Adversarial Torn-Paper Codes

Daniella Bar-Lev, *Student Member, IEEE*, Sagi Marcovich, *Student Member, IEEE*, Eitan Yaakobi, *Senior Member, IEEE*, and Yonatan Yehezkeally, *Member, IEEE*

***Abstract*— We study the *adversarial torn-paper channel*. This problem is motivated by applications in DNA data storage where the DNA strands that carry information may break into smaller pieces which are received out of order. Our model extends the previously researched probabilistic setting to the worst-case. We develop code constructions for any parameters of the channel for which non-vanishing asymptotic rate is possible and show our constructions achieve asymptotically optimal rate while allowing for efficient encoding and decoding. Finally, we extend our results to related settings included multi-strand storage, presence of substitution errors, or incomplete coverage.**

***Index Terms*— Sequence reconstruction, DNA sequences, error correction codes, worst-case analysis.**

I. Introduction

HIGH density and extreme longevity make DNA an appealing medium for data storage, especially for archival purposes [4], [10], [36]. Advances in DNA synthesis and sequencing technologies and recent proofs of concept [5], [9], [17], [25] have ignited active research into the capacity and challenges of data storage in this medium.

An aspect of this medium is that typically only short DNA sequences may be read; information molecules are therefore broken up into pieces and then read out of order, such as in shotgun sequencing [6], [14], [23], [27]. Multiple channel models have recently been suggested and studied based on this property. An assumption of overlap in read substrings and (near) uniform coverage leads to the problem of string reconstruction from substring composition [3], [6], [15], [22], [29]; on the contrary, assuming no overlap in read substrings leads to the *torn-paper problem* [24], [26], a problem closely related to the shuffling channel [18], [19], [30], [35]. This problem is motivated by DNA-based storage systems, where the information is stored in synthesized strands of DNA molecules. However, during and after synthesis, the DNA strands may break into smaller segments and due to the lack of ordering among the strands in these systems, all broken segments can only be read out of order [32]. Thus, the goal is to successfully retrieve the data from this collection of read segments of the broken DNA strands.

In the *torn-paper channel* [26], [32], also known as the *chop-and-shuffle channel* [24], a long information string is segmented into non-overlapping substrings and their length has some known distribution. The channel outputs an unordered collection of these substrings, preserving their left-to-right orientation. Given the lengths' distribution, the goal is to determine the channel capacity and devise efficient coding techniques. The geometric distribution was first studied in [32], and later in [24] using the Varshamov-Tenengolts (VT) codes [34]. Subsequently, [26] considered almost arbitrary distributions while, additionally, extending the problem by introducing incomplete coverage, i.e., assuming some of the substrings are deleted with some probability.

The torn-paper channel was studied so far only in the probabilistic setting. The goal of this paper is to extend this channel to the worst case, referred to herein as the *adversarial torn-paper channel*. Namely, it is assumed that an information string is adversarially segmented into non-overlapping substrings, where the length of each substring is between $L_{\min}$ and $L_{\max}$, for some given $L_{\min}$ and $L_{\max}$. We show that the capacity of this channel is determined by $L_{\min}$, whereas the capacity of the probabilistic channel was shown to depend on the *average* substring length; nevertheless, we choose this adversarial model here for ease of analysis, and observe that under this setting the average substring length might indeed approach $L_{\min}$. For further discussion of an average-restricted adversary, see Section V.

We study the noiseless adversarial torn-paper channel for a single information string, as well as multiple strings, which is motivated by DNA sequencing technologies where multiple strings are sequenced simultaneously [8], [21], [28]. We also extend the model to either allow for substitution errors affecting the information string prior to segmentation, or for incomplete coverage due to deletion of several segments after the segmentation. In all cases we investigate the values of $L_{\min}$ and $L_{\max}$ that permit codes with non-vanishing asymptotic rates, and develop constructions of codes with

Manuscript received 26 August 2022; revised 20 March 2023; accepted 29 June 2023. Date of publication 6 July 2023; date of current version 15 September 2023. This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 801434). It was also funded by the European Union (ERC, DNAStorage, 865630). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. An earlier version of this paper was presented in part at the 2022 IEEE International Symposium on Information Theory [DOI: 10.1109/ISIT50566.2022.9834766]. *(Daniella Bar-Lev, Sagi Marcovich, and Yonatan Yehezkeally contributed equally to this work.) (Corresponding author: Yonatan Yehezkeally.)*

Daniella Bar-Lev, Sagi Marcovich, and Eitan Yaakobi are with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel (e-mail: daniellalev@cs.technion.ac.il; sagimar@cs.technion.ac.il; yaakobi@cs.technion.ac.il).

Yonatan Yehezkeally is with the Institute for Communications Engineering, School of Computation, Information and Technology, Technical University of Munich, 80333 Munich, Germany (e-mail: yonatan.yehezkeally@tum.de).

Communicated by S. Ghorpade, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2023.3292895.

Digital Object Identifier 10.1109/TIT.2023.3292895

efficient encoding and decoding algorithms, asymptotically achieving optimal rates.

The rest of this paper is organized as follows. In Section II, the definitions and notations that will be used throughout the paper are presented, as well as a lower bound on $L_{\min}$ required for the existence of codes for the adversarial torn-paper channel with non-vanishing asymptotic rates. In Section III we first study the application of a known code construction to the adversarial channel, and observe its limitations in that setting; then, we present the basic construction used throughout the paper for the noiseless case of the single-strand adversarial torn-paper channel, and extend it to the multi-strand case. In Section IV we extend our construction to two noisy settings, including substitution errors or incomplete coverage. We conclude with a summary and remarks in Section V.

II. Definitions and Preliminaries

Let Σ be a finite alphabet of size q. For convenience of presentation, we assume Σ is equipped with a ring structure, and in particular identify elements $0, 1 \in \Sigma$. For a positive integer n, let $[n]$ denote the set $[n] \triangleq \{0, 1, \ldots, n-1\}$. Let Σ^* denote the set of all finite strings over Σ. The length of a string $\boldsymbol{x} \in \Sigma^*$ is denoted by $|\boldsymbol{x}|$. We also denote, for $\boldsymbol{x} = (x_i)_{i\in[n]} \in \Sigma^n$, its *support* $\mathrm{supp}(\boldsymbol{x}) \triangleq \{i \in [n] : x_i \neq 0\}$, and $\|\boldsymbol{x}\| \triangleq |\mathrm{supp}(\boldsymbol{x})|$. For strings $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$, we denote their concatenation by $\boldsymbol{x} \circ \boldsymbol{y}$. We say that $\boldsymbol{v}$ is a *substring*, or *segment*, of $\boldsymbol{x}$ if there exist strings $\boldsymbol{u}, \boldsymbol{w}$ (perhaps empty) such that $\boldsymbol{x} = \boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{w}$. If $|\boldsymbol{v}| = \ell$, we specifically say that $\boldsymbol{v}$ is an *ℓ-substring* (*ℓ-segment*) of $\boldsymbol{x}$. If $|\boldsymbol{u}| = i$ then it is said that $\boldsymbol{v}$ is the substring (similarly, ℓ-substring) of $\boldsymbol{x}$ at *location* i. We say that $\boldsymbol{v}$ appears *cyclically* in $\boldsymbol{x}$, at location i, if $\boldsymbol{x} = \boldsymbol{u}\circ\boldsymbol{w}$ and $\boldsymbol{v}$ is the substring of $\boldsymbol{w}\circ\boldsymbol{u}$ at location $(i-|\boldsymbol{u}|)$. For example, 010 is the 3-substring of 00101 at location 1, and also its 3-substring at location 3, where the latter is a cyclic appearance. We avoid using the term *index* as it is reserved to elements of presented constructions.

In our setting, information is stored in an unordered collection of strings over Σ; it might be allowed for the same string to appear with multiplicity in the collection, which is encapsulated in the following formal definition:

$$\mathcal{X}_{n,k} \triangleq \{S = \{\!\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}\}\!\} : \forall i, \boldsymbol{x}_i \in \Sigma^n\}.$$

Here, $\{\!\{a, a, b, \ldots\}\!\}$ denotes a multiset; i.e., elements appear with multiplicity (but no order). Note that $|\mathcal{X}_{n,k}| = \binom{k+q^n-1}{k}$. It is assumed that a message $S \in \mathcal{X}_{n,k}$ is read by segmenting all elements of S into non-overlapping substrings of lengths between some fixed values $L_{\min}$ and $L_{\max}$, and all segments are received, possibly with multiplicity, without order or information on which element they originated from. More formally, a *segmentation* of the string $\boldsymbol{x}$ is a multiset $\{\!\{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{m-1}\}\!\}$, where $\boldsymbol{x}$ can be presented as $\boldsymbol{x} = \boldsymbol{u}_0 \circ \boldsymbol{u}_1 \circ \cdots \circ \boldsymbol{u}_{m-1}$. In case $L_{\min} \leqslant |\boldsymbol{u}_i| \leqslant L_{\max}$ for $0 \leqslant i < m-1$ and $|\boldsymbol{u}_{m-1}| \leqslant L_{\max}$, then the segmentation is called an $(L_{\min}, L_{\max})$-*segmentation*. The set of all $(L_{\min}, L_{\max})$-segmentations of $\boldsymbol{x}$ is denoted by $\mathcal{T}_{L_{\min}}^{L_{\max}}(\boldsymbol{x})$ and is referred as the $(L_{\min}, L_{\max})$-*segmentation spectrum of* $\boldsymbol{x}$. For example,

$$\mathcal{T}_2^3(00101) = \Big\{\{\!\{001, 01\}\!\}, \{\!\{00, 101\}\!\}, \{\!\{00, 10, 1\}\!\}\Big\}.$$

These definitions are naturally extended for a multiset $S \in \mathcal{X}_{n,k}$, so a *segmentation* of S is a union (as a multiset) of segmentations of all the strings in S (and the same holds for an $(L_{\min}, L_{\max})$-segmentation), and $\mathcal{T}_{L_{\min}}^{L_{\max}}(S)$, the $(L_{\min}, L_{\max})$-*segmentation spectrum of* S, is the set of all $(L_{\min}, L_{\max})$-segmentations of S.

Note that our channel model only restricts the length of the last segment to be at most $L_{\max}$. Such a relaxation is motivated in applications where segmentation of the strings occurs sequentially, so that it might happen that the last segment is shorter than $L_{\min}$, but not larger than $L_{\max}$.

A code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ is said to be an $(L_{\min}, L_{\max})$-*multistrand torn-paper code* if for all $S, S' \in \mathcal{C}$, $S \neq S'$, it holds that all possible $(L_{\min}, L_{\max})$-segmentations of S, S' are distinct. That is, $\mathcal{T}_{L_{\min}}^{L_{\max}}(S) \cap \mathcal{T}_{L_{\min}}^{L_{\max}}(S') = \emptyset$. For $k = 1$, we simply refer to $(L_{\min}, L_{\max})$-*single strand torn-paper codes*.

In case $L_{\min} = L_{\max} = \ell$, then for convenience, we let $\mathcal{T}_\ell(\boldsymbol{x}) \triangleq \mathcal{T}_\ell^\ell(\boldsymbol{x})$ and $\mathcal{T}_\ell(S) \triangleq \mathcal{T}_\ell^\ell(S)$ and note that in this case $|\mathcal{T}_\ell(\boldsymbol{x})| = |\mathcal{T}_\ell(S)| = 1$. For example, if $S = \{\!\{01010, 00101, 11101\}\!\}$ (which may be thought of as a multiset), then

$$\mathcal{T}_2(S) = \Big\{\{\!\{01, 01, 0, 00, 10, 1, 11, 10, 1\}\!\}\Big\}.$$

Note that $\mathcal{T}_\ell(S)$ is only one possible channel output given input S. Nevertheless, $\mathcal{T}_{L_{\min}}(S) \subseteq \mathcal{T}_{L_{\min}}^{L_{\max}}(S)$ for all S and $L_{\min} \leqslant L_{\max}$, hence every $(L_{\min}, L_{\max})$-multistrand torn-paper code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ satisfies

$$|\mathcal{C}| \leqslant |\{\mathcal{T}_{L_{\min}}(S) : S \in \mathcal{X}_{n,k}\}|. \tag{1}$$

For all $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ we denote the *rate*, *redundancy* of $\mathcal{C}$ by $R(\mathcal{C}) \triangleq \frac{\log|\mathcal{C}|}{\log|\mathcal{X}_{n,k}|}$, $\mathrm{red}(\mathcal{C}) \triangleq \log|\mathcal{X}_{n,k}| - \log|\mathcal{C}|$, respectively. Throughout the paper, we use the base-q logarithms.

For two non-negative functions f, g of a common variable n, denoting $L \triangleq \limsup_{n\to\infty} \frac{f(n)}{g(n)}$ (in the wide sense, i.e., $L = \infty$ if $\frac{f(n)}{g(n)}$ is unbounded) we say that $f = o_n(g)$ if $L = 0$, $f = \Omega_n(g)$ if $L > 0$, $f = O_n(g)$ if $L < \infty$, and $f = \omega_n(g)$ if $L = \infty$. If f is not positive, we say $f = O_n(g)$ ($f = o_n(g)$) if $|f| = O_n(g)$) (respectively, $|f| = o_n(g)$). We say that $f = \Theta_n(g)$ if $f = \Omega_n(g)$ and $f = O_n(g)$. If clear from context, we omit the subscript from aforementioned notations.

We conclude this section by observing a lower bound on the required segment length $L_{\min}$ for multi-strand torn-paper codes to achieve non-vanishing rates, and in particular rates approaching one.

