf{y}}$ overlaps $\mathbf{w}_i$ in at most L_{over} positions, since $L_{\text{over}}<L_{\min}$, $\mathbf{w}_i'$ must contain a copy of the first $\mathbf{p}\circ 1^{d_I}$ of $\mathbf{w}_i$, and so, the location of $\mathbf{w}_i'$ in $\mathbf{w}_i$ can be determined by identifying the first occurrence of the marker $\mathbf{p}$ in $\mathbf{w}_i'$.
   b) If $\hat{\mathbf{y}}$ overlaps $\mathbf{w}_i$ in at least L_{over} positions, then $\hat{\mathbf{y}}$ and the length-L_{over} prefix of $\mathbf{w}_i'$ share a length-L substring of $\mathbf{v}_i$. Since $\mathbf{v}_i$ is $(L,4e+1)$-SD, we can match the suffix of $\hat{\mathbf{y}}$ and the prefix of $\mathbf{w}_i'$ correctly. Then the location of $\mathbf{w}_i'$ in $\mathbf{w}$ can be deduced from the location of $\hat{\mathbf{y}}$ in $\mathbf{w}$.

∎

B. The Case of $L_{\min}\nmid n$

Now, we consider the case that $L_{\min}$ does not divide n. Take $n_L=\lfloor n/L_{\min}\rfloor$. Construction B can yield a trace maximal reconstruction code of block length $n_LL_{\min}$. Our approach

is to extend this code to have length n. Let N_i be defined as in (6) and $m(N_i)$ be defined as in Lemma 23. For any message $\mathbf{m} \in \Sigma^{\sum_{i\in[2^I]} m(N_i)}$, partition $\mathbf{m}$ into 2^I substrings, each of length $m(N_i)$:

$$\mathbf{m} = \mathbf{m}_0 \circ \mathbf{m}_1 \circ \cdots \circ \mathbf{m}_{2^I-1}.$$

For each $i \in [2^I - 1]$, let

$$\mathbf{v}_i = \mathcal{E}_{\mathtt{SD}}^{(i)}(\mathbf{m}_i) \in \Sigma^{N_i}.$$

The main difference from the previous case is the encoding of $\mathbf{m}_{2^I-1}$. We recall that the encoder $\mathcal{E}_{\mathtt{SD}}^{(i)}$ first encodes the message $\mathbf{m}_i$ to an SD and WWL sequence of length probably less than N_i. Then it extends the sequence by appending a sequence $\bar{\mathbf{s}}$ and taking the first N_i bits of the concatenation. For $i = 2^I - 1$, we modify the encoder $\mathcal{E}_{\mathtt{SD}}^{(2^I-1)}$ by taking the first $N_{2^I-1} + L_{\min} - r$ bits of the concatenation. Recalling that

$$\log(N_{2^I-1}) = \log n - I \pm O(1) = \Theta(\log n)$$

and $L_{\min} = \lceil a \log n \rceil$, this is possible since asymptotically the length of $\bar{\mathbf{s}}$ is larger than $N_{2^I-1}+L_{\min} - r$, see (5). We denote this modified encoder as $\mathcal{E}_{\mathtt{SDE}}^{(2^I-1)}$ and let

$$\mathbf{v}_{2^I-1} = \mathcal{E}_{\mathtt{SDE}}^{(2^I-1)}(\mathbf{m}_{2^I-1}).$$

Then $\mathbf{v}_{2^I-1}$ is $(\lfloor K/4 \rfloor, d_S)$-WWL and (L, d_S)-SD and has length $N_{2^I-1}+L_{\min}-r = (\lfloor n_L/2^I \rfloor+1)(L_{\min}-r)$. Moreover, the message $\mathbf{m}_{2^I-1}$ can be decoded from the first N_{2^I-1} bits of $\mathbf{v}_{2^I-1}$. In other words, the last $L_{\min}-r$ bits are redundant.

Then, we proceed similarly as in Construction B and obtain an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code of block length $(n_L + 1)L_{\min}$. Note that the last $L_{\min}$ bits are redundant, and so, we delete $(n_L + 1)L_{\min} - n$ of them to form an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code of length n, with code rate

$$\begin{aligned}\frac{\sum_{i\in[2^I]} m(N_i)}{n} &= \left(\frac{1-1/a}{1-\gamma} - o(1)\right)\frac{n_L L_{\min}}{n} \\ &= \frac{1-1/a}{1-\gamma} - o(1).\end{aligned}$$

C. Handling Noise Which Occurs Before Sequencing

Up to now, we have studied $(L_{\min}, L_{\text{over}}, e)$-trace reconstruction codes, which allow reconstructing the maximum reconstructible-string from an erroneous trace $\mathcal{Y}$ of a codeword $\mathbf{w}$. We use $M(\mathcal{Y})$ to denote the maximum reconstructible-string of $\mathcal{Y}$. If $\mathcal{Y}$ is reliable, then $M(\mathcal{Y}) = \mathbf{w}$. However, if $\mathcal{Y}$ is not reliable, then $M(\mathcal{Y})$ is different from $\mathbf{w}$. This may happen especially when some symbols are covered by a small number of substrings or when the sequence $\mathbf{w}$ is subject to errors before its substrings are sampled. In the remainder of this section, we modify Construction B to combat such errors. Our construction, which is presented below, borrows the idea from [2, Construction B].

Construction C: Assume that $L_{\min} \mid n$ and take $n_L = n/L_{\min}$. Let $N \triangleq \lfloor n_L/2^I \rfloor (L_{\min} - r)$. According to Lemma 23, there is an integer $m(N)$ with $N - m(N) = \Theta(N/\log N)$ and an invertible map $\mathcal{E}_{\mathtt{SD}} : \Sigma^{m(N)} \to \Sigma^N$ which can encode sequences of $\Sigma^{m(N)}$ into $(\lfloor K/4 \rfloor, d_S)$-WWL and (L, d_S)-SD sequences. Let $\mathcal{E}_{\mathtt{SDE}} : \Sigma^{m(N)} \to \Sigma^{N+L_{\min}-r}$ be an encoder which modifies[9] $\mathcal{E}_{\mathtt{SD}}$ by taking the first $N + L_{\min} - r$ bits of the concatenation.

For any message $\mathbf{m} \in \Sigma^{(2^I-2\tau)m(N)}$, we first use a $[2^I, 2^I - 2\tau, 2\tau + 1]_{2^{m(N)}}$ Reed-Solomon code[10] to encode $\mathbf{m}$ into a codeword $\bar{\mathbf{m}} \in \Sigma^{2^I m(N)}$. We partition $\bar{\mathbf{m}}$ into sequences of length $L_{\min} - r$:

$$\bar{\mathbf{m}} = \bar{\mathbf{m}}_0 \circ \bar{\mathbf{m}}_1 \circ \cdots \circ \bar{\mathbf{m}}_{2^I-1}.$$

For each $i \in [2^I]$, let

$$\mathbf{v}_i \triangleq \begin{cases} \mathcal{E}_{\mathtt{SDE}}(\bar{\mathbf{m}}_i) \in \Sigma^{N+L_{\min}-r} & \text{if } i < n_L \bmod 2^I, \\ \mathcal{E}_{\mathtt{SD}}(\bar{\mathbf{m}}_i) \in \Sigma^N & \text{otherwise.} \end{cases}$$

Then we proceed similarly as in Construction B to obtain a sequence $\mathbf{w}$ of length n. We use $\hat{\mathcal{C}}_{\text{Trace}}$ to denote the code produced by this construction. ■

Lemma 29: Let $\mathbf{w}$ be a codeword of $\hat{\mathcal{C}}_{\text{Trace}}$ and $\mathcal{Y}$ be an $(L_{\min}, L_{\text{over}}, e, \tau)$-erroneous trace of $\mathbf{w}$. Then we can recover $\mathbf{m}$ from $\mathcal{Y}$.