Lemma 1: If $\log(k) = o(n)$ and $L_{\min} = a\log(nk) + O_{nk}(1)$ for some $a \geqslant 1$, then

$$\begin{aligned}&\log|\mathcal{X}_{n,k}| - \log|\{\mathcal{T}_{L_{\min}}(S) : S \in \mathcal{X}_{n,k}\}| \\ &\quad \geqslant nk\left(\frac{1}{a} - a\frac{\log(k)}{n} - O\left(\frac{\log\log(nk)}{\log(nk)}\right)\right).\end{aligned}$$

Proof: First, note that

$$|\mathcal{X}_{n,k}| = \binom{k+q^n-1}{k} \geqslant \frac{q^{nk}}{k!} \geqslant \frac{q^{nk}}{k^k},$$

and hence $\log|\mathcal{X}_{n,k}| \geqslant (n - \log(k))k$. Next, since $|\{\mathcal{T}_{L_{\min}}(S) : S \in \mathcal{X}_{n,k}\}|$ is monotonically non-decreasing

in n, we have that

$$|\{\mathcal{T}_{L_{\min}}(S) : S \in \mathcal{X}_{n,k}\}| \leqslant \binom{k\lceil n/L_{\min}\rceil + q^{L_{\min}} - 1}{q^{L_{\min}} - 1} \leqslant \binom{k\lceil n/L_{\min}\rceil + q^{L_{\min}}}{q^{L_{\min}}}.$$

Now, for $v \geqslant u \geqslant 0$ we observe

$$\log\binom{u+v}{u} \leqslant \log\frac{1}{u!}(u+v)^u \leqslant u\log\Big(e(1+\frac{v}{u})\Big) \leqslant u\Big(\Big(1+\frac{u}{v}\Big)\log(e) + \log(\frac{v}{u})\Big) \leqslant u(2\log(e) + \log(\frac{v}{u})),$$

where we used $\log(1+x) \leqslant \frac{\log(e)}{x} + \log(x)$. Setting $u \triangleq k\lceil n/L_{\min}\rceil \leqslant \frac{nk}{L_{\min}} + k$ and $v \triangleq q^{L_{\min}} = \Theta((nk)^a)$, we have $\frac{v}{u} = \Theta\big((nk)^{a-1}L_{\min}\big) = \Theta\big((nk)^{a-1}\log(nk)\big)$, and therefore $\log\big(\frac{v}{u}\big) = (a-1)\log(nk) + \log\log(nk) + O(1)$. We then conclude

$$\begin{aligned}
&\log|\{\mathcal{T}_{L_{\min}}(S) : S \in \mathcal{X}_{n,k}\}| \\
&\leqslant \left(\frac{nk}{L_{\min}} + k\right)\big((a-1)\log(nk) + \log\log(nk) + O(1)\big) \\
&= (a-1)\frac{nk\log(nk)}{L_{\min}} + k(a-1)\log(nk) + \left(\frac{nk}{L_{\min}} + k\right)(\log\log(nk) + O(1)) \\
&= (a-1)\frac{nk\log(nk)}{L_{\min}} + k(a-1)\log(nk) + O\left(\frac{nk\log\log(nk)}{\log(nk)}\right) \\
&= nk\left((a-1)\frac{\log(nk)}{L_{\min}} + (a-1)\frac{\log(k)}{n} + O\left(\frac{\log(n)}{n}\right) + O\left(\frac{\log\log(nk)}{\log(nk)}\right)\right) \\
&= nk\left(\frac{a-1}{a+O(1/\log(nk))} + (a-1)\frac{\log(k)}{n} + O\left(\frac{\log\log(nk)}{\log(nk)}\right)\right) \\
&= nk\left(\frac{a-1}{a} + (a-1)\frac{\log(k)}{n} + O\left(\frac{\log\log(nk)}{\log(nk)}\right)\right),
\end{aligned}$$

which verifies the lemma's statement. ■

We note that throughout this paper, we perform redundancy analysis to the second-most-significant term, and retain the order or magnitude for the reminder; since proofs demonstrate that this asymptotic notation does not in fact hide significant coefficients, we believe this representation is faithful for the purpose of finite-length analysis, as well.

The implications of Lemma 1 are more clearly stated in the next corollary.

Corollary 2: Let $\mathcal{C}$ be any $(L_{\min}, L_{\max})$-multistrand torn-paper code. Assuming $\log(k) = o(n)$, if $L_{\min} = (a + o_{nk}(1))\log(nk)$, for some $a \geqslant 1$, then $R(\mathcal{C}) \leqslant 1 - \frac{1}{a} + o_{nk}(1)$.

Proof: From Eq. (1) and Lemma 1 we have

$$R(\mathcal{C}) \leqslant \frac{\log|\{\mathcal{T}_{L_{\min}}(S) : S \in \mathcal{X}_{n,k}\}|}{\log|\mathcal{X}_{n,k}|} \leqslant 1 - \frac{nk}{\log|\mathcal{X}_{n,k}|}\left(\frac{1}{a} - a\frac{\log(k)}{n} - O\left(\frac{\log\log(nk)}{\log(nk)}\right)\right),$$

which, together with $\log|\mathcal{X}_{n,k}| \leqslant nk$, justifies the claim. ■

III. Constructions of Torn-Paper Codes

In this section we study constructions of torn-paper codes, in context of the bound of Corollary 2.

A. Related Works: Pilot-Based Construction

An explicit and efficient coding scheme was presented in [32] for the probabilistic torn-paper channel. Therein, it was argued that an indexing approach to coding is challenging due to the a priori unknown locations of segmentation by the channel, hence this construction relied on interleaving a *pilot* (or *phase-detection sequence*). We describe this scheme below to study its performance in the adversarial channel.

Construction P: [32, Sec. VII] Fix an integer $m > 1$. Let n be a multiple of m, to be determined later, and s an integer satisfying $s \geqslant \log(n/m)$. Let $\boldsymbol{p} \in \Sigma^{n/m}$ be any (n/m)-segment of a de Bruijn sequence [11] of order s, which we refer to as the *pilot*.

For $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^{n/m}$, denote $\boldsymbol{x} \perp^s \boldsymbol{y}$ if $\boldsymbol{x}, \boldsymbol{y}$ have no common s-segment, i.e., if for all $i, j \in [n/m - s + 1]$ it holds that $\boldsymbol{x}^{(i)} \neq \boldsymbol{y}^{(j)}$, where $\boldsymbol{x}^{(i)}$ ($\boldsymbol{y}^{(j)}$) is the s-segment of $\boldsymbol{x}$ (respectively, $\boldsymbol{y}$) at location i (respectively, j). Then, we denote $\mathcal{O}_{\boldsymbol{p}} \triangleq \big\{\boldsymbol{c} \in \Sigma^{n/m} : \boldsymbol{c} \perp^s \boldsymbol{p}\big\}$.

For any code $\mathcal{C} \subseteq \Sigma^{n/m}$, we construct a code $\mathcal{C}_{\text{pilot}} \subseteq \Sigma^n$ as follows: for every choice of $m-1$ elements $(\boldsymbol{c}_j)_{j\in[m-1]} \subseteq \mathcal{C} \cap \mathcal{O}_{\boldsymbol{p}}$ (allowing for repetition), we interleave a single symbol from each $\boldsymbol{p}, \boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{m-2}$, in order, to construct a codeword $\boldsymbol{c} \in \mathcal{C}_{\text{pilot}}$. □

Example 3: Let $q = 2, m = 2, n = 12, s = 3$. We choose 00010111 as the binary de Bruijn sequence of order s, and let $\boldsymbol{p} \triangleq 000101$ be its (n/m)-prefix. Then,

$$\mathcal{O}_{\boldsymbol{p}} = \big\{011100, 011110, 011111, 111100, 111110, 111111\big\}.$$

Letting $\mathcal{C} \triangleq \Sigma^5$, and for any choice of $m-1 = 1$ element of $\mathcal{C} \cap \mathcal{P}_{\boldsymbol{p}} = \mathcal{O}_{\boldsymbol{p}}$, we interleave $\boldsymbol{p}$ with that element to derive the code

$$\mathcal{C}_{\text{pilot}} = \big\{000101110010, 000101110110, 000101110111, 010101110010, 010101110110, 010101110111\big\}.$$

Lemma 4: [32, Sec. VII-B] For all $s \geqslant \log(n/m)$ it holds that $\mathcal{C}_{\text{pilot}}$ is an $(ms, L_{\max})$-single strand torn-paper code, for any $L_{\max} \geqslant ms$.

Proof: We replicate the proof for completeness. Observe that every (ms)-segment $\boldsymbol{u}$ of $\boldsymbol{c} \in \mathcal{C}_{\text{pilot}}$ contains s consecutive symbols from each $\boldsymbol{p}, \boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{m-2}$; since $\boldsymbol{c}_j \perp^s \boldsymbol{p}$

for every $j \in [m-1]$, the s-segment of $\boldsymbol{p}$ can be uniquely identified. Since $\boldsymbol{p}$ is a segment of a de Bruijn sequence of order s, the location in $\boldsymbol{p}$ of the observed segment can be deduced, and hence the location of $\boldsymbol{u}$ in $\boldsymbol{c}$ can readily be obtained. ■

Example 5: Continuing Example 3, assume $010101110110 \in \mathcal{C}_{\text{pilot}}$ is passed through an adversarial torn-paper channel with $L_{\min} = ms = 6$ and, say, $L_{\max} = 8$. The received segments are

$$010101, 110110.$$

Taking the first segment, we decompose the two interleaved strings

$$\bar{\boldsymbol{c}}_0 = 000, \bar{\boldsymbol{c}}_1 = 111;$$

we identify $\bar{\boldsymbol{c}}_0$ as *the* s-substring of $\boldsymbol{p}$ at location 0, implying that $\bar{\boldsymbol{c}}_1$ is the substring of $\boldsymbol{c}_0$ at location 0. Similarly, we decompose the second segment into

$$\tilde{\boldsymbol{c}}_0 = 101, \tilde{\boldsymbol{c}}_1 = 110;$$

since 101 is the s-substring of $\boldsymbol{p}$ at location 3, we also have that $\tilde{\boldsymbol{c}}_1$ is the substring of $\boldsymbol{c}_0$ at location 3, i.e., $\boldsymbol{c}_0 = 111110 \in \mathcal{O}_{\boldsymbol{p}}$, confirming $010101110110 \in \mathcal{C}_{\text{pilot}}$ was the transmitted sequence.

For the probabilistic channel studied in [32], $\mathcal{C}$ in Construction P was chosen to be an error-correcting code. Note from the proof of Lemma 4 that in our chosen adversarial setting, this is redundant; that element of the construction is preserved in our presentation to support the discussion in Section V regarding alternate models.

Next, we turn to find the achievable rates of Construction P.

Corollary 6: $R(\mathcal{C}_{\text{pilot}}) = \left(1 - \frac{1}{m}\right) R(\mathcal{C} \cap \mathcal{O}_{\boldsymbol{p}})$.

Proof: Observe that $|\mathcal{C}_{\text{pilot}}| = |\mathcal{C} \cap \mathcal{O}_{\boldsymbol{p}}|^{m-1} = q^{n\left(1-\frac{1}{m}\right)\log(|\mathcal{C}\cap\mathcal{O}_{\boldsymbol{p}}|)/\frac{n}{m}} = q^{n\left(1-\frac{1}{m}\right)R(\mathcal{C}\cap\mathcal{O}_{\boldsymbol{p}})}$. ■

The following lemma was implied by [32, Sec. VII-A].

Lemma 7: For all $\mathcal{C} \subseteq \Sigma^{n/m}$ there exists $\boldsymbol{z} \in \Sigma^{n/m}$ such that

$$R((\boldsymbol{z} + \mathcal{C}) \cap \mathcal{O}_{\boldsymbol{p}}) \geqslant R(\mathcal{C}) - (1 - R(\mathcal{O}_{\boldsymbol{p}})),$$

where $\boldsymbol{z} + \mathcal{C} \triangleq \{\boldsymbol{z} + \boldsymbol{c} : \boldsymbol{c} \in \mathcal{C}\}$.

Proof: Observe that

$$\begin{aligned}\sum_{\boldsymbol{z}\in\Sigma^{n/m}} |(\boldsymbol{z}+\mathcal{C}) \cap \mathcal{O}_{\boldsymbol{p}}| &= \sum_{\boldsymbol{z}\in\Sigma^{n/m}} \sum_{\substack{\boldsymbol{c}_1\in\mathcal{C}\\ \boldsymbol{c}_2\in\mathcal{O}_{\boldsymbol{p}}}} \mathbb{1}_{\boldsymbol{z}+\boldsymbol{c}_1=\boldsymbol{c}_2} \\ &= \sum_{\substack{\boldsymbol{c}_1\in\mathcal{C}\\ \boldsymbol{c}_2\in\mathcal{O}_{\boldsymbol{p}}}} \sum_{\boldsymbol{z}\in\Sigma^{n/m}} \mathbb{1}_{\boldsymbol{z}=\boldsymbol{c}_2-\boldsymbol{c}_1} \\ &= \sum_{\substack{\boldsymbol{c}\in\mathcal{C}\\ \boldsymbol{o}\in\mathcal{O}_{\boldsymbol{p}}}} 1 = |\mathcal{C}| \cdot |\mathcal{O}_{\boldsymbol{p}}|.\end{aligned}$$

It follows from the pigeonhole principle that there exists $\boldsymbol{z} \in \Sigma^{n/m}$ such that $|(\boldsymbol{z}+\mathcal{C}) \cap \mathcal{O}_{\boldsymbol{p}}| \geqslant q^{-n/m}|\mathcal{C}| \cdot |\mathcal{O}_{\boldsymbol{p}}|$, which concludes the proof. ■

In the rest of the section, it remains to analyze what values of s assure that $1 - R(\mathcal{O}_{\boldsymbol{p}}) = o_n(1)$; we also discuss the implications of these available choices.

Lemma 8: [32, Sec. VII-A] If $s \triangleq \lceil (2+\delta)\log(n/m) \rceil$ for some $\delta > 0$, then, using $\mathcal{C} \triangleq \Sigma^{n/m}$ in Construction P,

$$\begin{aligned}R(\mathcal{C}_{\text{pilot}}) &\geqslant 1 - \frac{1}{m} - \frac{m-1}{n} \cdot \frac{1}{(n/m)^{\delta} - 1} \\ &= 1 - \frac{1}{m} - o_n(1).\end{aligned}$$

Proof: Again, we replicate the proof here. Denote for a uniformly chosen $\boldsymbol{c} \in \Sigma^{n/m}$ the event $A_{i,j}$ that $\boldsymbol{c}^{(i)} = \boldsymbol{p}^{(j)}$. Clearly $\Pr(A_{i,j}) = q^{-s}$; using the union bound, $\Pr(\boldsymbol{c} \perp^s \boldsymbol{p}) \geqslant 1 - (n/m)^2 q^{-s} \geqslant 1 - (n/m)^{-\delta}$, i.e.,

$$|\mathcal{O}_{\boldsymbol{p}}| \geqslant q^{n/m}\left(1 - (n/m)^{-\delta}\right).$$

It follows from Corollary 6 that

$$\begin{aligned}R(\mathcal{C}_{\text{pilot}}) &= \left(1 - \frac{1}{m}\right) R(\mathcal{O}_{\boldsymbol{p}}) \\ &\geqslant \left(1 - \frac{1}{m}\right)\left(1 + \frac{m}{n}\log\left(1 - (n/m)^{-\delta}\right)\right) \\ &\geqslant 1 - \frac{1}{m} - \frac{(m-1)(n/m)^{-\delta}}{n(1 - (n/m)^{-\delta})}.\end{aligned}$$

■

Unfortunately, Lemma 8 doesn't match the upper bound of Corollary 2; asymptotically, it produces rate $1 - \frac{2+\delta}{a}$, where $a \triangleq \frac{ms}{\log(n)}$. Further, the construction may only be applied when a is (approximately) an *even integer* $\geqslant 4$. The former can be remedied by replacing the union bound in the analysis of [32, Sec. VII-A] with the Lovász local lemma [33] (similarly to techniques used independently in [12] and [38]), as follows.

Lemma 9: Let $s \triangleq \lceil \log(n/m) + \log\log(n/m) + \log(3e) \rceil$. Then, using $\mathcal{C} \triangleq \Sigma^{n/m}$ in Construction P,

$$\begin{aligned}R(\mathcal{C}_{\text{pilot}}) &\geqslant \left(1 - \frac{1}{m}\right) \cdot \left(1 - \frac{\log(e)}{2\log(n/m)}\right) \\ &= 1 - \frac{1}{m} - O\left(\frac{1}{\log(n)}\right).\end{aligned}$$

Proof: Denote for a uniformly chosen $\boldsymbol{c} \in \Sigma^{n/m}$ the event $A_{i,j}$ that $\boldsymbol{c}^{(i)} = \boldsymbol{p}^{(j)}$. Clearly $p \triangleq \Pr(A_{i,j}) = q^{-s}$, and $A_{i,j}$ is jointly independent of $\{A_{i',j'} : |i - i'| \geqslant s\}$, i.e., all except $(n/m - s)(2s-1) - 1 \leqslant 2sn/m - 1$ distinct events.

For sufficiently large n, observe that

$$\begin{aligned}sq^{-s} &\leqslant \frac{\log(n/m) + \log\log(n/m) + \log(3e)}{(n/m)\log(n/m)3e} \\ &= \frac{m}{2en} \cdot \frac{2}{3}\left(1 + \frac{\log\log(n/m) + \log(3e)}{\log(n/m)}\right) < \frac{m}{2en},\end{aligned}$$

where the first inequality is justified by $(s+r)q^{-(s+r)} \leqslant sq^{-s}$ for $r \geqslant 0$ and $s \geqslant \log(e)$. Rearranging, we have $m \geqslant 2epsn$. Therefore, letting $x \triangleq \frac{ep}{1+ep}$ (hence, $\frac{x}{1-x} = ep$), and recalling for all $x \in (0,1)$ that $1 - x \geqslant \exp(\frac{-x}{1-x})$, we have

$$\begin{aligned}x(1-x)^{2sn/m-1} &= ep(1-x)^{2sn/m} \\ &\geqslant p\exp\left(1 - 2epsn/m\right) > p.\end{aligned}$$

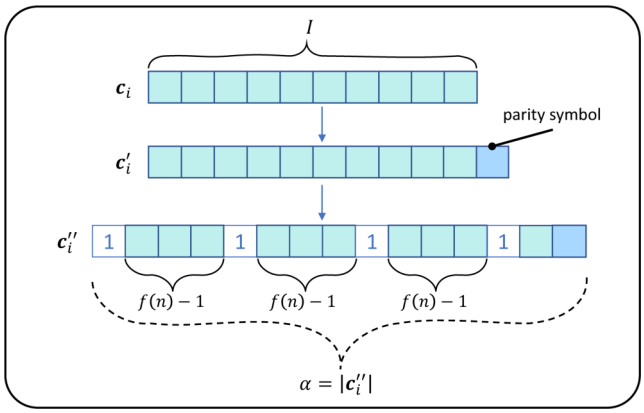

Fig. 1. Index generation (best viewed in online colored version).

It therefore follows from the local lemma that

$$\Pr(\boldsymbol{c} \perp^s \boldsymbol{p}) \geqslant (1-x)^{(n/m)^2} \geqslant \exp\left(-ep(n/m)^2\right) = e^{-e(n/m)^2 q^{-s}} \geqslant e^{-n/2sm},$$

where again we used the fact that $m \geqslant 2epsn$. That is, $|\mathcal{O}_{\boldsymbol{p}}| \geqslant q^{n/m} e^{-n/2sm} = \left(qe^{-1/2s}\right)^{n/m}$, and

$$R(\mathcal{O}_{\boldsymbol{p}}) \geqslant 1 - \frac{\log(e)}{2s}.$$

Hence, Corollary 6 concludes the proof. ■

Based on Lemma 9, Construction P achieves $1 - \frac{1}{a} - o_n(1)$ rate, where $a \triangleq \frac{ms}{\log(n)}$, asymptotically matching the bound of Corollary 2. It also expands the values of a for which the construction may be applied; however, unfortunately a is still restricted to be (approximately) an *integer* $\geqslant 2$. Moreover, encoding $\mathcal{C}_{\text{pilot}}(n)$ involves a choice of $\boldsymbol{p}$, and the authors are not aware of a straightforward way to make this choice while optimizing $R(\mathcal{O}_{\boldsymbol{p}})$; it further requires encoding into (potentially, a sub-code of) $\mathcal{O}_{\boldsymbol{p}}$, which is also, to the best of our knowledge, not readily done in an efficient manner. To bridge that gap, we present in the next section a construction based on an indexing approach, which can be applied for any $a > 1$, asymptotically matching Corollary 2 for all choices.

B. Index-Based Construction

In this section, an index-based construction of single-strand torn-paper codes is presented and is then extended for multiple strands.

It is assumed from here on out that $L_{\min} = \lceil a \log(n) \rceil$, for some $a > 1$ which is fixed throughout this section. We propose the following construction of length-n $(L_{\min}, L_{\max})$-single strand torn-paper codes. The construction is based on the following components.

Definition 10: For an integer I, let $(\boldsymbol{c}_i)_{i \in [q^I]}$, $\boldsymbol{c}_i \in \Sigma^I$ be codewords of a q-ary Gray code, in order. Denote by $\boldsymbol{c}_i'$ the concatenation of $\boldsymbol{c}_i$ with a single parity symbol (i.e., the sum of the entries in $\boldsymbol{c}_i'$ is zero). Further, denote by $\boldsymbol{c}_i''$ the result of inserting '1's into $\boldsymbol{c}_i'$ at every location divisible by $f(n)$ (since the locations of substrings start with 0, the first bit of $\boldsymbol{c}_i''$ is always '1'). The process is illustrated in Figure 1. Note that $\alpha \triangleq |\boldsymbol{c}_i''| = \left\lceil \frac{f(n)}{f(n)-1}(I+1) \right\rceil$ for all $i \in [q^I]$. We refer to $\boldsymbol{c}_i$ (or simply i) as an *index* in the construction and to $\boldsymbol{c}_i''$ as an *encoded index*.

This index structure is motivated by the property indicated in the following lemma.

Lemma 11: Let $\boldsymbol{c}$ be an α-substring of $\boldsymbol{c}_i'' \circ \boldsymbol{c}_{i+1}''$, for some $i \in [q^I - 1]$. Then i can uniquely be recovered from $\boldsymbol{c}$.

Proof: Since $\boldsymbol{c}_i''$ and $\boldsymbol{c}_{i+1}''$ differ only at the parity symbol and one additional coordinate (which corresponds to the only position where $\boldsymbol{c}_i$ and $\boldsymbol{c}_{i+1}$ differ), $\boldsymbol{c}$ is either $\boldsymbol{c}_i''$ or a copy of $\boldsymbol{c}_{i+1}''$ with an erroneous parity symbol. To obtain i it suffices to distinguish these two cases, which may be done by calculating the parity symbol of $\boldsymbol{c}''$; If the parity symbol is correct then i equals to the decoding of $\boldsymbol{c}$ (with the Gray-code decoder), and otherwise i equals to the decoding of $\boldsymbol{c}$ minus one. ■

Definition 12: Let f, N be integers. The *Run-length limited (RLL) encoder* E_N^{RLL} receives strings of length $m(N)$ and returns strings of length N that do not contain zero runs of length f. Constructions of such encoders can be taken from [20] or [37, Lem. 4].

Construction A: The main idea of the construction is that every codeword should constitute a concatenation of length-$L_{\min}$ segments with the following structure: an index, followed by a marker, then encoded data. Let $f(n)$ be any integer-valued function satisfying $f(n) = \omega(1)$ and $f(n) = o(\log(n))$ (see Theorem 18 for a choice optimizing the redundancy of this construction). Further assume $n \geqslant L_{\min} \geqslant \alpha + f(n) + 2$. Let $I \triangleq \lceil \log(n/L_{\min}) \rceil$, $K \triangleq \lfloor n/L_{\min} \rfloor - 1$ and $N \triangleq L_{\min} - \alpha - f(n) - 2$. The constructed $(L_{\min}, L_{\max})$-single strand torn-paper code, denoted by $\mathcal{C}_A(n)$, is defined by the encoder $\text{Enc}_A : \Sigma^{Km(N)} \to \Sigma^n$ in Algorithm 1, and illustrated in Figure 2. □

Algorithm 1 Encoder for Construction A

Input: $\boldsymbol{x} = (x_0, x_1, \ldots, x_{Km(N)-1}) \in \Sigma^{Km(N)}$
Output: $\text{Enc}_A(\boldsymbol{x})$
for $i \leftarrow 0$ *to* $K-1$ **do**
  $\boldsymbol{x}_i \leftarrow (x_{im(N)}, x_{im(N)+1}, \ldots, x_{(i+1)m(N)-1})$ // $|\boldsymbol{x}_i| = m(N)$
  $\boldsymbol{y}_i \leftarrow E_N^{RLL}(\boldsymbol{x}_i)$ // $\boldsymbol{y}_i$ contains no zero runs of length $f(n)$
  $\boldsymbol{z}_i \leftarrow \boldsymbol{c}_i'' \circ 10^{f(n)}1 \circ \boldsymbol{y}_i$ // $|\boldsymbol{z}_i| = L_{\min}$
end
$\boldsymbol{z}_K \leftarrow \boldsymbol{c}_K'' \circ 10^{f(n)}10^N$ // $|\boldsymbol{z}_K| = L_{\min}$
$\boldsymbol{z} \leftarrow \boldsymbol{z}_0 \circ \boldsymbol{z}_1 \circ \cdots \circ \boldsymbol{z}_K \circ 0^{n \bmod L_{\min}}$ // $|\boldsymbol{z}| = n$
return $\boldsymbol{z}$

In the rest of the paper, we call the strings $\boldsymbol{x}_i$ (respectively, $\boldsymbol{y}_i$) in the constructions an *information block* (*encoded block*); the strings $10^{f(n)}1$ are called *markers*; finally, a string $\boldsymbol{z}_i$ will simply be referred to as a segment of $\boldsymbol{z}$. Note that the last segment $\boldsymbol{z}_K$ of $\boldsymbol{z}$ deliberately does not contain data, to account for the possibility that a part of $\boldsymbol{z}_K \circ 0^{n \bmod L_{\min}}$ might be partitioned by an adversarial channel in such a way that it does not contain, at its suffix, a prefix of an index. We observe that once the encoded blocks $\boldsymbol{y}_i$'s are obtained, encoding (including the generation of the Gray code) then requires a number of operations linear in n. By [20] and [37], encoding

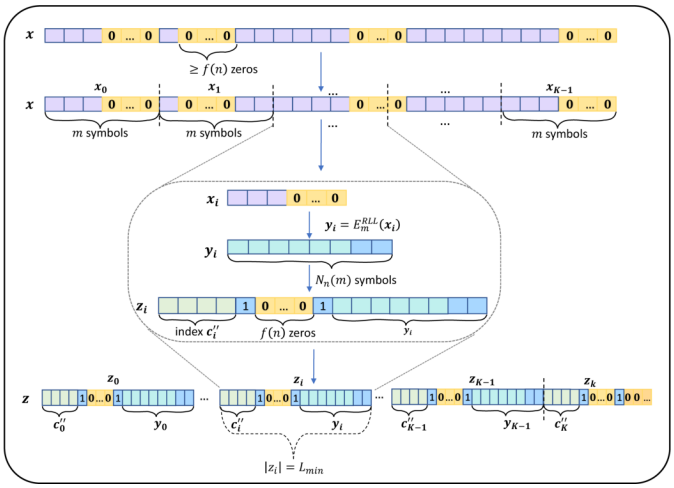

Fig. 2. Illustration of Algorithm 1 (best viewed in online colored version).

each $\boldsymbol{x}_i$ into $\boldsymbol{y}_i$ may also be achieved with a linear number of operations. Hence, the complexity of Construction A is linear with n.

Example 13: We demonstrate the operation of Enc_A. Let $q = 2, n = 45, L_{\min} = 14, f(n) = 2$. For index generation, we utilize the binary Gray code $(00, 01, 11, 10)$, whose encoded indices are (in order)

$$(101010, 101111, 111110, 111011)$$

(observe $\alpha = 6$). Let $N = L_{\min} - f(n) - 2 - \alpha = 4$, and observe an encoder E_N^{RLL} exists with $m(N) = 3$, defined by the lexicographic ordering of the $f(n)$-run-length-limited sequences of length N

$$\{0101, 0110, 0111, 1010, \\ 1011, 1101, 1110, 1111\}.$$

Noting that $K = 2$, we demonstrate, e.g., the encoding of the information sequence 001110. Observe, $\boldsymbol{x}_0 = 001, \boldsymbol{x}_1 = 110$, hence $\boldsymbol{y}_0 = 0110, \boldsymbol{y}_1 = 1110$. We then have

$$\boldsymbol{z}_0 = 101010\ 1001\ 0110$$
$$\boldsymbol{z}_1 = 101111\ 1001\ 1110$$
$$\boldsymbol{z}_2 = 111110\ 1001\ 0000,$$

and $\boldsymbol{z} = \boldsymbol{z}_0 \circ \boldsymbol{z}_1 \circ \boldsymbol{z}_2 \circ 000$.

Next, it is shown that the constructed code $\mathcal{C}_A(n)$ is an $(L_{\min}, L_{\max})$-single strand torn-paper code.

Theorem 14: For all $L_{\max} \geqslant L_{\min}$, $\mathcal{C}_A(n)$ is an $(L_{\min}, L_{\max})$-single strand torn-paper code with a linear-run-time decoder.