Proof: With the same argument as the proof of Theorem 28, we can show that $\hat{\mathcal{C}}_{\text{Trace}}$ is an $(L_{\min}, L_{\text{over}}, e)$-trace reconstruction code of Σ^n. Since $\mathcal{Y}$ is also an $(L_{\min}, L_{\text{over}}, e)$-erroneous trace of $\mathbf{w}$, the maximum reconstructible-substring $M(\mathcal{Y})$ can be decoded from $\mathcal{Y}$. By reversing the operations in Construction C, we obtain a sequence $\bar{\mathbf{m}}' \in \Sigma^{2^I m(N)}$ from $M(\mathcal{Y})$. We partition $\bar{\mathbf{m}}'$ into 2^I segments of the same length, i.e., $\bar{\mathbf{m}}' = \bar{\mathbf{m}}'_0 \circ \bar{\mathbf{m}}'_1 \circ \cdots \circ \bar{\mathbf{m}}'_{2^I-1}$. Since $d_H(M(\mathcal{Y}), \mathbf{w}) \leqslant \tau$, then there are at most τ indices $i \in [2^I]$ such that $\bar{\mathbf{m}}_i \neq \bar{\mathbf{m}}'_i$. Hence, we can run the decoder of the Reed-Solomon code on $\bar{\mathbf{m}}'$ to recover $\bar{\mathbf{m}}$. ■

Theorem 30: Suppose that $\tau = O\left(n^{\frac{1-\gamma a}{1-\gamma}}\right)$. Then the code $\hat{\mathcal{C}}_{\text{Trace}}$ obtained in Construction C is an $(L_{\min}, L_{\text{over}}, e, \tau)$-trace reconstruction code of Σ^n with rate

$$R(\hat{\mathcal{C}}_{\text{Trace}}) = \frac{1-1/a}{1-\gamma} - o(1).$$

Proof: Since $\tau = O\left(n^{\frac{1-\gamma a}{1-\gamma}}\right)$, we have $2\tau/2^I = o(1)$. Hence, the code rate

$$\begin{aligned} &R(\hat{\mathcal{C}}_{\text{Trace}}) \\ &= \frac{(2^I - 2\tau)m(N)}{n} \\ &= \frac{2^I m(N)}{n} - \frac{2\tau N}{n}\left(1 - \Theta\left(\frac{1}{\log N}\right)\right) \\ &\geqslant \frac{2^I m(N)}{n} - \frac{2\tau}{2^I}\left(1 - \frac{r}{L_{\min}}\right)\left(1 - \Theta\left(\frac{1}{\log N}\right)\right) \\ &= \frac{2^I m(N)}{n} - o(1). \end{aligned}$$

[9]This encoder closely resembles the encoder $\mathcal{E}_{\mathtt{SDE}}^{(2^I-1)}$ with the only difference being the message length.

[10]The Reed-Solomon code is over the finite field of size $2^{m(N)}$. The message is partitioned into groups of $m(N)$ bits, and each group is translated to a single symbol from the finite field. After encoding the reverse translation to bits is performed. Note that $m(N) = N - \Theta(N/\log N)$, $\log(N) = \Theta(\log n)$ and $I = O(\log n)$. Hence, $m(N) > I$ and so, the Reed-Solomon code exists.

Consider the N_i's which are defined in (6). We have that

$$N_i \triangleq \begin{cases} N + L_{\min} - r & \text{if } i < n_L \bmod 2^I, \\ N & \text{otherwise.} \end{cases}$$

Hence,

$$\begin{aligned} R(\hat{\mathcal{C}}_{\text{Trace}}) &= \frac{2^I m(N)}{n} - o(1) \\ &\geqslant \frac{\sum_{i\in[2^I]} m(N_i) - 2^I(L_{\min} - r)}{n} - o(1) \\ &= R(\mathcal{C}_{\text{Trace}}) - o(1) = \frac{1 - 1/a}{1-\gamma} - o(1). \end{aligned}$$

■

D. $(L_{\min}, 0, e)$-Trace Maximal Reconstruction Codes

In this subsection, we consider the case of $L_{\text{over}} = 0$.

Construction D: Suppose that $L_{\min} = \lceil a \log n \rceil$, $L_{\text{over}} = 0$ and $L_{\min} \mid n$. As before, we denote $n_L \triangleq \frac{n}{L_{\min}}$ and $K \triangleq \lceil \sqrt{\log n} \rceil$. However, this time, we let $I \triangleq \lceil \log n_L \rceil$ and $r_I \triangleq \lceil (3d+8) \log I \rceil$ where $d = 2e+1$ and $\ell = d\lceil \log d \rceil + 2d$. Then according to Theorem 17, there is a collection of $(3\lceil \frac{3}{2} \log(I + r_I) \rceil + \ell, d)$-WWL sequences $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{2^I-1} \in \Sigma^{I+r_I}$ such that the concatenation $\mathbf{c}_0 \circ \mathbf{c}_1 \circ \cdots \circ \mathbf{c}_{2^I-1}$ is an $(I + r_I, d)$-SD sequence.

Denote $m' \triangleq L_{\min} - (I + r_I + K + \ell)$. Let $\mathcal{E}_{\texttt{WWL}}$ be the encoder in [14, Algorithm 2] which can encode sequences of $\Sigma^{m'-d}$ into $(\lceil K/4 \rceil, d)$-WWL sequences[11] of $\Sigma^{m'}$. For a message $\mathbf{m} = \mathbf{m}_0 \circ \mathbf{m}_1 \circ \cdots \circ \mathbf{m}_{n_L-1}$ where $\mathbf{m}_i \in \Sigma^{m'-d}$ for $i \in [n_L]$, let $\mathbf{w}_i \triangleq \mathcal{E}_{\texttt{WWL}}(\mathbf{m}_i)$ for all $i \in [n_L]$.

Denote $\mathbf{p} \triangleq 0^K \circ \mathbf{u}$ where $\mathbf{u}$ is a d-auto-cyclic sequence of length ℓ. Let

$$\mathbf{w} = \mathbf{p} \circ \mathbf{c}_0 \circ \mathbf{w}_0 \circ \mathbf{p} \circ \mathbf{c}_1 \circ \mathbf{w}_1 \circ \cdots \circ \mathbf{p} \circ \mathbf{c}_{n_L-1} \circ \mathbf{w}_{n_L-1}.$$

Output $\mathbf{w}$ as the codeword which encodes the message $\mathbf{m}$. The image under this mapping is the code that we construct. ■

Theorem 31: The code obtained in Construction D is an $(L_{\min}, 0, e)$-trace maximal reconstruction code of Σ^n with rate

$$1 - \frac{1}{a} - O\left(\frac{1}{\sqrt{\log n}}\right).$$

Sketch of Proof: The code has rate

$$\begin{aligned} \frac{n_L(m'-d)}{n} &= \frac{m'-d}{L_{\min}} = \frac{L_{\min} - (I + r_I + K + \ell + d)}{L_{\min}} \\ &= 1 - \frac{1}{a} - O\left(\frac{1}{\sqrt{\log n}}\right). \end{aligned}$$