The proof of Theorem 14 is carried by presenting an explicit decoder to $\mathcal{C}_A(n)$ as follows. Let $\boldsymbol{z} \in \mathcal{C}_A(n)$ and let $\boldsymbol{z} = \boldsymbol{u}_0 \circ \boldsymbol{u}_1 \circ \cdots \circ \boldsymbol{u}_{s-1}$ so that $\{\!\{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{s-1}\}\!\}$ is an $(L_{\min}, L_{\max})$-segmentation of $\boldsymbol{z}$. The main task of the decoding algorithm is to successfully retrieve the location within $\boldsymbol{z}$ of each of the s segments of the $(L_{\min}, L_{\max})$-segmentation. For every segment $\boldsymbol{u}_j$, $j \in [s]$, the decoder first finds the location i such that the first (maybe partial) occurrence of an encoded index in the segment $\boldsymbol{u}_j$ is of $\boldsymbol{c}_i''$ (see below for a proof that this is possible). Given i and the location of $\boldsymbol{c}_i''$ in $\boldsymbol{u}_j$, the location of the segment $\boldsymbol{u}_j$ within $\boldsymbol{z}$ can be calculated. Then, according to the location in $\boldsymbol{z}$ for each segment in the $(L_{\min}, L_{\max})$-segmentation, one can simply concatenate the segments in the correct order to obtain the codeword $\boldsymbol{z}$. Finally, by removing the markers and the encoded indices and applying the RLL decoder for each of the strings $\boldsymbol{y}_i$'s, the information string $\boldsymbol{x}$ is retrieved.

Consider the case where a segment $\boldsymbol{u}$ is a proper substring of the suffix of $\boldsymbol{z}$ of length $(n \bmod L_{\min}) + N + f(n)$, i.e., $\boldsymbol{z}_K 0^{n \bmod L_{\min}}$ (note that this does not imply that $\boldsymbol{u}$ is itself a suffix of $\boldsymbol{z}$). Then, $\boldsymbol{u}$ does not intersect $\boldsymbol{y}_i$ for any $i \in [K]$, and may safely be discarded. We see next that these cases may be identified efficiently.

Lemma 15: Let $\boldsymbol{z} \in \mathcal{C}_A(n)$ and let $\boldsymbol{u}$ be a proper substring of $\boldsymbol{z}_K 0^{n \bmod L_{\min}}$. If n is sufficiently large (specifically, if $(a-1)\lceil \log(n) \rceil > 2\ f(n) + 1$), then this fact can efficiently be identified.

Proof: Observe that either $|\boldsymbol{u}| < L_{\min}$ or $\boldsymbol{u}$ contains a suffix of '0's of length at least

$$L_{\min} - \alpha - f(n) - 1 \geqslant (a-1)\lceil \log(n) \rceil - f(n) - 1,$$

i.e., longer than $f(n)$, which can easily be identified. ■

By Lemma 15, it is sufficient to retrieve the location of any segment which is not a substring of the suffix of length $(n \bmod L_{\min}) + N + f(n)$ of $\boldsymbol{z}$. For any such $\boldsymbol{u}$, the calculation of the index i such that $\boldsymbol{c}_i''$ is the first (perhaps partial) occurrence of an encoded index within $\boldsymbol{u}$, is given in Algorithm 2.

Algorithm 2 Index Retrieval From a Segment

Input: An L-segment $\boldsymbol{u}$ of a codeword of $\mathcal{C}_A(n)$, where $L \geqslant L_{\min}$.
Output: The index of $\boldsymbol{u}$ within $\boldsymbol{z}$, $\mathrm{Ind}(\boldsymbol{u})$
$\boldsymbol{u}' \leftarrow$ the $L_{\min}$-length prefix of $\boldsymbol{u}$
$j \leftarrow$ the starting index of the unique occurrence of $10^{f(n)}1$ within $\boldsymbol{u}'$; if none exists, of the cyclic occurrence
$\boldsymbol{c}'' \leftarrow$ the (cyclic) α-substring of $\boldsymbol{u}$ strictly preceding j
$\boldsymbol{c}' \leftarrow$ the non-padded subsequence of $\boldsymbol{c}''$
$\boldsymbol{c} \leftarrow$ the I-prefix of $\boldsymbol{c}'$
$\mathrm{Ind} \leftarrow$ the index of $\boldsymbol{c}$ in the Gray code
if *the last symbol of* $\boldsymbol{c}'$ *is not the parity of* $\boldsymbol{c}$ **then**
  $\mathrm{Ind} \leftarrow \mathrm{Ind} - 1$
end
return Ind

Any L-segment $\boldsymbol{u}$ of $\boldsymbol{z} \in \mathcal{C}_A(n)$, such that $L \geqslant L_{\min}$, contains at least part of one of the encoded indices $\boldsymbol{c}_i''$. If $\boldsymbol{c}_i''$ is the first encoded index to intersect $\boldsymbol{u}$, we denote by $\mathrm{Ind}(\boldsymbol{u}) \triangleq i$ the *index of* $\boldsymbol{u}$. Note that this index does not depend on the information that was encoded in the construction, but rather, only on the location of $\boldsymbol{u}$ in $\boldsymbol{z}$. Algorithm 2 ensures that it is possible to determine the index of every L-segment $\boldsymbol{u}$ of $\boldsymbol{z}$, where $L \geqslant L_{\min}$.

The correctness of Algorithm 2 follows from the next lemma.

Lemma 16: Let $\boldsymbol{z} \in \mathcal{C}_A(n)$, $L \geqslant L_{\min}$, and let $\boldsymbol{u}$ be an L-segment of $\boldsymbol{z}$ which is not a substring of the suffix of

length $(n \bmod L_{\min}) + N + f(n)$ of $\boldsymbol{z}$. Then, Algorithm 2 successfully returns the index $\mathrm{Ind}(\boldsymbol{u})$ of $\boldsymbol{u}$.

Proof: Let $\boldsymbol{u}$ be a substring of $\boldsymbol{z}$ and w.l.o.g. assume that $|\boldsymbol{u}| = L_{\min}$. From the RLL encoding of the strings $\boldsymbol{x}_i$'s, observe that $\boldsymbol{u}$ does not contain any occurrences of $10^{f(n)}1$ except those explicitly added to the encoded indices by Construction A. Since $|\boldsymbol{z}_j| = L_{\min}$ for all j, either $\boldsymbol{u}$ contains an occurrence of $10^{f(n)}1$ or it has a suffix-prefix pair whose concatenation is $10^{f(n)}1$ (this follows from Construction A and the assumption that $\boldsymbol{u}$ does not begin with a proper suffix of $\boldsymbol{z}_K 0^{n \bmod L_{\min}}$). In both cases, we will show that the precise location of the (perhaps incomplete) occurrence of $\boldsymbol{c}_i''$ in $\boldsymbol{u}$ can be deduced, for some i.

Let j be the (unique) location in $\boldsymbol{u}$ of the substring $10^{f(n)}1$. If $j \geqslant \alpha$, then $\boldsymbol{u}$ contains a complete occurrence of the encoded index $\boldsymbol{c}_i''$, and so the index $\boldsymbol{c}_i$, and therefore i, are readily obtained. Otherwise, $j < \alpha$ and let $\boldsymbol{c}''$ be the cyclic α-substring of $\boldsymbol{u}$ strictly preceding the substring $10^{f(n)}1$ which starts at location $L_{\min} - (\alpha - j)$. The substring $\boldsymbol{c}''$ is obtained by the concatenation of the $(\alpha - j)$-suffix of $\boldsymbol{u}$ with the j-prefix of $\boldsymbol{u}$. The proof is now concluded by Lemma 11. ■

We remark that the described procedure operates in runtime which is linear in the substring length. In addition, if $\boldsymbol{z}$ can be reconstructed from its non-overlapping substrings, then the strings $\boldsymbol{y}_i$'s are readily obtained, and $\boldsymbol{x}$ may be decoded (again, see [20] and [37]). These algorithms also require a linear number of operations. This completes the proof of Theorem 14.

Example 17: We return to Example 13, to demonstrate the operation of Theorem 14. Recall, for $q = 2, n = 45, L_{\min} = 14, f(n) = 2$, that we have constructed the following codeword

$$\boldsymbol{z} = 101010100101101011111001111011111010010000000.$$

Suppose that we receive the following $(14, 20)$-segmentation of $\boldsymbol{z}$:

$$\{\!\{10101010010110101, 1111001111011111, 010010000000\}\!\}.$$

Note since $|010010000000| = 12 < L_{\min}$, it might readily be inferred that it is the suffix of $\boldsymbol{z}$. We therefore only need identify the locations of the other two segments.

- The segment 101010 1001 0110101 contains the marker $10^{f(n)}1 = 1001$, and we therefore conclude that 101010 (given $\alpha = 6$) is an encoded-index, which as we recall corresponds to the Gray-code element $c_0 = 00$. It follows that $\boldsymbol{y}_0 = 0110$, and 101 is a prefix of $\boldsymbol{z}_1$ (observe that the following segment of $\boldsymbol{z}_1$ could not have been immediately identify, if more segments were received).
- Next, the segment 111 1001 111011111 also contains a marker, implying that 111 is the suffix of c_i'', and 111 the prefix of c_{i+1}''. Concatenating, we have the index $\boldsymbol{c}'' = 111111$ and $\boldsymbol{c}' = 11\ 1$, which is an instance of $\boldsymbol{c}_2$ containing an erroneous parity symbol. Hence we deduce $i = 1$, and $\boldsymbol{y}_1 = 1110$.

Together, the decoding $\boldsymbol{x}_0 = 001$ and $\boldsymbol{x}_1 = 110$ may now be performed, reconstructing the original information sequence.

Lastly the redundancy of Construction A is analyzed.

Theorem 18: Using the RLL encoders of [20] and [37] in Construction A, it holds that

$$\begin{aligned}\mathrm{red}(\mathcal{C}_A(n)) &\leqslant \frac{n}{a}\Bigg(1 + \frac{f(n)}{\log(n)} + \frac{1}{f(n)-1} + \\ &\qquad \frac{9 + 2/(f(n)-1)}{\log(n)} + \frac{4a}{q^{f(n)}} + \frac{2a^2+2}{n}\Bigg) \\ &= \frac{n}{a}\left(1 + (1+o(1))\left(\frac{f(n)}{\log(n)} + \frac{1}{f(n)}\right)\right).\end{aligned}$$

In particular, the redundancy is optimized for $f(n) = (1 + o(1))\sqrt{\log(n)}$, i.e.,

$$\mathrm{red}(\mathcal{C}_A(n)) \leqslant \frac{n}{a}\left(1 + \frac{2 + o(1)}{\sqrt{\log(n)}}\right).$$

Proof: From Construction A, observe that $\mathrm{red}(\mathcal{C}_A(n)) = (n \bmod L_{\min}) + L_{\min} + K(L_{\min} - m(N))$ and

$$\begin{aligned}L_{\min} - N &= \alpha + f(n) + 2 \\ &= \left\lceil \frac{f(n)}{f(n)-1}(I+1)\right\rceil + f(n) + 2 \\ &\leqslant \frac{f(n)}{f(n)-1}(I+1) + f(n) + 3 \\ &\leqslant \frac{f(n)}{f(n)-1}(\log(n/L_{\min}) + 2) + f(n) + 3 \\ &\leqslant \log(n) + f(n) + \frac{\log(n)}{f(n)-1} + 5 + \frac{2}{f(n)-1}.\end{aligned}$$

Further, by [37, Lem. 4] one may efficiently encode $\boldsymbol{x} \mapsto \boldsymbol{y}$ such that $N - m(N) \leqslant \left\lceil \frac{q}{q-2} \cdot \frac{N}{q^{f(n)}} \right\rceil$ (For $q = 2$ [20, Sec. III] showed $N - m(N) \leqslant 2\lceil N/q^{f(n)-1}\rceil$), and we shall use the overly zealous upper bound $N - m(N) \leqslant \frac{4N}{q^{f(n)}} + 2 \leqslant \frac{4a\log(n)}{q^{f(n)}} + \frac{4}{q} + 2 \leqslant \frac{4a\log(n)}{q^{f(n)}} + 4$.

Finally, we get that

$$\begin{aligned}\mathrm{red}(\mathcal{C}_A(n)) &= K(L_{\min} - m(N)) + L_{\min} + (n \bmod L_{\min}) \\ &\leqslant \frac{n}{a\log(n)}\Bigg(\log(n) + f(n) + \frac{\log(n)}{f(n)-1} + \\ &\qquad 9 + \frac{2}{f(n)-1} + \frac{4a\log(n)}{q^{f(n)}}\Bigg) + 2L_{\min} \\ &\leqslant \frac{n}{a}\Bigg(1 + \frac{f(n)}{\log(n)} + \frac{1}{f(n)-1} + \\ &\qquad \frac{9 + 2/(f(n)-1)}{\log(n)} + \frac{4a}{q^{f(n)}} + \frac{2a^2+2}{n}\Bigg),\end{aligned}$$

which completes the proof of the first part. The second part follows by substitution of $f(n) = (1 + o(1))\sqrt{\log(n)}$ into the former. ■

By Theorem 18 and Corollary 2, efficient encoding and decoding is possible at asymptotically optimal rates. In comparison to Construction P (by Lemma 9), Construction A asymptotically achieves rate $1 - \frac{1}{a} - O\left(\frac{f(n)}{\log(n)} + \frac{1}{f(n)}\right)$ instead of $1 - \frac{1}{\lceil a \rceil} - O\left(\frac{1}{\log(n)}\right)$, for any channel parameter $a > 1$ (here, the integer value is used since Construction P must be operated at $m \triangleq \lceil a \rceil$ to produce an $(L_{\min}, L_{\max})$-torn-paper code). For completeness, we also include specific construction parameters

TABLE I

CONSTRUCTION P (LEMMA 8): SPECIFIC PARAMETERS $(m, s, R(\mathcal{C}_{\text{pilot}}(n)))$

$L_{\min}\backslash n$	60	250	4000	60,000	400,000	6,000,000
10	2, 5, 0.379	n/a	n/a	n/a	n/a	n/a
50	15, 3, **0.856**	10, 5, **0.844**	5, 10, **0.798**	3, 16, 0.667	2, 25, 0.5	2, 25, 0.5
100	n/a	10, 10, **0.9**	10, 10, **0.9**	6, 16, 0.833	5, 20, 0.8	4, 25, 0.75
300	n/a	n/a	32, 9, **0.968**	25, 12, **0.96**	20, 15, **0.95**	15, 20, **0.933**
1000	n/a	n/a	125, 8, **0.992**	100, 10, **0.989**	64, 15, **0.984**	50, 20, **0.98**

(Bold-face indicates Lemma 8 provides highest lower-bound on rate.)

TABLE II

CONSTRUCTION P (LEMMA 9): SPECIFIC PARAMETERS $(m, s, R(\mathcal{C}_{\text{pilot}}(n)))$

$L_{\min}\backslash n$	60	250	4000	60,000	400,000	6,000,000
10	2, 5, **0.45**	n/a	n/a	n/a	n/a	n/a
50	15, 3, 0.778	10, 5, 0.81	5, 10, 0.76	5, 10, **0.76**	4, 12, **0.719**	3, 16, **0.646**
100	n/a	10, 10, 0.855	10, 10, 0.855	10, 10, **0.855**	8, 12, 0.839	6, 16, 0.807
300	n/a	n/a	25, 12, 0.92	25, 12, 0.92	25, 12, 0.92	20, 15, 0.918
1000	n/a	n/a	50, 20, 0.956	50, 20, 0.956	50, 20, 0.956	50, 20, 0.956

(Bold-face indicates Lemma 9 provides highest lower-bound on rate. Background pattern indicates that the choice of m, s is only guaranteed by Lemma 9.)

TABLE III

CONSTRUCTION A (THEOREM 18): SPECIFIC PARAMETERS $(f, I, N, K, R(\mathcal{C}_A(n)))$

$L_{\min}\backslash n$	60	250	4000	60,000	400,000	6,000,000
10	n/a	n/a	n/a	n/a	n/a	n/a
50	n/a	2, 2, 40, 4, 0.56	3, 4, 38, 79, 0.711	3, 6, 35, 1199, 0.659	4, 7, 34, 7999, 0.66	4, 9, 31, 119999, 0.6
100	n/a	2, 1, 92, 1, 0.32	3, 3, 89, 39, 0.839	3, 5, 86, 599, 0.829	4, 6, 85, 3999, **0.84**	4, 8, 82, 59999, **0.81**
300	n/a	n/a	3, 2, 291, 12, 0.843	3, 4, 288, 199, 0.925	4, 6, 9, 285, 1332, 0.939	4, 8, 282, 19999, 0.93
1000	n/a	n/a	3, 1, 992, 3, 0.721	3, 3, 989, 59, 0.942	4, 5, 986, 399, 0.976	4, 7, 984, 5999, 0.976

(Bold-face indicates Theorem 18 provides highest lower-bound on rate.)

for several arbitrary choices of $n, L_{\min}$, and compare resulting rates, in Tables I to III (all for $q = 4$). It should however be stressed that, for Construction P, the choice of s, m optimizing the resulting rate $R(\mathcal{C}_{\text{pilot}}(n)) \geqslant \left(1 - \frac{1}{m}\right) \cdot R(\mathcal{O}_{\boldsymbol{p}})$ is not straightforward, even given the lower bounds of Lemmas 8 and 9; indeed, $R(\mathcal{O}_{\boldsymbol{p}})$ cannot easily be computed, for an optimal choice of $\boldsymbol{p}$. We rely in our comparison on the lower-bounds of Lemmas 8 and 9 instead; note in particular that even for the same choice of n, m, s, i.e., for a specific code, these might provide distinct lower-bounds on the rate. As mentioned above, even then it is not immediately clear how to efficiently encode and decode $\mathcal{C}_{\text{pilot}}(n)$.

Next, we consider the case of $k > 1$ and $\log(k) = o(n)$. We know from Corollary 2 that if $\limsup \frac{L_{\min}}{nk} \leqslant 1$ then any family of $(L_{\min}, L_{\max})$-multistrand torn-paper codes will only achieve vanishing asymptotic rate; hence we assume $L_{\min} = \lceil a \log(nk) \rceil$ for some $a > 1$. The following theorem summarizes our main results regarding $(L_{\min}, L_{\max})$-multistrand torn-paper codes.

Theorem 19: Take n, k such that $k > 1$, $\log(k) = o(n)$, and let $L_{\min} = \lceil a \log(nk) \rceil$, for $a > 1$. There exists a linear run-time (in the substrings length, i.e., nk) encoder-decoder pair for $(L_{\min}, L_{\max})$-multistrand torn-paper codes achieving $1 - \frac{1}{a} - o_{nk}(1)$ asymptotic rate.

Proof: Theorem 19 is justified by a simple amendment of Construction A. We encode $\boldsymbol{x} \in \Sigma^{kKm}$ into $\{\!\{\boldsymbol{z}^{(j)} : j \in [k]\}\!\}$, where $\left|\boldsymbol{z}^{(j)}\right| = n$ for all $j \in [k]$, as follows. We modify $I \triangleq \lceil \log(k \lceil n/L_{\min} \rceil) \rceil$ (recall, also, $\alpha \triangleq \left\lceil \frac{f(n)}{f(n)-1}(I+1) \right\rceil$) and $L_{\min} = \lceil a \log(nk) \rceil$. We then denote $\boldsymbol{x} = \boldsymbol{x}^{(0)} \circ \boldsymbol{x}^{(1)} \circ \cdots \circ \boldsymbol{x}^{(k-1)}$, where $\left|\boldsymbol{x}^{(j)}\right| = Km$ for all $j \in [k]$, and apply Algorithm 1 to $(\boldsymbol{x}^{(j)})_{j \in [k]}$ in succession; observe that every operation requires only $\lceil n/L_{\min} \rceil$ distinct indices, and we utilize available indices in order throughout the k operations.

We observe that the proofs of Lemmas 15 and 16 hold without change, hence this amendment encodes into an $(L_{\min}, L_{\max})$-multistrand torn-paper code, which we denote $\mathcal{C}_A(n, k) \in \mathcal{X}_{n,k}$. Finally, following the proof of Theorem 18 we have

$$\begin{aligned}\operatorname{red}(\mathcal{C}_A(n,k)) &= k\big(K(L_{\min} - m(N)) + L_{\min} + \\ &\qquad (n \bmod L_{\min})\big) \\ &\leqslant \frac{nk}{a}\left(1 + (1+o(1))\left(\frac{f(nk)}{\log(nk)} + \frac{1}{f(nk)}\right)\right);\end{aligned}$$

As in Theorem 18, for $f(n) = (1+o(1))\sqrt{\log(nk)}$ we have

$$\operatorname{red}(\mathcal{C}_A(n,k)) \leqslant \frac{nk}{a}\left(1 + \frac{2+o(1)}{\sqrt{\log(nk)}}\right).$$

From Lemma 1 we have $\log|\mathcal{X}_{n,k}| \geqslant (n - \log(k))k$, concluding the proof. ■

Again, by Corollary 2 and Theorem 18 the rate of the construction is asymptotically optimal.

IV. ERROR-CORRECTING TORN-PAPER CODES

In this section, we extend the study of torn-paper codes to a noisy setup. We consider two models of noise. The first one assumes that the encoded string, before segmentation, suffers at most some t substitution errors. The second model corresponds to the case where some of the segments are deleted during segmentation.

A. Substitution-Correcting Torn-Paper Codes

For a string $\boldsymbol{x}$, its *t-error torn-paper ball*, denoted by $\mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x};t)$, is defined as the set of all possible $(L_{\min}, L_{\max})$-segmentations after introducing at most t errors to $\boldsymbol{x}$, that is,

$$\mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x};t) \triangleq \bigcup_{\boldsymbol{y}\in B_t(\boldsymbol{x})} \mathcal{T}_{L_{\min}}^{L_{\max}}(\boldsymbol{y}),$$

where $B_t(\boldsymbol{x}) = \{\boldsymbol{y} : d_H(\boldsymbol{x},\boldsymbol{y}) \leqslant t\}$ is the radius-t Hamming ball centered at $\boldsymbol{x}$. A code $\mathcal{C}$ is called a *t-error single-strand torn-paper code* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$, $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$, it holds that

$$\mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x}_1;t) \cap \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x}_2;t) = \emptyset.$$

Our goal in this section is to show how to adjust Construction A in order to produce t-error single-strand torn-paper codes. We first explain the main ideas of the required modifications. Let $\boldsymbol{z} = \mathrm{Enc}_A(\boldsymbol{x}) \in \mathcal{C}_A(n)$ (encoded with Algorithm 1) and let $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$ be an $(L_{\min}, L_{\max})$-segmentation of some word $\boldsymbol{z}'$, where $d_H(\boldsymbol{z},\boldsymbol{z}') \leqslant t$. The main task of the noiseless decoder of $\mathcal{C}_A(n)$ was to first calculate the index, and thus the location in $\boldsymbol{z}$, of every segment $\boldsymbol{u} \in \mathcal{U}$. However, in the presence of errors, calculating the index of a segment $\boldsymbol{u} \in \mathcal{U}$ based on the first (perhaps partial) occurrence of an encoded index within $\boldsymbol{u}$ might result with the misplacement of all the (perhaps partial) information blocks $\boldsymbol{y}_i$ that are contained in $\boldsymbol{u}$. Hence, a more careful approach is necessary for index decoding.

Before presenting our construction for t-error single-strand torn-paper codes, we introduce several additional required definitions. For a string $\boldsymbol{u}$, define $\mathcal{T}_{L_{\min}}^{+}(\boldsymbol{u})$ to be the multiset of non-overlapping $L_{\min}$-segments of $\boldsymbol{u}$, where the last segment is of length ℓ, $L_{\min} \leqslant \ell < 2L_{\min}$. A segment $\boldsymbol{w} \in \mathcal{T}_{L_{\min}}^{+}(\boldsymbol{u})$ is called *A-decodable* if, informally, Algorithm 2 returns (perhaps erroneous) output when given $\boldsymbol{w}$ as input. More formally, if $\boldsymbol{w}$ satisfies one of the following conditions.

1) $\boldsymbol{w}$ either contains a unique complete occurrence of $10^{f(n)}1$, or it doesn't contain complete occurrences but contains a cyclic occurrence (if $L_{\min} < |\boldsymbol{w}| < 2L_{\min}$, require instead that either the $L_{\min}$-prefix or the $L_{\min}$-suffix of $\boldsymbol{w}$ contain a cyclic occurrence).
2) $\boldsymbol{w}$ contains precisely two complete occurrences of $10^{f(n)}1$, and there exist a unique pair of occurrences (either complete or complete-to-suffix/prefix) whose locations are at distance precisely $L_{\min}$. Recall that $\boldsymbol{w}$ cannot contain more than two complete occurrences of $10^{f(n)}1$, except in the presence of errors, hence those cases can safely be discarded (see part 1 in the proof of Theorem 20 for a formal proof).

Let $\boldsymbol{w}$ be an A-decodable segment. Then, by definition, there is at least one occurrence (perhaps cyclic) of $10^{f(n)}1$ within $\boldsymbol{w}$ and, if there is more than a single occurrence, then there is exactly one pair of occurrences such that the difference between their locations is $L_{\min}$. Consider the α-segments of $\boldsymbol{w}$ preceding these occurrences as encoded indices; if the (first) occurrence of $10^{f(n)}1$ in $\boldsymbol{w}$ is at location $\ell < \alpha$, concatenate the $(\alpha - \ell)$-segment of $\boldsymbol{w}$ at location $L_{\min} + \ell - \alpha$, to the ℓ-prefix of $\boldsymbol{w}$, and consider the resulting length-α string to be a *cyclic encoded index*.

An A-decodable segment $\boldsymbol{w}$ is called *valid* if, informally, there appears at least one 'valid' encoded index in $\boldsymbol{w}$, and no conflicting pair of such indices. More formally, a valid segment $\boldsymbol{w}$ is an A-decodable segment that satisfies one of the following conditions:

1) $\boldsymbol{w}$ contains no complete encoded index, hence it contains only a cyclic encoded index.
2) $\boldsymbol{w}$ contains a single complete encoded index, and its parity symbol is correct.
3) $\boldsymbol{w}$ contains two complete encoded indices, and either exactly one of their parity symbols is correct, or both are correct and the indices are consecutive in the applied Gray code.

Construction B: We construct a concatenated code, using Construction A as inner-code, and an arbitrary $(K, q^{m(N)M}, 2t+1)_{q^{m(N)}}$ error-correcting code $\mathcal{C}_{\mathrm{EC}}$, with an encoding algorithm $\mathrm{Enc}_{\mathrm{EC}}\colon (\Sigma^{m(N)})^M \to (\Sigma^{m(N)})^K$, as outer-code (here, K, N are the parameters of Construction A). The resulting t-error $(L_{\min}, L_{\max})$-single-strand torn-paper code is denoted $\mathcal{C}_B(n)$, with the associated encoder $\mathrm{Enc}_B : \Sigma^{m(N)M} \to \Sigma^n$. □

We observe the following property of Construction B. Assume one retrieves a noisy version $\boldsymbol{z}'$ of $\boldsymbol{z} = \mathrm{Enc}_B(\boldsymbol{x})$, e.g., from any reconstruction algorithm; further assume that $\boldsymbol{z}, \boldsymbol{z}'$ agree on all locations containing encoded indices $\boldsymbol{c}_i''$ or markers $10^{f(n)}1$ (as their locations in $\boldsymbol{z}$ are known a priori and do not depend on the information $\boldsymbol{x}$). Thus, one extracts from $\boldsymbol{z}'$ (perhaps erroneous) encoded information blocks, denoted $\boldsymbol{y}_i'$. Denote by e the number of encoded information blocks that were not recovered (e.g., due to conflicts in the reconstruction algorithm), and by s the number of encoded blocks that were recovered incorrectly (i.e., $\boldsymbol{y}_i' \neq \boldsymbol{y}_i$). Since the information string $(\boldsymbol{x}_i)_{i\in[M]} \in \Sigma^{m(N)M}$ is encoded using a $(K, q^{m(N)M}, 2t+1)_{q^{m(N)}}$ error-correcting code, it suffices that $2s + e \leqslant 2t$ to guarantee correct decoding.

In order to reconstruct a noisy version $\boldsymbol{z}'$ of $\boldsymbol{z}$, we define a modification of Algorithm 2, as follows. First, given an $(L_{\min}, L_{\max})$-segmentation $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$ we apply the reconstruction algorithm not directly to $\mathcal{U}$, but rather to valid segments in

$$\mathcal{T}_{L_{\min}}^{+}(\mathcal{U}) \triangleq \{\!\{\mathcal{T}_{L_{\min}}^{+}(\boldsymbol{u}) : \boldsymbol{u} \in \mathcal{U}\}\!\}.$$

Secondly, in case a valid $\boldsymbol{w} \in \mathcal{T}_{L_{\min}}^{+}(\mathcal{U})$ contains multiple (perhaps cyclic) occurrences of an encoded index, the algorithm selects one to decode by prioritizing complete occurrences over cyclic ones, and in the case of complete occurrences, accepting the first containing a correct parity symbol (since $\boldsymbol{w}$ is valid, such occurrence exists in this case). Decoding of the selected encoded index is then performed as described in Algorithm 2, and denoted by $\mathrm{Ind}'(\boldsymbol{w})$.