Now, let $\mathbf{y}$ be a length-$L_{\min}$ substring of some codeword $\mathbf{w}$. Then $\mathbf{y}$ must contain either a copy of $\mathbf{p} \circ \mathbf{c}_i$ or a suffix of $\mathbf{p} \circ \mathbf{c}_i$ together with a prefix of $\mathbf{p} \circ \mathbf{c}_{i+1}$. Note that $\mathbf{w}_i$'s and $\mathbf{c}_j$'s are $(K/4, d)$-WWL sequences and each has length $\Theta(\log n)$. Since $K = \lceil \sqrt{\log n} \rceil$, it can be checked that the concatenations $\mathbf{c}_i \circ \mathbf{w}_i$'s satisfy the conditions in Lemma 25. Thus, even if $\mathbf{y}$ suffers from e errors, we can still locate the marker $\mathbf{p}$ in $\mathbf{y}$. Then we can run the locating algorithm of the robust positioning sequence $\mathbf{c}_0 \circ \mathbf{c}_1 \circ \cdots \circ \mathbf{c}_{2^I-1}$ to determine the index i or $i+1$, and hence the location of $\mathbf{y}$. ■

For the case of $L_{\min} \nmid n$, let $n_L = \lceil n/L_{\min} \rceil$. We first construct an $(L_{\min}, 0, e)$-trace maximal reconstruction code of $\Sigma^{n_L L_{\min}}$, where the length-$L_{\min}$ suffix of every codeword is fixed. This can be achieved by using Construction D with messages $\mathbf{m} = \mathbf{m}_0 \circ \mathbf{m}_1 \circ \cdots \circ \mathbf{m}_{n_L-2} \circ 0^{m'-d}$, where $\mathbf{m}_i \in \Sigma^{m'-d}$ for $i \in [n_L - 1]$. Then we truncate it to be of length n. In this way, we get a code of rate

$$\begin{aligned} &\frac{\lfloor n/L_{\min} \rfloor (L_{\min} - (I + r_I + K + \ell + d))}{n} \\ &\geqslant \left(1 - \frac{L_{\min} - 1}{n}\right)\left(1 - \frac{I + r_I + K + \ell + d}{L_{\min}}\right) \\ &= 1 - \frac{1}{a} - O\left(\frac{1}{\sqrt{\log n}}\right). \end{aligned}$$

For $(L_{\min}, 0, e, \tau)$-erroneous trace reconstruction, we proceed similarly as in [2, Construction B]. We first use an $(n_L, 2^{(m'-d)(n_L-r)}, 2\tau+1)_{2^{m'-d}}$ code to encode a message $\mathbf{m} = \mathbf{m}_0 \circ \mathbf{m}_1 \circ \cdots \circ \mathbf{m}_{n_L-r-1} \in \Sigma^{(m'-d)(n_L-r)}$ to a sequence $\bar{\mathbf{m}} = \bar{\mathbf{m}}_0 \circ \bar{\mathbf{m}}_1 \circ \cdots \circ \bar{\mathbf{m}}_{n_L-1} \in \Sigma^{(m'-d)n_L}$. Then we use the encoder outlined in Construction D to get a codeword $\mathbf{w}$. We note that Construction B in [2] only concerns errors before sequencing, while our construction incorporates errors both before and after sequencing.

V. Multi-Strand Reconstruction

In this section, instead of reconstructing a single sequence, we consider the problem of reconstructing a *multiset* of k sequences of length n from the union of their traces. The following construction of multi-strand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction codes is adapted from [25, Construction C].

Construction E: Let $N \triangleq k(n - L_{\text{over}}) + L_{\text{over}}$. We take an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code $\mathcal{C}$ of Σ^N. For each codeword $\mathbf{x} \in \mathcal{C}$, let

$$\begin{aligned} \mathcal{S}(\mathbf{x}) \triangleq \{&\mathbf{x}_{0+[n]}, \mathbf{x}_{n-L_{\text{over}}+[n]}, \mathbf{x}_{2(n-L_{\text{over}})+[n]}, \ldots, \\ &\mathbf{x}_{(k-1)(n-L_{\text{over}})+[n]}\} \in \mathcal{X}_{n,k}. \end{aligned}$$

The code we construct is $\mathcal{D}$, defined as,

$$\mathcal{D} \triangleq \{\mathcal{S}(\mathbf{x}) \ : \ \mathbf{x} \in \mathcal{C}\} \subseteq \mathcal{X}_{n,k}.$$

■

Lemma 32: Let $L_{\min} > L_{\text{over}}$. Then the code $\mathcal{D}$ from Construction E is a multi-strand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code of $\mathcal{X}_{n,k}$.

Proof: It is easy to see that an $(L_{\min}, L_{\text{over}}, e)$-erroneous trace $\mathcal{Y}$ of $\mathcal{S}(\mathbf{x})$ is also an $(L_{\min}, L_{\text{over}}, e)$-erroneous trace of $\mathbf{x}$. Since $\mathcal{C}$ is a trace maximal reconstruction code, then for each $\mathbf{y} \in \mathcal{Y}$, we can determine its location in $\mathbf{x}$. Hence, we can determine the index i such that $\mathbf{y} \in \mathcal{Y}_i$ and determine the location of $\mathbf{y}$ in $\mathbf{x}_i$. ■

Lemma 33 ([25, Lemma 16]): $\log|\mathcal{X}_{n,k}| = k(n - \log(k/\mathtt{e})) + o(k)$[12].

[11] Note that $m' = \Theta(\log n)$ and $K = \lceil \sqrt{\log n} \rceil$. Hence, $K/4 \gg \mathcal{F}(m', d) = \log m' + (d-1)\log\log m' + O(1)$. Then according to Lemma 35 in [14], the encoder $\mathcal{E}_{\texttt{WWL}}$ does work.

[12] We use $\mathtt{e}$ to denote $\exp(1)$ in order to avoid confusion with e which denotes the number of errors.

Theorem 34: Suppose that $\limsup_{n\to\infty} \log k/n < 1$, $L_{\text{over}} = \lceil \log(nk) \rceil + (24e+13)\lceil \log\lceil \log(nk) \rceil \rceil + (4e+1)\lceil \log(4e+1) \rceil + 20e + 5$ and $L_{\min} > L_{\text{over}}$. For sufficiently large n, there is a multi-strand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code of $\mathcal{X}_{n,k}$ whose rate is $1 - o(1)$.

Proof: Let $N = k(n - L_{\text{over}}) + L_{\text{over}}$. Then $L_{\text{over}} \geqslant \lceil \log N \rceil + (24e+13)\lceil \log\lceil \log N \rceil \rceil + (4e+1)\lceil \log(4e+1) \rceil + 20e + 5$. According to Corollary 22, there is an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code $\mathcal{C}$ of Σ^n whose rate is $1 - o(1)$. Applying Construction E with this code, we obtain a multi-strand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code $\mathcal{D}$ of $\mathcal{X}_{n,k}$ with $|\mathcal{D}| = |\mathcal{C}|$. Note that

$$\begin{aligned}\frac{N}{\log|\mathcal{X}_{n,k}|} &= \frac{k(n - L_{\text{over}}) + L_{\text{over}}}{k(n - \log(k/\mathsf{e})) + o(k)} \\ &= \frac{n - L_{\text{over}} + L_{\text{over}}/k}{n - \log k + O(1)} \\ &= 1 - \frac{L_{\text{over}} - \log k - L_{\text{over}}/k + O(1)}{n - \log k + O(1)} \\ &= 1 - O\left(\frac{\log n}{n}\right).\end{aligned}$$

Hence, the code rate is

$$\begin{aligned}R(\mathcal{D}) = \frac{\log|\mathcal{D}|}{\log|\mathcal{X}_{n,k}|} &= \frac{\log|\mathcal{C}|}{N}\frac{N}{\log|\mathcal{X}_{n,k}|} \\ &= (1 - o(1))\left(1 - O\left(\frac{\log n}{n}\right)\right) \\ &= 1 - o(1).\end{aligned}$$

■

Now, we consider the case of $L_{\text{over}} \leqslant \log(nk)$. When $(L_{\min}, L_{\text{over}}) = (\ell, \ell - 1)$, it has been shown in [25, Corollary 17] that if $\ell = \log(nk) - \omega(1)$, then the code rate of any multi-strand $(L_{\min}, L_{\text{over}})$-trace reconstruction code is $o(1)$.