For an $(L_{\min}, L_{\max})$-segmentation $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$, we define the set

$$\mathcal{Z}(\mathcal{U}) \triangleq \big\{(\mathrm{Ind}'(\boldsymbol{w}), \boldsymbol{w}) : \boldsymbol{w} \in \mathcal{T}_{L_{\min}}^{+}(\mathcal{U}) \text{ is valid}\big\}.$$

If $(j, \boldsymbol{w}), (j, \boldsymbol{w}') \in \mathcal{Z}(\mathcal{U})$ for some j and $\boldsymbol{w} \neq \boldsymbol{w}'$, we define a restriction $\mathcal{Z}'(\mathcal{U})$ of $\mathcal{Z}(\mathcal{U})$ by including only the shortest, lexicographically-least, segment (i.e., $\mathcal{Z}'(\mathcal{U})$ defines a proper function).

Given the set $\mathcal{Z}'(\mathcal{U})$ we decode a string $\boldsymbol{z}'$ as follows.

1) Fill the encoded indices and the markers in $\boldsymbol{z}'$ in the correct locations as defined in Algorithm 1 (note again that these locations do not depend on the information).
2) Next, we iterate over any pair $(\mathrm{Ind}'(\boldsymbol{w}), \boldsymbol{w}) \in \mathcal{Z}'(\mathcal{U})$ and update $\boldsymbol{z}'$ with the symbols of the encoded blocks $\boldsymbol{y}_i$'s within $\boldsymbol{w}$; If there is a collision of symbols in the same position within an encoded block $\boldsymbol{y}'_i$ for some i, $i \in [K]$, we erase $\boldsymbol{y}'_i$ completely from $\boldsymbol{z}'$.
3) If an encoded block $\boldsymbol{y}'_i$ is partially filled at the end of the process (i.e., there are missing symbols within $\boldsymbol{y}'_i$) we erase the encoded block $\boldsymbol{y}'_i$.

The output $\boldsymbol{z}'$ of this decoding procedure over the segmentation $\mathcal{U} \in \mathcal{BT}^{L_{\max}}_{L_{\min}}(\boldsymbol{z}; t)$ is denoted by $\mathrm{Dec}_B(\mathcal{U}) \triangleq \boldsymbol{z}'$.

We now prove that $\mathcal{C}_B(n)$ is a t-error single-strand torn-paper code.

Theorem 20: Let $\boldsymbol{z} = \mathrm{Enc}_B(\boldsymbol{x})$, $\mathcal{U} \in \mathcal{BT}^{L_{\max}}_{L_{\min}}(\boldsymbol{z}; t)$, and let $\boldsymbol{z}' = \mathrm{Dec}_B(\mathcal{U})$ be the noisy version of $\boldsymbol{z}$ reconstructed by the aforementioned algorithm. Then, it holds that $2s + e \leqslant 2t$, where e, s are defined as previously explained; i.e., any inner-channel error propagates as, at most, either one outer-channel error or two outer-channel erasures.

Proof: By definition, $\mathcal{U}$ is obtained by first introducing up to t errors to $\boldsymbol{z}$, and then performing an $(L_{\min}, L_{\max})$-segmentation to the obtained word. For the rest of the proof, we fix an arbitrary $(L_{\min}, L_{\max})$-segmentation pattern, and for $\boldsymbol{z} \in \Sigma^n$ we denote $\mathcal{U} \in \mathcal{T}^{L_{\max}}_{L_{\min}}(\boldsymbol{z})$ obtained from this pattern by $\mathcal{U} = T(\boldsymbol{z})$. In particular, observe for $\|\boldsymbol{v}\| \leqslant t$ that $T(\boldsymbol{z} + \boldsymbol{v}) \in \mathcal{BT}^{L_{\max}}_{L_{\min}}(\boldsymbol{z}; t)$.

For convenience, we denote by $\boldsymbol{z}'_{\boldsymbol{v}} \triangleq \mathrm{Dec}_B(T(\boldsymbol{z} + \boldsymbol{v}))$, and by $e_{\boldsymbol{v}}$ (respectively, $s_{\boldsymbol{v}}$) the number of encoded information blocks $\boldsymbol{y}'_i$ in $\boldsymbol{z}'_{\boldsymbol{v}}$ which were not recovered (recovered erroneously). We shall prove the following proposition, which justifies the claim. Let $\boldsymbol{z} \triangleq \mathrm{Enc}_B(\boldsymbol{x})$, $\boldsymbol{v} \in \Sigma^n$ such that $\|\boldsymbol{v}\| \leqslant t$. Then $e_{\boldsymbol{v}} + 2s_{\boldsymbol{v}} \leqslant 2\|\boldsymbol{v}\|$.

The proof is done by induction on $\|\boldsymbol{v}\|$. First observe by Lemma 16 that $e_{\boldsymbol{0}} = s_{\boldsymbol{0}} = 0$ (here, $\boldsymbol{0}$ is the all-zero string). For the induction step, assume that the claim holds for any $\boldsymbol{v}' \in \Sigma^n$, $\|\boldsymbol{v}'\| < t$. Let $\boldsymbol{v} \in \Sigma^n$, $\|\boldsymbol{v}\| = t$. Take any $\boldsymbol{u}' \in \mathcal{T}^{+}_{L_{\min}}(T(\boldsymbol{z} + \boldsymbol{v}))$ affected by $t' > 0$ errors. Decompose $\boldsymbol{v} = \boldsymbol{v}' + \boldsymbol{v}''$ such that $\|\boldsymbol{v}'\| = t'$, $\|\boldsymbol{v}''\| = t - t'$, and $\boldsymbol{u}'$ contains the support of $\boldsymbol{v}'$. Consider the decoder output $\boldsymbol{z}'_{\boldsymbol{v}''}$; by the induction assumption,

$$e_{\boldsymbol{v}''} + 2\ s_{\boldsymbol{v}''} \leqslant 2(t - t').$$

We denote by $\boldsymbol{u}$ the segment corresponding to $\boldsymbol{u}'$ in $\boldsymbol{z}'_{\boldsymbol{v}''}$. Note that $\boldsymbol{u}$ contains no errors and its index is recovered correctly by the decoder. Hence, each encoded block that intersects $\boldsymbol{u}$ is either correct in $\boldsymbol{z}'_{\boldsymbol{v}''}$, or it is erased due to errors in other segments.

Denoting by δ the number of encoded information blocks intersecting $\boldsymbol{u}$, we let (i) ρ_1 be the number of those recovered correctly in $\boldsymbol{z}'_{\boldsymbol{v}''}$; (ii) ρ_2 be the number of those erased in $\boldsymbol{z}'_{\boldsymbol{v}''}$ due to collisions resulting from incorrect index-decoding in other segments; and (iii) ρ_3 be the number of those erased in $\boldsymbol{z}'_{\boldsymbol{v}''}$ due to erasures of other, intersecting, segments. Observe that $\delta = \rho_1 + \rho_2 + \rho_3 \in \{1, 2, 3\}$, depending on $|\boldsymbol{u}|$ and its location).

The rest of the proof is done by cases.

1) If $\boldsymbol{u}'$ is not valid, then all encoded information blocks intersecting $\boldsymbol{u}'$ are erased at the decoder. Hence, the ρ_1 correctly recovered blocks in $\boldsymbol{z}'_{\boldsymbol{v}''}$ which intersect $\boldsymbol{u}$ are erased in $\boldsymbol{z}'_{\boldsymbol{v}}$.
   In addition, each of the ρ_2 blocks corresponding to blocks intersecting $\boldsymbol{u}$ which are erased due to collisions, might instead cause incorrect recovery of encoded information blocks in $\boldsymbol{u}'$. Hence, $e_{\boldsymbol{v}} \leqslant e_{\boldsymbol{v}''} + \rho_1 - \rho_2$ and $s_{\boldsymbol{v}} \leqslant s_{\boldsymbol{v}''} + \rho_2$, and we note
   $$\begin{aligned} e_{\boldsymbol{v}} + 2\ s_{\boldsymbol{v}} &\leqslant (e_{\boldsymbol{v}''} + 2\ s_{\boldsymbol{v}''}) + \rho_1 + \rho_2 \\ &\leqslant 2(t - t') + \delta = 2\,\|u\| - (2t' - \delta). \end{aligned}$$
   Since $t' \geqslant 1$, we have $e_{\boldsymbol{v}} + 2\ s_{\boldsymbol{v}} \leqslant 2\,\|u\|$ unless $\delta = 3$; however, in that case $\boldsymbol{u}$ contains two complete instances of $10^{f(n)}1$ whose locations are at distance $L_{\min}$, both preceded by complete occurrences of encoded indices, and since $\boldsymbol{u}'$ is not valid we have $t' > 1$, which also concludes the proof.
2) If $\boldsymbol{u}'$ is valid but its index is incorrectly decoded, then the ρ_1 encoded information blocks that are recovered correctly in $\boldsymbol{z}'_{\boldsymbol{v}''}$ are erased in $\boldsymbol{z}'_{\boldsymbol{v}}$, and ρ_2 encoded information blocks, corresponding to those intersecting $\boldsymbol{u}$ which are erased in $\boldsymbol{z}'_{\boldsymbol{v}''}$ due to collisions, might be recovered incorrectly in $\boldsymbol{z}'_{\boldsymbol{v}}$.
   Furthermore, the placement of $\boldsymbol{u}'$ at an incorrectly decoded location causes δ additional encoded information blocks to be either erased (due to collisions) or incorrectly recovered (where the correct blocks appear in invalid segments, i.e., are erased in $\boldsymbol{z}'_{\boldsymbol{v}''}$). Denoting the number of blocks of the former type by δ_1, and the latter δ_2, we then have $e_{\boldsymbol{v}} = e_{\boldsymbol{v}''} + \rho_1 - \rho_2 + \delta_1 - \delta_2$ and $s_{\boldsymbol{v}} \leqslant s_{\boldsymbol{v}''} + \rho_2 + \delta_2$. Hence,
   $$\begin{aligned} e_{\boldsymbol{v}} + 2\ s_{\boldsymbol{v}} &\leqslant (e_{\boldsymbol{v}''} + 2\ s_{\boldsymbol{v}''}) + \rho_1 + \rho_2 + \delta_1 + \delta_2 \\ &\leqslant 2(t - t') + 2\delta = 2\,\|\boldsymbol{v}\| - 2(t' - \delta). \end{aligned}$$
   To conclude, we require $t' \geqslant \delta$. Indeed, observe that if $\delta = 2$ then $\boldsymbol{u}$ contains a complete occurrence of an encoded index followed by $10^{f(n)}1$, requiring $t' \geqslant 2$ for incorrect recovery. Likewise, if $\delta = 3$ then $\boldsymbol{u}$ contains two complete occurrences of encoded indices whose locations are at distance $L_{\min}$, each followed by $10^{f(n)}1$; incorrect recovery of the index therefore requires at least two errors in one of them in addition to further errors in the other index or $10^{f(n)}1$ marker, or an error in each $10^{f(n)}1$ marker in addition to further errors to generate such a marker at an alternative location, hence $t' \geqslant 3$ as well.
3) Finally, if the index of $\boldsymbol{u}'$ is decoded correctly (and, in particular, $\boldsymbol{u}'$ is valid), then recalling that the index of $\boldsymbol{u}$ is also decoded correctly, we clearly have $e_{\boldsymbol{v}} = e_{\boldsymbol{v}''}$. Since any error in $\boldsymbol{u}'$ can cause an error in at most a single encoded information block, we have that $s_{\boldsymbol{v}} \leqslant s_{\boldsymbol{v}''} + t'$. Hence,
   $$\begin{aligned} e_{\boldsymbol{v}} + 2s_{\boldsymbol{v}} &\leqslant e_{\boldsymbol{v}''} + 2(s_{\boldsymbol{v}''} + t') \\ &= (e_{\boldsymbol{v}''} + 2\ s_{\boldsymbol{v}''}) + 2\ t' \leqslant 2\ t = 2\,\|\boldsymbol{v}\|\,. \end{aligned}$$

■

Theorem 21: Denote the redundancy of the outer-code $\mathcal{C}_{EC}$ used in Construction B by $\rho_{EC} \triangleq K - M$. Then, operating Enc_A as in Theorem 18, with $f(n) = (1+o(1))\sqrt{\log(n)}$, we have

$$\begin{aligned}\mathrm{red}(\mathcal{C}_B(n)) \leqslant \frac{n}{a}\Bigg(1 + \frac{f(n)}{\log(n)} + \frac{1}{f(n)-1} + \\ \frac{9+2/(f(n)-1)}{\log(n)} + \frac{4a}{q^{f(n)}} + \frac{2a^2+2}{n}\Bigg) + \\ \rho_{EC}\Bigg((a-1)\log(n) - 2\sqrt{\log(n)} - \\ 11 - \frac{3}{\sqrt{\log(n)}-1} - \frac{4a\log(n)}{q^{\sqrt{\log(n)}}}\Bigg) \\ = \frac{n}{a}\left(1 + \frac{2+o(1)}{\sqrt{\log(n)}}\right) + \\ \rho_{EC}\Big((a-1)\log(n) - (2+o(1))\sqrt{\log(n)}\Big).\end{aligned}$$

Furthermore, when $a > 2$ then the outer-code $\mathcal{C}_{EC}$ can be an MDS code and hence $\rho_{EC} = 2t$.

Proof: By Construction B, $\mathrm{red}(\mathcal{C}_B(n)) = \mathrm{red}(\mathcal{C}_A(n)) + \rho_{EC} \cdot m(N)$.

We recall from the proof of Theorem 18 that for $f(n) = \lceil\sqrt{\log(n)}\rceil$ it holds that

$$\begin{aligned}m(N) \geqslant L_{\min} - \log(n) - f(n) - \frac{\log(n)}{f(n)-1} - \\ 9 - \frac{2}{f(n)-1} - \frac{4a\log(n)}{q^{f(n)}} \\ \geqslant (a-1)\log(n) - 2\sqrt{\log(n)} - \\ 11 - \frac{3}{\sqrt{\log(n)}-1} - \frac{4a\log(n)}{q^{\sqrt{\log(n)}}},\end{aligned}$$

satisfying the former part of claim.

Next, for $a > 2$ we observe that $m(N) > \log(n) - \log\log(n) + O_n(1) = \log(K)$, implying that an RS code may be used in Construction B, satisfying the latter part. ■

Before concluding the section, we outline an extension of Construction B to the case $k > 1$, i.e., to *t-error multi-strand torn-paper codes*.

Corollary 22: Take n, k such that $k > 1$, $\log(k) = o(n)$; let $L_{\min} = \lceil a\log(nk)\rceil$, for $a > 1$, and take some $L_{\max} \geqslant L_{\min}$. Amend Construction B as was done in Theorem 19 to Construction A, using a $(kK, q^{m(N)M}, 2t+1)_{q^{m(N)}}$ error-correcting code $\mathcal{C}_{\mathrm{EC}}$, with redundancy $\rho_{EC} \triangleq kK - M$. Then the resulting code $\mathcal{C}_B(n,k)$ is a t-error $(L_{\min}, L_{\max})$-multistrand torn-paper code, satisfying

$$\begin{aligned}\mathrm{red}(\mathcal{C}_B(n,k)) \leqslant \frac{nk}{a}\left(1 + \frac{2+o(1)}{\sqrt{\log(nk)}}\right) + \\ \rho_{EC}\Big((a-1)\log(nk) - \\ (2+o(1))\sqrt{\log(nk)}\Big).\end{aligned}$$

Proof: The proof of Theorem 20 applies without change. As in Theorem 19, we have

$$m(N) = (a-1)\log(nk) - (1+o(1))\left(f(nk) + \frac{\log(nk)}{f(nk)}\right),$$

and following the steps of Theorem 21, we have the claimed upper bound on redundancy, for $f(n) = (1+o(1))\sqrt{\log(nk)}$. ■

B. Deletion-Correcting Torn-Paper Codes

For a string $\boldsymbol{x}$, its *t-deletion torn-paper ball*, $\mathcal{DT}^{L_{\max}}_{L_{\min}}(\boldsymbol{x};t)$, is defined as all the subsets with at most t missing segments of all the possible $(L_{\min}, L_{\max})$-segmentations of $\boldsymbol{x}$, that is,

$$\mathcal{DT}^{L_{\max}}_{L_{\min}}(\boldsymbol{x};t) \triangleq \bigcup_{S \in \mathcal{T}^{L_{\max}}_{L_{\min}}(\boldsymbol{x})} \{S' \subseteq S : |S| - |S'| \leqslant t\}.$$

A code $\mathcal{C}$ is called a *t-deletion torn-paper code* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$ it holds that $\mathcal{DT}^{L_{\max}}_{L_{\min}}(\boldsymbol{x}_1;t) \cap \mathcal{DT}^{L_{\max}}_{L_{\min}}(\boldsymbol{x}_2;t) = \emptyset$.

In this section, we utilize *burst-erasure-correcting (BEC) codes* in our constructions, which are defined next. For a string $\boldsymbol{x}$, its *t-burst L-erasures ball*, denoted by $\mathcal{B}^L_{\mathrm{BE}}(\boldsymbol{x};t)$, is defined as the set of all strings that can be obtained from $\boldsymbol{x}$ by at most t burst of erasures, each of length at most L. A code $\mathcal{C}$ is called a *t-burst L-erasure correcting code* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$, $\mathcal{B}^L_{\mathrm{BE}}(\boldsymbol{x}_1;t) \cap \mathcal{B}^L_{\mathrm{BE}}(\boldsymbol{x}_2;t) = \emptyset$.

Next, we present a generic construction of t-deletion torn-paper codes. Let $\widehat{L}_{\max} \triangleq L_{\max} - \left\lceil\frac{L_{\max}}{L_{\min}}\right\rceil(\alpha + f(n) + 2)$. This construction is based on Construction A and assumes the existence of a systematic linear t-burst $\widehat{L}_{\max}$-erasure correcting code, denoted by $\mathcal{C}_{\mathrm{BEC}}$.

Construction C: Let $\rho > 0$ be an integer that is determined next. This construction uses the following family of codes:

Systematic BEC encoding. Let $\mathrm{Enc}_{\mathrm{BEC}}\colon \Sigma^{(K-\rho)N} \to \Sigma^{\rho_{\mathrm{BEC}}}$ denote the systematic encoder of the code $\mathcal{C}_{\mathrm{BEC}}$, such that for any string $\boldsymbol{v} \in \Sigma^{(K-\rho)N}$, $\boldsymbol{v} \circ \mathrm{Enc}_{\mathrm{BEC}}(\boldsymbol{v}) \in \mathcal{C}_{\mathrm{BEC}}$ (for convenience we assume that $\mathrm{Enc}_{\mathrm{BEC}}(\boldsymbol{v})$ returns only the encoded systematic redundancy symbols). The redundancy of this encoder is denoted by ρ_{BEC}. The parameter ρ is defined $\rho \triangleq \left\lceil\frac{1}{N}\left\lfloor\rho_{\mathrm{BEC}} \cdot \frac{f(n)}{f(n)-1}\right\rfloor\right\rceil$.

Next, we utilize a generalized concatenated coding approach, where Construction A is used as inner-code for $K - \rho$ information blocks, and with a slight adjustment also for the ρ redundant blocks, as follows:

1) *The length of the input string* $\boldsymbol{x}$. The input of this construction is $\boldsymbol{x} \in \Sigma^{(K-\rho)m(N)}$. That is, this construction has additional redundancy of $\rho m(N)$ symbols compared to Construction A. The input string is divided to $K - \rho$ information blocks each of length $m(N)$, denoted by $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{K-\rho-1}$.
2) *The generation of the encoded blocks* $\boldsymbol{y}_i$*'s*. The first $K - \rho$ blocks are generated from the corresponding $\boldsymbol{x}_i$'s using the RLL encoder E^{RLL}_m similarly to Construction A. Let $\boldsymbol{y}^* \triangleq \boldsymbol{y}_0 \circ \cdots \circ \boldsymbol{y}_{K-\rho-1} \in \Sigma^{(K-\rho)N}$ denote their concatenation. Next, we apply $\mathrm{Enc}_{\mathrm{BEC}}$ to obtain $\boldsymbol{w} \triangleq \mathrm{Enc}_{\mathrm{BEC}}(\boldsymbol{y}^*)$, and denote by $\boldsymbol{w}^*$ the result of inserting '1's into $\boldsymbol{w}$ at every location divisible by $f(n)$ (in particular, $\boldsymbol{y}^* \circ \boldsymbol{w}^*$ does not contain a length-$f(n)$ zero-run). Then, $\boldsymbol{w}^*$ is divided to the remaining segments $\boldsymbol{y}_{K-\rho}, \ldots, \boldsymbol{y}_{K-1} \in \Sigma^N$ (if $|\boldsymbol{w}^*|$ is not a multiple of N, $\boldsymbol{y}_{K-1}$ is padded with 1's to length N). Note that the parameter ρ satisfies $\rho N \geqslant \left\lfloor\rho_{\mathrm{BEC}} \cdot \frac{f(n)}{f(n)-1}\right\rfloor = |\boldsymbol{w}^*|$,

hence one may continue to follow the steps of Construction A without change.

We now indeed continue identically to Construction A. That is, an index and a marker are appended to the beginning of each encoded block $\boldsymbol{y}_i$ to construct a segment $\boldsymbol{z}_i$ of length $L_{\min}$. Then, $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{K-1}$ are concatenated along with $\boldsymbol{z}_K \circ 0^{n \bmod L_{\min}} = \boldsymbol{c}''_K \circ 10^{f(n)} 10^{N+(n \bmod L_{\min})}$ to obtain the encoded output string $\boldsymbol{z} \in \Sigma^n$. □

Let $\mathcal{C}_{\mathrm{del}}(n)$ denote the constructed code. The correctness of Construction C and redundancy calculation are proved in the next theorem.

Theorem 23: $\mathcal{C}_{\mathrm{del}}(n)$ is a t-deletion torn-paper code. Furthermore, it holds that

$$\mathrm{red}(\mathcal{C}_{\mathrm{del}}(n)) = \mathrm{red}(\mathcal{C}_A(n)) + m(N)\left\lceil \frac{1}{N}\left\lfloor \rho_{\mathrm{BEC}} \cdot \frac{f(n)}{f(n)-1}\right\rfloor\right\rceil.$$

Proof: Let $\boldsymbol{z} \in \mathcal{C}_{\mathrm{del}}(n)$ be the encoded codeword of the input string $\boldsymbol{x}$, and take $\mathcal{U} \in \mathcal{DT}^{L_{\max}}_{L_{\min}}(\boldsymbol{z}; t)$. We shall prove that one can uniquely decode $\boldsymbol{x}$.

From Lemma 16, for every $\boldsymbol{u} \in \mathcal{U}$ which is not a substring of the suffix of length $(n \mod L_{\min}) + N + f(n)$ of $\boldsymbol{z}$, its index $\mathrm{Ind}(\boldsymbol{u})$ can be decoded using Algorithm 2. The string $\boldsymbol{z}$ can then be reconstructed by the locations of each received segment, with some segments erased (at identifiable locations). Let $\boldsymbol{z}' \in (\Sigma \cup \{?\})^n$ denote this partially reconstructed string, where '?' stands for erased symbols.

From the definition of $\mathcal{DT}^{L_{\max}}_{L_{\min}}(\boldsymbol{z}; t)$, at most t segments of $\boldsymbol{z}$ are missing from $\mathcal{U}$. Therefore, $\boldsymbol{z}' \in \mathcal{B}^{L_{\max}}_{\mathrm{BE}}(\boldsymbol{z}; t)$. By removing coordinates of $\boldsymbol{z}'$ corresponding to indices or markers (including '?' symbols), a string $\boldsymbol{y}' \in \mathcal{B}^{\widehat{L}_{\max}}_{\mathrm{BE}}(\boldsymbol{y}_0 \circ \cdots \circ \boldsymbol{y}_{K-1}; t)$ is obtained, since there are at most $L_{\max} - \widehat{L}_{\max} = \lceil \frac{L_{\max}}{L_{\min}} \rceil (\alpha + f(n) + 2)$ symbols in any $L_{\max}$-segment of $\boldsymbol{z}$ belonging to either index or marker.

Finally, we remove from $\boldsymbol{y}'$ coordinates corresponding to '1's inserted into $\boldsymbol{w}^*$; thus, we obtain a string $\widehat{\boldsymbol{y}} \in \mathcal{B}^{\widehat{L}_{\max}}_{\mathrm{BE}}(\boldsymbol{y}^* \circ \boldsymbol{w}; t)$. A decoder for $\mathcal{C}_{\mathrm{BEC}}$ may be invoked on $\widehat{\boldsymbol{y}}$ to retrieve $\boldsymbol{y}^* = \boldsymbol{y}_0 \circ \cdots \circ \boldsymbol{y}_{K-\rho-1}$, and consequently $\boldsymbol{x}$ is obtained by applying the RLL decoder to each $\boldsymbol{y}_i$, $i \in [K-\rho]$.

To conclude the proof we observe that the asserted redundancy follows by definition, as precisely $\rho m(N)$ less information symbols are input at the encoder, in comparison to Construction A. ■

Next, we note that an extension to the case $k > 1$, i.e., to *t-deletion multi-strand torn-paper codes*, is again straightforward.

Corollary 24: Amending Construction C, one constructs a t-deletion multi-strand torn-paper code $\mathcal{C}_{\mathrm{del}}(n,k)$ with redundancy

$$\mathrm{red}(\mathcal{C}_{\mathrm{del}}(n,k)) = \mathrm{red}(\mathcal{C}_A(n,k)) + m(N)\left\lceil \frac{1}{N}\left\lfloor \rho_{\mathrm{BEC}} \cdot \frac{f(n)}{f(n)-1}\right\rfloor\right\rceil.$$

Proof: Here, an information string $\boldsymbol{x} \in \Sigma^{(kK-\rho)m(N)}$ is encoded with E^{RLL}_m into $\boldsymbol{y}^*$, and $\boldsymbol{w}^*$ is obtained utilizing a systematic BEC encoder on strings in $\Sigma^{(kK-\rho)N}$. It is segmented into $\boldsymbol{y}_{kK-\rho}, \ldots, \boldsymbol{y}_{kK} \in \Sigma^N$; again, observing $\rho N \geqslant \rho_{\mathrm{BEC}}$ assures that this is possible. Then, each K segments $\boldsymbol{y}_j$ are encoded, in order, with the remaining steps of Algorithm 1, where again $I \triangleq \lceil \log(k\lceil n/L_{\min}\rceil)\rceil$ and $L_{\min} = \lceil a \log(nk)\rceil$, and indices are utilized by each operation in succession. It is straightforward that the proof of Theorem 23 can be followed to show that $\mathcal{C}_{\mathrm{del}}(n,k)$ is a t-deletion multi-strand torn-paper code, with the above redundancy. ■

Before concluding the section, we discuss the cases of $t \in \{1,2\}$, in which more is known on the construction of BEC codes.

For $t = 1$, we use a systematic interleaving parity BEC code as the code $\mathcal{C}_{\mathrm{BEC}}$. Namely, the redundancy string $\boldsymbol{w} = \mathrm{Enc}_{\mathrm{BEC}}(\boldsymbol{y}^*)$ is of length $\rho_{\mathrm{BEC}} = \widehat{L}_{\max}$, and

$$w_i \triangleq \sum\nolimits_{k \in \left[\left\lceil \frac{(K-\rho)N-i}{\widehat{L}_{\max}}\right\rceil\right]} y^*_{i+k\widehat{L}_{\max}}$$

for all $i \in [\widehat{L}_{\max}]$, i.e., w_i is a single parity symbol for $\left(y^*_i, y^*_{i+\widehat{L}_{\max}}, \ldots\right)$. Denote this code by $\mathcal{C}_{\mathrm{del},1}$.

For $t = 2$, we state for completeness the following basic proposition which draws the connection between burst-error-correcting codes and burst-erasure-correcting codes. We note that this fact has been mentioned before in [7], for a single burst of errors.

Lemma 25: For $0 < \ell \leqslant n$ and $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^n$, it holds that $\boldsymbol{x}, \boldsymbol{y}$ are confusable under t bursts of errors of lengths at most ℓ if and only if they are confusable under $2t$ bursts of erasures of lengths at most ℓ.