In the following, we assume that $L_{\min} = a\log(nk)$ and $L_{\text{over}} = \gamma L_{\min}$ where $a > 0$ and $0 \leqslant a\gamma \leqslant 1$. Let

$$L^* \triangleq (n - L_{\text{over}}) \bmod (L_{\min} - L_{\text{over}}).$$

We first present some upper bounds on the rate of multi-strand $(L_{\min}, L_{\text{over}})$-trace reconstruction codes with $L_{\text{over}} \leqslant \log(nk)$: Lemma 36 and Lemma 39 address the cases of $L_{\min} = \lceil a\log(nk) \rceil$ with $a > 1$ and $a < 1$, respectively, while Corollary 37 and Lemma 38 handle the case of $L_{\min} = \log(nk) + o(\log(nk))$.

Lemma 35 ([25, In the proof of Lemma 8]): For all $v \geqslant u \geqslant 0$, $\log \binom{u+v}{u} < u(2\log \mathsf{e} + \log v - \log u)$.

Lemma 36: Suppose that $L_{\min} = \lceil a\log(nk) \rceil$ and $L_{\text{over}} = \lceil \gamma L_{\min} \rceil$ where $a > 1$ and $0 \leqslant a\gamma \leqslant 1$. Let $\mathcal{C}$ be a multi-strand $(L_{\min}, L_{\text{over}})$-trace reconstruction code of $\mathcal{X}_{n,k}$. Then it holds that

$$\frac{\log|\mathcal{C}|}{nk} \leqslant \left(\frac{1 - 1/a}{1 - \gamma}\right)\left(1 - \gamma\frac{L_{\min}}{n}\right) + \frac{1/a - \gamma}{1 - \gamma} \cdot \frac{L^*}{n} + O\left(\frac{\log n}{n}\right).$$

In particular, if $\log k = o(n)$, then the code rate satisfies

$$R(\mathcal{C}) \leqslant \frac{1 - 1/a}{1 - \gamma} + o(1),$$

and if $\log k = \kappa n + o(n)$ where $0 < \kappa < 1$ is a real constant, then the code rate satisfies

$$R(\mathcal{C}) \leqslant \frac{1 - a\gamma\kappa}{1 - \kappa}\left(\frac{1 - 1/a}{1 - \gamma}\right) + \frac{1/a - \gamma}{(1 - \gamma)(1 - \kappa)} \cdot \frac{L^*}{n} + o(1).$$

Proof: For a sequence $\mathbf{x} \in \Sigma^n$, let

$$\begin{aligned}\hat{\mathcal{Y}}(\mathbf{x}) \triangleq \Bigg\{ &\mathbf{x}_{i(L_{\min} - L_{\text{over}}) + [L_{\min}]} \ : \\ &i \in \left[\frac{n - L_{\text{over}} - L^*}{L_{\min} - L_{\text{over}}} - 1\right] \Bigg\} \\ &\cup \{\mathbf{x}[n - L_{\min} - L^*, n - 1]\}.\end{aligned}$$

For a codeword $\mathcal{S} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{k-1}\} \in \mathcal{C}$, let

$$\hat{\mathcal{Y}}(\mathcal{S}) \triangleq \bigcup_{i=0}^{k-1} \hat{\mathcal{Y}}(\mathbf{x}_i).$$

Then $\hat{\mathcal{Y}}(\mathcal{S})$ is an $(L_{\min}, L_{\text{over}})$-trace of $\mathcal{S}$.

Since $\mathcal{C}$ is an $(L_{\min}, L_{\text{over}})$-trace reconstruction code, necessarily $\hat{\mathcal{Y}}(\mathcal{S}) \neq \hat{\mathcal{Y}}(\mathcal{S}')$ for any two different codewords $\mathcal{S}$ and $\mathcal{S}'$. It follows that

$$|\mathcal{C}| \leqslant \left|\left\{\hat{\mathcal{Y}}(\mathcal{S}) \ : \ \mathcal{S} \in \mathcal{C}\right\}\right|.$$

Note that $\hat{\mathcal{Y}}(\mathcal{S})$ is a multiset consisting of $k\frac{n - L_{\min} - L^*}{L_{\min} - L_{\text{over}}}$ sequences of $\Sigma^{L_{\min}}$ and k sequences of $\Sigma^{L_{\min} + L^*}$. Hence,

$$|\mathcal{C}| \leqslant \binom{k\left(\frac{n - L_{\min} - L^*}{L_{\min} - L_{\text{over}}}\right) + 2^{L_{\min}} - 1}{2^{L_{\min}} - 1} \cdot \binom{k + 2^{L_{\min} + L^*} - 1}{2^{L_{\min} + L^*} - 1}. \tag{7}$$

We denote the first binomial coefficient in (7) as A and the second one as B. Since $2^{L_{\min}} \geqslant (nk)^a > \frac{k(n - L_{\min})}{L_{\min} - L_{\text{over}}}$ and $2^{L_{\min} + L^*} > k$, according to Lemma 35, we have that

$$\begin{aligned}&\frac{\log A}{nk} \\ &< \frac{k}{nk}\left(\frac{n - L_{\min} - L^*}{L_{\min} - L_{\text{over}}}\right) \\ &\quad \times \left(2\log \mathsf{e} + L_{\min} - \log\left(\frac{k(n - L_{\min} - L^*)}{L_{\min} - L_{\text{over}}}\right)\right) \\ &= \frac{1 - (L_{\min} + L^*)/n}{L_{\min} - L_{\text{over}}}(L_{\min} - \log(nk) + O(\log\log(nk))) \\ &= \left(1 - \frac{L_{\min} + L^*}{n}\right)\frac{L_{\min} - \log(nk)}{L_{\min} - L_{\text{over}}} + O\left(\frac{\log\log(nk)}{\log(nk)}\right) \\ &= \frac{1 - 1/a}{1 - \gamma}\left(1 - \frac{L_{\min} + L^*}{n}\right) + O\left(\frac{\log\log(nk)}{\log(nk)}\right),\end{aligned} \tag{8}$$

and

$$\begin{aligned}\frac{\log B}{nk} &< \frac{1}{n}(2\log \mathsf{e} + L_{\min} + L^* - \log k) \\ &= \frac{(1 - 1/a)L_{\min}}{n} + \frac{L^*}{n} + O\left(\frac{\log n}{n}\right).\end{aligned} \tag{9}$$

Combining (7), (8) and (9), we have that

$$\frac{\log|\mathcal{C}|}{nk} \leqslant \left(\frac{1 - 1/a}{1 - \gamma}\right)\left(1 - \gamma\frac{L_{\min}}{n}\right) + \frac{1/a - \gamma}{1 - \gamma} \cdot \frac{L^*}{n} + O\left(\frac{\log n}{n}\right). \tag{10}$$

If $\log k = o(n)$, then $L_{\min}/n = a\log(nk)/n = o(1)$ and $L^*/n < L_{\min}/n = o(1)$. It follows that

$$\frac{\log|\mathcal{C}|}{nk} \leqslant \left(\frac{1-1/a}{1-\gamma}\right)(1-o(1)) + o(1) = \frac{1-1/a}{1-\gamma} + o(1).$$

Recall that $\log|\mathcal{X}_{n,k}| = k(n-\log(k/\mathsf{e})) + o(k)$. Hence, the code rate

$$\begin{aligned} R(\mathcal{C}) &= \frac{\log|\mathcal{C}|}{\log|\mathcal{X}_{n,k}|} = \frac{\log|\mathcal{C}|}{nk} \cdot \frac{nk}{k(n-\log(k/\mathsf{e})) + o(k)} \\ &\leqslant \left(\frac{1-1/a}{1-\gamma} + o(1)\right)\frac{1}{1-o(1)} = \frac{1-1/a}{1-\gamma} + o(1). \end{aligned}$$

If $\log k = \kappa n + o(n)$ where $0 < \kappa < 1$ is a real constant, then

$$\begin{aligned} \frac{nk}{\log|\mathcal{X}_{n,k}|} &= \frac{nk}{k(n-\log(k/\mathsf{e})) + o(k)} \\ &= \frac{1}{1-\kappa-o(1)} = \frac{1}{1-\kappa} + o(1). \end{aligned}$$

Therefore, it follows from (10) that the code rate satisfies

$$\begin{aligned} &R(\mathcal{C}) \\ &= \frac{\log|\mathcal{C}|}{\log|\mathcal{X}_{n,k}|} = \frac{\log|\mathcal{C}|}{nk}\frac{nk}{\log|\mathcal{X}_{n,k}|} \\ &\leqslant \left(\left(\frac{1-1/a}{1-\gamma}\right)(1-a\gamma\kappa) + \frac{1/a-\gamma}{1-\gamma}\frac{L^*}{n} + O\left(\frac{\log n}{n}\right)\right) \\ &\quad \times \left(\frac{1}{1-\kappa} - O\left(\frac{1}{n}\right)\right) \\ &= \frac{1-a\gamma\kappa}{1-\kappa}\left(\frac{1-1/a}{1-\gamma}\right) + \frac{1/a-\gamma}{(1-\gamma)(1-\kappa)} \cdot \frac{L^*}{n} + o(1). \end{aligned}$$

■

We note that Lemma 36 generalizes [25, Lemma 8], which focuses on the case of $k = 1$ and states that the rate of any single-strand $(L_{\min}, L_{\text{over}})$-trace reconstruction code is at most $\frac{1-1/a}{1-\gamma} + O\left(\frac{\log\log n}{\log n}\right)$.

Corollary 37: Suppose that $\log k = o(n)$. Let $\mathcal{C}$ be a multi-strand $(L_{\min}, L_{\text{over}})$-trace reconstruction code of $\mathcal{X}_{n,k}$. If $L_{\min} \leqslant \log(nk) + o(\log(nk))$, then $R(\mathcal{C}) = o(1)$.

Proof: Since $\mathcal{C}$ is also a multi-strand $(\lceil a\log(nk)\rceil, 0)$-trace reconstruction code for any $a > 1$, it follows from Lemma 36 that $R(\mathcal{C}) \leqslant 1-1/a+o(1)$ for all $a > 1$. Hence, $R(\mathcal{C}) = o(1)$. ■

Lemma 38: Suppose that $k \leqslant 2^n$. Let $\mathcal{C}$ be a multi-strand $(L_{\min}, L_{\text{over}})$-trace reconstruction code of $\mathcal{X}_{n,k}$. If $L_{\min} \leqslant \log(nk) + o(\log(nk))$ and $L_{\min} - L_{\text{over}} = \Theta(\log(nk))$, then $R(\mathcal{C}) = o(1)$.

Proof: It suffices to consider the case of $L_{\min} = \log(nk) + o(\log(nk))$. The proof is similar to that of Lemma 36. In this case, we denote

$$\begin{aligned} \hat{\mathcal{Y}}(\mathbf{x}) \triangleq &\left\{\mathbf{x}_{i(L_{\min}-L_{\text{over}})+[L_{\min}]} \;:\; i \in \left[\frac{n-L_{\text{over}}-L^*}{L_{\min}-L_{\text{over}}}\right]\right\} \\ &\cup \{\mathbf{x}[n-L_{\min}, n-1]\}. \end{aligned}$$

Since $L_{\min} - L^* \geqslant L_{\text{over}}$, each $\hat{\mathcal{Y}}(\mathcal{S}) = \bigcup_{i=0}^{k-1}\hat{\mathcal{Y}}(\mathbf{x}_i)$ is still an $(L_{\min}, L_{\text{over}})$-trace, and it consists of $k\left(\frac{n-L_{\text{over}}-L^*}{L_{\min}-L_{\text{over}}}+1\right)$ sequences of $\Sigma^{L_{\min}}$. Hence, we have that

$$|\mathcal{C}| \leqslant \binom{\frac{k(n+L_{\min}-2L_{\text{over}}-L^*)}{L_{\min}-L_{\text{over}}} + 2^{L_{\min}} - 1}{2^{L_{\min}} - 1}.$$

Since $L_{\min} = \log(nk) + o(\log(nk))$, we have

$$\frac{k(n+L_{\min}-2L_{\text{over}}-L^*)}{L_{\min}-L_{\text{over}}} < 2^{L_{\min}}.$$

Using the inequality in Lemma 35 and noting that $n+L_{\min}-2L_{\text{over}}-L^* \geqslant n-L_{\text{over}}$, we get that

$$\begin{aligned} &\frac{1}{nk}\log\binom{\frac{k(n+L_{\min}-2L_{\text{over}}-L^*)}{L_{\min}-L_{\text{over}}} + 2^{L_{\min}} - 1}{2^{L_{\min}} - 1} \\ &\leqslant \frac{k(n+L_{\min}-2L_{\text{over}}-L^*)}{(L_{\min}-L_{\text{over}})nk} \\ &\quad \times \left(2\log\mathsf{e} + L_{\min} - \log\left(\frac{k(n-L_{\text{over}})}{L_{\min}-L_{\text{over}}}\right)\right). \end{aligned} \tag{11}$$

Since $L_{\min} \leqslant \log(nk) + o(\log(nk))$ and $L_{\min} - L_{\text{over}} = \Theta(\log(nk))$, we have that $L_{\text{over}} \leqslant c_1\log(nk) \leqslant c_2 n$ for some constants $c_1, c_2 < 1$. It follows that $\log(k(n-L_{\text{over}})) = \log(nk) - O(1)$. Hence,

$$\begin{aligned} &2\log\mathsf{e} + L_{\min} - \log\left(\frac{k(n-L_{\text{over}})}{L_{\min}-L_{\text{over}}}\right) \\ &\leqslant 2\log\mathsf{e} + L_{\min} - \log(nk) + O(\log\log(nk)) \\ &= o(\log(nk)). \end{aligned}$$

Continuing (11), we have that

$$\begin{aligned} &\frac{1}{nk}\log\binom{\frac{k(n+L_{\min}-2L_{\text{over}}-L^*)}{L_{\min}-L_{\text{over}}} + 2^{L_{\min}} - 1}{2^{L_{\min}} - 1} \\ &\leqslant \left(1 + \frac{L_{\min}-2L_{\text{over}}-L^*}{n}\right)\frac{o(\log(nk))}{L_{\min}-L_{\text{over}}} = o(1). \end{aligned}$$

Hence,

$$R(\mathcal{C}) = \frac{\log|\mathcal{C}|}{\log|\mathcal{X}_{n,k}|} = \frac{\log|\mathcal{C}|}{nk} \cdot \frac{nk}{\log|\mathcal{X}_{n,k}|} = o(1).$$

■

Remark: We note that the condition $L_{\min} - L_{\text{over}} = \Theta(\log(nk))$ in Lemma 38 cannot be removed. A counterexample is the $(L_{\min}, L_{\text{over}}, e)$-trace reconstruction codes of rate $1 - o(1)$ in Theorem 34, where $L_{\text{over}} = \lceil\log nk\rceil + (24e + 13)\lceil\log\lceil\log nk\rceil\rceil + (4e+1)\lceil\log(4e+1)\rceil + 20e + 5$ and $L_{\min} \geqslant L_{\text{over}} + 1$.