Proof: Denote $\boldsymbol{x} = (x_j)_{j\in[n]}$, $\boldsymbol{y} = (y_j)_{j\in[n]}$, and $I_i \triangleq \bigcup_{j\in[t]}(k^{(i)}_j + [\ell])$, for $i = 0,1$ and some $\left\{k^{(i)}_j\right\}_{j\in[t]} \subseteq [n]$. Assume there exist $\boldsymbol{e}^{(0)}, \boldsymbol{e}^{(1)} \in \Sigma^n$ such that $\boldsymbol{x} + \boldsymbol{e}^{(0)} = \boldsymbol{y} + \boldsymbol{e}^{(1)}$, and $\mathrm{supp}(\boldsymbol{e}^{(i)}) \subseteq I_i$, $i = 1,2$. Then, one observes that $\boldsymbol{x}_{[n]\setminus(I_0\cup I_1)} = \boldsymbol{y}_{[n]\setminus(I_0\cup I_1)}$.

Conversely, assume $\boldsymbol{x}_{[n]\setminus I} = \boldsymbol{y}_{[n]\setminus I}$, where $I \subseteq \bigcup_{j\in[2t]}(k_j + [\ell])$ for some $\left\{k_j\right\}_{j\in[2t]} \subseteq [n]$, and without loss of generality $\left\{k_j\right\}_{j\in[2t]}$ are increasing, and $k_j \leqslant k_{t+1} - \ell$ for all $j \leqslant t$. Let $I_i \triangleq \bigcup_{j\in(it+[t])}(k_j + [\ell])$ for $i = 1,2$, and observe that $I_0 \cup I_1 = I$, $I_0 \cap I_1 = \emptyset$. For $i = 1,2$ and $j \in [n]$, let

$$e^{(i)}_j \triangleq \begin{cases} (-1)^i (y_j - x_j), & j \in I_i; \\ 0, & \text{otherwise.} \end{cases}$$

Then, denoting $\boldsymbol{e}^{(i)} = (e^{(i)}_j)_{j\in[n]}$ for $i = 1,2$, we have $\boldsymbol{x} + \boldsymbol{e}^{(0)} = \boldsymbol{y} + \boldsymbol{e}^{(1)}$, which completes the proof. ■

A construction of 2-deletion torn-paper codes is derived from Construction C, using a BEC code for $t = 2$. Hence, by Lemma 25 one may use an $\widehat{L}_{\max}$-burst error-correcting code. Observe that Construction C requires a systematic encoder, which is guaranteed by several prior works with redundancy at most $\log((K-\rho)N) + \widehat{L}_{\max}$; see, e.g. [1] and [2]. These constructions require the alphabet Σ to be a field, and are linear and cyclic, which ensures the existence of a systematic encoder. For simplicity of derivation we bound this redundancy (from above) by $\widehat{L}_{\max} + \log(n)$. Let $\mathcal{C}_{\mathrm{del},2}$ denote this code.

The next corollary summarizes these results. For convenience, denote the difference

$$\Delta\,\mathrm{red}(\mathcal{C}(n)) \triangleq \mathrm{red}(\mathcal{C}(n)) - \mathrm{red}(\mathcal{C}_A(n)),$$

for a t-deletion torn-paper code $\mathcal{C}(n) \subseteq \Sigma^n$.

Corollary 26: For a prime power q and all admissible values of n and $f(n)$ in Construction A, where $f(n) = \omega(1)$, $f(n) = o(\log(n))$ and with the RLL encoders of [20] and [37], it holds that

$$\Delta\,\mathrm{red}(\mathcal{C}_{\mathrm{del},1}(n)) \leqslant \widehat{L}_{\max} \cdot \frac{f(n)}{f(n)-1},$$
$$\Delta\,\mathrm{red}(\mathcal{C}_{\mathrm{del},2}(n)) \leqslant (\widehat{L}_{\max} + \log(n)) \cdot \frac{f(n)}{f(n)-1}.$$

In particular, for $f(n) = (1+o(1))\sqrt{\log(n)}$,

$$\Delta\,\mathrm{red}(\mathcal{C}_{\mathrm{del},1}(n)) \leqslant \widehat{L}_{\max}\left(1 + \frac{1-o(1)}{\sqrt{\log(n)}}\right),$$
$$\Delta\,\mathrm{red}(\mathcal{C}_{\mathrm{del},2}(n)) \leqslant (\widehat{L}_{\max} + \log(n))\left(1 + \frac{1-o(1)}{\sqrt{\log(n)}}\right).$$

Note that if $L_{\max} = o(n)$ the asymptotic rate of $\mathcal{C}_{\mathrm{del},1}(n)$ and $\mathcal{C}_{\mathrm{del},2}(n)$ is asymptotically equal to the rate of $\mathcal{C}_A(n)$. Thus, efficient encoding and decoding of t-deletion torn-paper codes, $t = 1, 2$, is possible at rates arbitrarily close to the optimum.

V. Conclusion

In this paper, we study the adversarial torn-paper channel, for which we present fundamental bounds and code constructions. We further study several extensions of this model, including multi-strand storage, substitution errors, or incomplete coverage. Importantly, our proposed constructions have linear-run-time encoders and decoders, and the resulting codes achieve asymptotically optimal rates.

We mention again that the adversarial model we assume in this work is chosen to enable analysis in the worst-case. More realistically, an adversarial channel where the *average* of the received segments' lengths is bounded from below might be analyzed; unfortunately, this channel turns out to be hard to analyze in the worst-case, and such analysis is left for future works. It will be remarked that by the same methods of Lemma 1, it can be shown that the capacity of such an adversarial channel is bounded from above by $1 - \frac{1}{a}$, where the lower bound on the average segment length is chosen to be $a\log(n)$. Coding for this channel appears to be more challenging; we point to the fact that an adversary is able to segment a non-vanishing fraction of the channel input into short substrings as a likely reason for that difficulty.

A naive solution, where the lower bound on the average segment length is $a\log(n)$ and $a > \frac{q}{q-1}$, is to apply Construction C with parameter a' satisfying $1 < a' < (1-\frac{1}{q})a$; the decoder then discards any received segment shorter than $a'\log(n)$, creating at most $\frac{n}{a\log(n)}$ bursts of erasures of lengths at most $(a'-1)\log(n)$ (in the reconstructed information sequence). To recover the information, a BEC code $\mathcal{C}_{\mathrm{BEC}}$ is used; since $\frac{1}{Km(N)}(a'-1)\log(n)\frac{n}{a\log(n)} = \frac{(a'-1)/a}{Km(N)/n} = \frac{a'}{a} + o(1) < 1 - \frac{1}{q}$, a positive-rate BEC code exists for all a' in the permissible range (since positive-rate $\frac{a'-1}{a}n$-erasure-correcting codes exist in $\Sigma^{Km(N)}$); hence a' can be optimized according to Theorem 23, i.e., to maximize the achieved rate of $(1+o_n(1))\big(1-\frac{1}{a'}\big)\cdot R(\mathcal{C}_{\mathrm{BEC}})$. (Naively, one might utilize codes correcting $\frac{n}{a\log(n)}\cdot(a'-1)\log(n)$ erasures; by the GV bound, the achievable rate of this construction is at least $(1+o_n(1))\big(1-\frac{1}{a'}\big)\big(1-H_q(\frac{a'}{a})\big)$.) Alternatively, if any integer a' falls within the given range, Construction P can also be used with parameter a', where again segments shorter than $(a'+o(1))\log(n)$ are discarded at the decoder, and reconstructed based on a BEC code $\mathcal{C}_{\mathrm{BEC}}$ correcting up to $\frac{n}{a\log(n)}$ bursts of erasures of lengths at most $(1+o(1))\log(n)$ in $\Sigma^{n/a'}$; however, the achieved rate of this construction is similarly $\big(1-\frac{1}{a'}\big)\cdot R(\mathcal{C}_{\mathrm{BEC}})$, and applicable BEC codes are equivalent (i.e., they correct the same number of bursts, of length $(1+o(1))\log(n)$ in $\Sigma^{n/a'}$ instead of length $(a'-1)\log(n)$ in $\Sigma^{Km(N)} = \Sigma^{(1-1/a'+o(1))n}$).

Finally, for future research, we believe that applying our methods to a generalized channel, including multiple sources of noise concurrently, one may achieve similar results. Studying the channel under edit-errors, including insertions/deletions in addition to substitutions, is also of great interest for applications to DNA data storage.

Acknowledgment

The authors gratefully acknowledge the valuable insight and advice offered to them by the two anonymous reviewers and associate editor, which were instrumental in streamlining the presentation of this article.

References

[1] K. A. S. Abdel-Ghaffar, "On the existence of optimum cyclic burst correcting codes over GF(q)," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 329–332, Mar. 1988.

[2] K. A. S. Abdel-Ghaffar, R. J. McEliece, A. M. Odlyzko, and H. C. A. van Tilborg, "On the existence of optimum cyclic burst-correcting codes," *IEEE Trans. Inf. Theory*, vol. IT-32, no. 6, pp. 768–775, Nov. 1986.

[3] 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, Jan. 2015.

[4] F. Balado, "Capacity of DNA data embedding under substitution mutations," *IEEE Trans. Inf. Theory*, vol. 59, no. 2, pp. 928–941, Feb. 2013.

[5] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system," *ACM SIGPLAN Notices*, vol. 51, no. 4, pp. 637–649, Mar. 2016.

[6] G. Bresler, M. Bresler, and D. Tse, "Optimal assembly for high throughput shotgun sequencing," *BMC Bioinf.*, vol. 14, no. S5, p. S18, Jul. 2013.

[7] R. Chien, L. Bahl, and D. Tang, "Correction of two erasure bursts (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 186–187, Jan. 1969.

[8] C.-S. Chin et al., "Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data," *Nature Methods*, vol. 10, no. 6, pp. 563–569, Jun. 2013.

[9] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, p. 1628, Sep. 2012.

[10] Contributing Members, "Preserving our digital legacy: An introduction to DNA data storage," DNA Data Storage Alliance, San Francisco, CA, USA, White Paper, Jun. 2021. [Online]. Available: https://dnastoragealliance.org/dev/wp-content/uploads/2021/06/DNA-Data-Storage-Alliance-An-Introduction-to-DNA-Data-Storage.pdf and https://dnastoragealliance.org/dev/publications/

[11] T. van Aardenne-Ehrenfest and N. G. de Bruijn, "Circuits and trees in oriented linear graphs," *Simon Stevin, Wis En Natuurkundig Tijdschrift*, vol. 28, pp. 203–217, Jan. 1951.

[12] 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.

[13] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–954, Mar. 2017.

[14] 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.

[15] S. Ganguly, E. Mossel, and M. Z. Rácz, "Sequence assembly from corrupted shotgun reads," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 265–269.

[16] N. Goldman et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013.

[17] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," *Angew. Chem. Int. Ed.*, vol. 54, no. 8, pp. 2552–2555, Feb. 2015.

[18] R. Heckel, I. Shomorony, K. Ramchandran, and D. N. C. Tse, "Fundamental limits of DNA storage systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 3130–3134.

[19] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "An upper bound on the capacity of the DNA storage channel," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Aug. 2019, pp. 1–5.

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

[21] N. J. Loman, J. Quick, and J. T. Simpson, "A complete bacterial genome assembled de novo using only nanopore sequencing data," *Nature Methods*, vol. 12, no. 8, pp. 733–735, Aug. 2015.

[22] A. Motahari, K. Ramchandran, D. Tse, and N. Ma, "Optimal DNA shotgun sequencing: Noisy reads are as good as noiseless reads," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 1640–1644.

[23] A. S. Motahari, G. Bresler, and D. N. C. Tse, "Information theory of DNA shotgun sequencing," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6273–6289, Oct. 2013.

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

[25] L. Organick et al., "Random access in large-scale DNA data storage," *Nature Biotechnol.*, vol. 36, no. 3, pp. 242–248, Mar. 2018.

[26] 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)*, Jul. 2021, pp. 1937–1942.

[27] A. N. Ravi, A. Vahid, and I. Shomorony, "Coded shotgun sequencing," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 1, pp. 147–159, Mar. 2022.

[28] S. L. Salzberg, "Mind the gaps," *Nature Methods*, vol. 7, no. 2, pp. 105–106, 2010.

[29] I. Shomorony, T. Courtade, and D. Tse, "Do read errors matter for genome assembly," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 919–923.

[30] I. Shomorony and R. Heckel, "Capacity results for the noisy shuffling channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 762–766.

[31] I. Shomorony, G. M. Kamath, F. Xia, T. A. Courtade, and D. N. Tse, "Partial DNA assembly: A rate-distortion perspective," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1799–1803.

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

[33] J. Spencer, "Asymptotic lower bounds for Ramsey functions," *Discrete Math.*, vol. 20, pp. 69–76, Jan. 1977.

[34] R. R. Varshamov and G. M. Tenengolts, "Code correcting single asymmetric errors," (in Russian), *Automatika I Telemkhanika*, vol. 26, no. 2, pp. 288–292, 1965.

[35] N. Weinberger and N. Merhav, "The DNA storage channel: Capacity and error probability bounds," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 5657–5700, Sep. 2022.

[36] P. C. Wong, K.-K. Wong, and H. Foote, "Organic data memory using the DNA approach," *Commun. ACM*, vol. 46, no. 1, pp. 95–98, Jan. 2003.

[37] Y. Yehezkeally, D. Bar-Lev, S. Marcovich, and E. Yaakobi, "Generalized unique reconstruction from substrings," *IEEE Trans. Inf. Theory*, early access, Apr. 21, 2023, doi: 10.1109/TIT.2023.3269124.

[38] Y. Yehezkeally and N. Polyanskii, "On codes for the noisy substring channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 1700–1705.

Daniella Bar-Lev (Student Member, IEEE) received the bachelor's degree in mathematics, the bachelor's degree in computer science, and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2019 and 2021, respectively, where she is currently pursuing the Ph.D. degree with the Department of Computer Science. Her research interests include algorithms, discrete mathematics, coding theory, and DNA storage.

Sagi Marcovich (Student Member, IEEE) received the B.Sc. degree in software engineering and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2016 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. His research interests include algorithms, information theory, and coding theory with applications to DNA-based storage.

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

Yonatan Yehezkeally (Member, IEEE) received the B.Sc. degree (cum laude) in mathematics and the M.Sc. (summa cum laude) and Ph.D. degrees in electrical and computer engineering from the Ben-Gurion University of the Negev, Israel, in 2013, 2017, and 2020 respectively. He is currently a Carl Friedrich von Siemens Post-Doctoral Research Fellow of the Alexander von Humboldt Foundation, with the Associate Professorship of coding and cryptography (Prof. Wachter-Zeh), School of Computation, Information and Technology, Technical University of Munich. His research interests include coding for novel storage media, with a focus on DNA-based storage and nascent sequencing technologies, as well as combinatorial structures and finite group theory.