Lemma 39: Suppose that $k \leqslant 2^n$. Let $\mathcal{C}$ be a multi-strand $(L_{\min}, L_{\text{over}})$-trace reconstruction code of $\mathcal{X}_{n,k}$. If $L_{\min} = \lceil a\log(nk)\rceil$ for some $a < 1$, then $R(\mathcal{C}) = o(1)$.

Proof: The proof is similar to that of Lemma 36. In this case, we denote

$$\hat{\mathcal{Y}}(\mathbf{x}) \triangleq \{\mathbf{x}_{0+[L_{\min}]}, \mathbf{x}_{1+[L_{\min}]}, \ldots, \mathbf{x}_{n-L_{\min}+[L_{\min}]}\}.$$

Then each $\hat{\mathcal{Y}}(\mathcal{S}) = \bigcup_{i=0}^{k-1}\hat{\mathcal{Y}}(\mathbf{x}_i)$ is still an $(L_{\min}, L_{\text{over}})$-trace, and it consists of $k(n-L_{\min}+1))$ sequences of $\Sigma^{L_{\min}}$, and so,

$$|\mathcal{C}| \leqslant \binom{k(n-L_{\min}+1) + 2^{L_{\min}} - 1}{2^{L_{\min}} - 1}.$$

We observe that $k(n-L_{\min}+1) \geqslant k(n-a\log n-a\log k) \geqslant k((1-a\kappa)n-a\log n) \geqslant cnk$ for some constant c and $2^{L_{\min}} \leqslant 2(nk)^a$. Since $a<1$, when n is sufficiently large, we have that $k(n-L_{\min}+1) \geqslant 2^{L_{\min}}$. Using the inequality in Lemma 35, we get that

$$\begin{aligned}&\frac{1}{nk}\log\binom{k(n-L_{\min}+1)+2^{L_{\min}}-1}{2^{L_{\min}}-1}\\ &\leqslant \frac{2^{L_{\min}}}{nk}(2\log \mathsf{e}+\log(k(n-L_{\min}+1))-L_{\min}). \quad (12)\end{aligned}$$

Noting that $kn > k(n-L_{\min}+1) \geqslant cnk$, we have that $\log(k(n-L_{\min}+1)) = \log(nk)-O(1)$. Continuing (12),

$$\begin{aligned}&\frac{1}{nk}\log\binom{k(n-L_{\min}+1)+2^{L_{\min}}-1}{2^{L_{\min}}-1}\\ &\leqslant \frac{2^{L_{\min}}}{nk}(2\log \mathsf{e}+\log(k(n-L_{\min}+1))-L_{\min})\\ &\leqslant \frac{2^{L_{\min}}}{nk}((1-a)\log(nk)+O(1))\\ &= \frac{(1-a)\log(nk)+O(1)}{(nk)^{1-a}} = o(1).\end{aligned}$$

Hence,

$$R(\mathcal{C}) = \frac{\log|\mathcal{C}|}{\log|\mathcal{X}_{n,k}|} = \frac{\log|\mathcal{C}|}{nk}\cdot\frac{nk}{\log|\mathcal{X}_{n,k}|} = o(1).$$

■

Note that a multistrand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code is also a multistrand $(L_{\min}, L_{\text{over}})$-trace reconstruction code. Hence, the upper bounds in Lemmas 36–39 also work for multistrand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction codes.

In the following, we study the lower bounds.

Theorem 40: Let $L_{\min} = \lceil a\log(nk)\rceil$ and $L_{\text{over}} = \lceil \gamma L_{\min}\rceil$, where $a>1$ and $0 \leqslant a\gamma \leqslant 1$. For all sufficiently large n,

1) if $\log k = o(n)$, then there is a multi-stand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code $\mathcal{D}$ of $\mathcal{X}_{n,k}$ of rate
$$R(\mathcal{D}) = \frac{1-1/a}{1-\gamma} - o(1);$$
2) if $\log k = \kappa n + o(n)$ where $0<\kappa<1$ is a real constant, then there is a multi-strand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code $\mathcal{D}$ of $\mathcal{X}_{n,k}$ of rate
$$R(\mathcal{D}) = \frac{1-a\gamma\kappa}{1-\kappa}\left(\frac{1-1/a}{1-\gamma}\right) - o(1).$$

Proof: Let $N = k(n-L_{\text{over}})+L_{\text{over}}$. Then $L_{\min} \geqslant \lceil a\log N\rceil$. According to Theorem 28 and Theorem 31, there is an $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code $\mathcal{C}$ of Σ^N whose rate is $\frac{1-1/a}{1-\gamma} - o(1)$. Applying Construction E with this code, we obtain a multi-strand $(L_{\min}, L_{\text{over}}, e)$-trace maximal reconstruction code $\mathcal{D}$ of $\mathcal{X}_{n,k}$ with $|\mathcal{D}| = |\mathcal{C}|$. Note that

$$\begin{aligned}\frac{N}{\log|\mathcal{X}_{n,k}|} &= \frac{k(n-L_{\text{over}})+L_{\text{over}}}{k(n-\log(k/e))+o(k)}\\ &= \frac{n-L_{\text{over}}+L_{\text{over}}/k}{n-\log k+O(1)}\\ &= 1-\frac{L_{\text{over}}-\log k-L_{\text{over}}/k+O(1)}{n-\log k+O(1)}\\ &= 1-\frac{(a\gamma-1)\log k+O(\log n)}{n-\log k+o(1)}.\end{aligned}$$

If $\log k = o(n)$, then $N/\log|\mathcal{X}_{n,k}| = 1-o(1)$, and so, we have that

$$R(\mathcal{D}) = \left(\frac{1-1/a}{1-\gamma}-o(1)\right)(1-o(1)) = \frac{1-1/a}{1-\gamma}-o(1).$$

If $\log k = \kappa n + o(n)$, then

$$\frac{N}{\log|\mathcal{X}_{n,k}|} = 1-\frac{(a\gamma-1)\kappa}{1-\kappa}-o(1) = \frac{1-a\gamma\kappa}{1-\kappa}-o(1),$$

and so, we have that

$$\begin{aligned}R(\mathcal{D}) &= \left(\frac{1-1/a}{1-\gamma}-o(1)\right)\left(\frac{1-a\gamma\kappa}{1-\kappa}-o(1)\right)\\ &= \frac{1-a\gamma\kappa}{1-\kappa}\left(\frac{1-1/a}{1-\gamma}\right)-o(1).\end{aligned}$$

■

When $\log k = o(n)$ or when $\log k = \kappa n + o(n)$ and $L^* = o(n)$, the lower bounds in Theorem 40 asymptotically achieve the upper bound in Lemma 36.

Next, we show that when $\log k = \kappa n + o(n)$ and $L_{\text{over}} = 0$, if $L^* \leqslant L_{\min} - (1+\epsilon)\log(nk) = (a-1-\epsilon)\log(nk)$ for a positive ϵ which is independent of n, then the upper bound in Lemma 36 still can be achieved.

Construction F: Suppose that $L_{\min} = \lceil a\log(nk)\rceil$ and $L_{\text{over}} = 0$. Denote $\bar{n} \triangleq \frac{n-L^*}{L_{\min}}$ and $K \triangleq \lceil\sqrt{\log(nk)}\rceil$. Let $I \triangleq \lceil\log(\bar{n}k)\rceil$ and $r_I \triangleq \lceil(3d+8)\log I\rceil$ where $d = 2e+1$. Then according to Theorem 17, there is a collection of $(3\lceil\frac{3}{2}\log(I+r_I)\rceil+\ell, d)$-WWL sequences $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{2^I-1} \in \Sigma^{I+r_I}$ such that the concatenation $\mathbf{c}_0\circ\mathbf{c}_1\circ\cdots\circ\mathbf{c}_{2^I-1}$ is an $(I+r_I, d)$-SD sequence.

Denote $n' \triangleq \bar{n}(L_{\min}-(I+r_I+K+\ell))+L^*$. Let $\mathcal{E}_{\texttt{WWL}}$ be the encoder in [14, Algorithm 2] which can encode sequences of $\Sigma^{n'-d}$ into $(\lceil K/4\rceil, d)$-WWL sequences[13] of $\Sigma^{n'}$. For a message $\mathbf{m} = \mathbf{m}_0\circ\mathbf{m}_1\circ\cdots\circ\mathbf{m}_{k-1}$ where $\mathbf{m}_i \in \Sigma^{n'-d}$ for $i\in[k]$, let $\mathbf{v}_i \triangleq \mathcal{E}_{\texttt{WWL}}(\mathbf{m}_i)$ for all $i\in[k]$. We partition each $\mathbf{v}_i$ into $\bar{n}+1$ substrings as follows:

$$\mathbf{v}_i = \mathbf{v}_{i,0}\circ\mathbf{v}_{i,1}\circ\cdots\mathbf{v}_{i,\bar{n}-1}\circ\mathbf{v}_{i,\bar{n}}$$

where $|\mathbf{v}_{i,j}| = L_{\min}-(I+r_I+K+\ell)$ for $j\in[\bar{n}]$ and $|\mathbf{v}_{i,\bar{n}}| = L^*$.

Denote $\mathbf{p} \triangleq 0^K\circ\mathbf{u}$ where $\mathbf{u}$ is a d-auto-cyclic sequence of length ℓ. For each $i\in[k]$, let

$$\begin{aligned}\mathbf{w}_i = \mathbf{v}_{i,0}\circ\mathbf{p}\circ\mathbf{c}_{i\bar{n}}\circ\mathbf{v}_{i,1}\circ\mathbf{p}\circ\mathbf{c}_{i\bar{n}+1}\circ\cdots\\ \circ\mathbf{v}_{i,\bar{n}-1}\circ\mathbf{p}\circ\mathbf{c}_{(i+1)\bar{n}-1}\circ\mathbf{v}_{i,\bar{n}}.\end{aligned}$$

Output $\{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{k-1}\}$ as the codeword which encodes the message $\{\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{k-1}\}$. The image of the mapping described here is the constructed code. ■

[13]Note that $n' = \Theta(n)$ and $K = \sqrt{\log(nk)} = \Theta(\sqrt{n})$. Hence, $K/4 \gg \mathcal{F}(n', d) = \log n' + (d-1)\log\log n' + O(1)$. Then according to Lemma 35 in [14], the encoder $\mathcal{E}_{\texttt{WWL}}$ does work.

Lemma 41: Suppose that $L^* \leqslant L_{\min} - (1+\epsilon)\log(nk)$ for a positive ϵ which is independent of n. Then the code obtained in Construction F is a multi-strand $(L_{\min}, 0, e)$-trace maximal reconstruction code of $\mathcal{X}_{n,k}$.

Proof: [Sketch of proof] Let $\mathbf{y}$ be a length-$L_{\min}$ substring of $\mathbf{w}_i$ for some $\mathbf{w}_i \in \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{k-1}\}$. Note that $L^* \leqslant L_{\min} - (1+\epsilon)\log(nk)$ and $|\mathbf{p} \circ \mathbf{c}_j| = K + \ell + I + r_I < (1+\epsilon)\log(nk)$. Then $\mathbf{y}$ must contain either a copy of $\mathbf{p} \circ \mathbf{c}_{i\bar{n}+j}$ or a suffix of $\mathbf{p} \circ \mathbf{c}_{i\bar{n}+j}$ together with a prefix of $\mathbf{p} \circ \mathbf{c}_{i\bar{n}+j+1}$. Note that $\mathbf{v}_{i,j}$'s and $\mathbf{c}_i$'s are $(K/4, d)$-WWL sequences and each has length $\Theta(\log(nk))$, except $\mathbf{v}_{i,\bar{n}}$'s which are of length L^*. Since $K = \lceil \sqrt{\log(nk)} \rceil$, it can be checked that the concatenations $\mathbf{c}_{i\bar{n}+j-1} \circ \mathbf{v}_{i,j}$'s satisfy the conditions in Lemma 25. Thus, even if $\mathbf{y}$ suffers from e errors, we can still locate the marker $\mathbf{p}$ in $\mathbf{y}$. Then we can run the locating algorithm of the robust positioning sequence $\mathbf{c}_0 \circ \mathbf{c}_1 \circ \cdots \circ \mathbf{c}_{2^I-1}$ to determine the index $i\bar{n}+j$ or $i\bar{n}+j+1$, and hence the location of $\mathbf{y}$. ∎

Theorem 42: Suppose that $\log k = \kappa n + o(n)$, $L_{\min} = \lceil a \log(nk) \rceil$ and $L_{\text{over}} = 0$, where $0 < \kappa < 1$ and $a > 1$. If $L^* \leqslant L_{\min} - (1+\epsilon)\log(nk)$ for a fixed positive ϵ which is independent of n, then there is a multi-strand $(L_{\min}, 0, e)$-trace maximal reconstruction code which has code rate

$$\frac{1-1/a}{1-\kappa} + \frac{1}{a(1-\kappa)} \cdot \frac{L^*}{n} - o(1)$$

Proof: Note that

$$\begin{aligned}
&\frac{n'-d}{n} \\
&= \frac{\bar{n}(L_{\min} - (I + r_I + K + \ell)) + L^* - d}{n} \\
&= \frac{n - \bar{n}(I + r_I + K + \ell) - d}{n} \\
&= 1 - \frac{1 - L^*/n}{L_{\min}}(I + r_I + K + \ell) - O\left(\frac{1}{n}\right) \\
&= 1 - \left(1 - \frac{L^*}{n}\right)\frac{\log(nk) + O(\sqrt{\log(nk)})}{a\log(nk)} - O\left(\frac{1}{n}\right) \\
&= 1 - \frac{1}{a} + \frac{L^*}{an} - O\left(\frac{1}{\sqrt{\log(nk)}}\right).
\end{aligned}$$

Hence, the code rate is

$$\begin{aligned}
\frac{(n'-d)k}{\log|\mathcal{X}_{n,k}|} &= \frac{(n'-d)k}{nk}\frac{nk}{\log|\mathcal{X}_{n,k}|} \\
&= \left(1 - \frac{1}{a} + \frac{L^*}{an} - o(1)\right)\left(\frac{1}{1-\kappa} - o(1)\right) \\
&= \frac{1-1/a}{1-\kappa} + \frac{1}{a(1-\kappa)}\frac{L^*}{n} - o(1).
\end{aligned}$$

∎

Finally, we note that the multi-strand $(L_{\min}, 0, e)$-trace reconstruction code in Construction F only guarantees recovering message from reliable $(L_{\min}, 0, e)$-erroneous traces, the occurrence of which might be rare since $L_{\text{over}} = 0$ and each symbol is usually included in a small number of substrings in $\mathcal{Y}$. Nevertheless, we can use a $(k, 2^{(n'-d)(k-r_o)}, 2\tau+1)_{2^{n'-d}}$ code to encode the message, like what we have done in Construction C, so that even if there are in total τ errors in $\mathcal{Y}$, we still can decode the message. The rate of this trace reconstruction code is

$$\left(1 - \frac{r_o}{k}\right)\left(\frac{1-1/a}{1-\kappa} + \frac{1}{a(1-\kappa)} \cdot \frac{L^*}{n}\right) - o(1).$$

REFERENCES

[1] J. Acharya, H. Das, O. Milenkovic, A. Orlitsky, and S. Pan, "String reconstruction from substring compositions," *SIAM J. Discrete Math.*, vol. 29, no. 3, pp. 1340–1371, 2015.

[2] D. Bar-Lev, S. Marcovich, E. Yaakobi, and Y. Yehezkeally, "Adversarial torn-paper codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 2934–2939.

[3] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," in *Proc. ACM-SIAM Symp. Discrete Algorithms (SODA)*, New Orleans, LA, USA, Jan. 2004, pp. 910–918.

[4] R. Berkowitz and S. Kopparty, "Robust positioning patterns," in *Proc. 27th Annu. ACM-SIAM Symp. Discrete Algorithms*, Arlington, VA, USA, 2016, pp. 1937–1951.

[5] A. M. Bruckstein, T. Etzion, R. Giryes, N. Gordon, R. J. Holt, and D. Shuldiner, "Simple and robust binary self-location patterns," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4884–4889, Jul. 2012.

[6] Y. M. Chee, D. T. Dao, H. M. Kiah, S. Ling, and H. Wei, "Robust positioning patterns with low redundancy," *SIAM J. Comput.*, vol. 49, no. 2, pp. 284–317, Jan. 2020.

[7] D. T. Dao, H. Mao Kiah, and H. Wei, "Maximum length of robust positioning sequences," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 108–113.

[8] M. Dudík and L. J. Schulman, "Reconstruction from subsequences," *J. Combinat. Theory, A*, vol. 103, no. 2, pp. 337–348, 2003.

[9] O. Elishco, R. Gabrys, E. Yaakobi, and M. Médard, "Repeat-free codes," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 5749–5764, Sep. 2021.

[10] R. Gabrys and O. Milenkovic, "Unique reconstruction of coded strings from multiset substring spectra," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 7682–7696, Dec. 2019.

[11] H. M. Kiah, G. J. Puleo, and O. Milenkovic, "Codes for DNA sequence profiles," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3125–3146, Jun. 2016.

[12] V. I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences," *J. Combinat. Theory A*, vol. 93, no. 2, pp. 310–332, 2001.

[13] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 2–22, Jan. 2001.

[14] M. Levy and E. Yaakobi, "Mutually uncorrelated codes for DNA storage," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3671–3691, Jun. 2019.

[15] B. Manvel, A. Meyerowitz, A. Schwenk, K. Smith, and P. Stockmeyer, "Reconstruction of sequences," *Discrete Math.*, vol. 94, no. 3, pp. 209–219, 1991.

[16] S. Marcovich and E. Yaakobi, "Reconstruction of strings from their substrings spectrum," *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4369–4384, Jul. 2021.

[17] S. Nassirpour, I. Shomorony, and A. Vahid, "Reassembly codes for the chop-and-shuffle channel," 2022, *arXiv:2201.03590*.

[18] S. Pattabiraman, R. Gabrys, and O. Milenkovic, "Coding for polymer-based data storage," *IEEE Trans. Inf. Theory*, vol. 69, no. 8, pp. 4812–4836, Aug. 2023.

[19] A. N. Ravi, A. Vahid, and I. Shomorony, "Capacity of the torn paper channel with lost pieces," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, VIC, Australia, Jul. 2021, pp. 1937–1942.

[20] A. D. Scott, "Reconstructing sequences," *Discrete Math.*, vol. 175, nos. 1–3, pp. 231–238, 1997.

[21] I. Shomorony and A. Vahid, "Torn-paper coding," *IEEE Trans. Inf. Theory*, vol. 67, no. 12, pp. 7904–7913, Dec. 2021.

[22] E. Ukkonen, "Approximate string-matching with q-grams and maximal matches," *Theor. Comput. Sci.*, vol. 92, no. 1, pp. 191–211, 1992.

[23] C. Wang, J. Sima, and N. Raviv, "Break-resilient codes for forensic 3D fingerprinting," 2023, *arXiv:2310.03897*.

[24] H. Wei, "Nearly optimal robust positioning patterns," *IEEE Trans. Inf. Theory*, vol. 68, no. 1, pp. 193–203, Jan. 2022.

[25] Y. Yehezkeally, D. Bar-Lev, S. Marcovich, and E. Yaakobi, "Generalized unique reconstruction from substrings," *IEEE Trans. Inf. Theory*, vol. 69, no. 9, pp. 5648–5659, Sep. 2023.

[26] Y. Yehezkeally and N. Polyanskii, "On codes for the noisy substring channel," 2021, *arXiv:2102.01412*.

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

He was a Post-Doctoral Fellow with Capital Normal University, Beijing, China; a Research Fellow with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore; and a Post-Doctoral Fellow with the School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Israel. He is currently a Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China. He is also an Associate Researcher with the Peng Cheng Laboratory, Shenzhen, China. His research interests include combinatorial design theory, coding theory, and their intersections.

Dr. Wei received the 2017 Kirkman Medal from the Institute of Combinatorics and its Applications.

Moshe Schwartz (Fellow, 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 2012–2014, he was a Visiting Scientist with 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 (on a leave of absence), and the Department of Electrical and Computer Engineering, McMaster University, Canada. 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 of IEEE TRANSACTIONS ON INFORMATION THEORY from 2014 to 2021. Since 2021, he has been an Area Editor for Coding and Decoding of IEEE TRANSACTIONS ON INFORMATION THEORY. He has been an Editorial Board Member of *Journal of Combinatorial Theory, Series A*, since 2021.

Gennian Ge received the M.S. and Ph.D. degrees in mathematics from Suzhou University, Suzhou, Jiangsu, China, in 1993 and 1996, respectively. After that, he became a member of Suzhou University. He was a Post-Doctoral Fellow with the Department of Computer Science, Concordia University, Montreal, QC, Canada, from September 2001 to August 2002; and a Visiting Assistant Professor with the Department of Computer Science, University of Vermont, Burlington, VT, USA, from September 2002 to February 2004. He was a Full Professor with the Department of Mathematics, Zhejiang University, Hangzhou, Zhejiang, China, from March 2004 to February 2013. Currently, he is a Full Professor with the School of Mathematical Sciences, Capital Normal University, Beijing, China. His research interests include combinatorics, coding theory, information security, and their interactions. He received the 2006 Hall Medal from the Institute of Combinatorics and its Applications. He is on the editorial board of *Journal of Combinatorial Theory, Series A*, IEEE TRANSACTIONS ON INFORMATION THEORY, *Designs Codes and Cryptography*, *Journal of Combinatorial Designs*, *Journal of Algebraic Combinatorics*, *Science China Mathematics*, and *Applied Mathematics—A Journal of Chinese Universities*.