Decoding for Optimal Expected Normalized Distance over the t-Deletion Channel

Daniella Bar-Lev*†, **Yotam Gershon***‡, **Omer Sabary***§, and **Eitan Yaakobi**†

† Department of Computer Science, Technion — Israel Institute of Technology, Haifa, 3200003 Israel

‡ Department of Electrical Engineering, Technion — Israel Institute of Technology, Haifa, 3200003 Israel

§ Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA

Emails: `{daniellalev,yaakobi}@cs.technion.ac.il`, `yotamgr@campus.technion.ac.il`, `osabary@ucsd.edu`

***Abstract*—This paper studies optimal decoding for a special case of the deletion channel, referred by the *t-deletion channel*, which deletes exactly t symbols of the transmitted word uniformly at random. The goal of the paper is to understand how such an optimal decoder operates in order to minimize the *expected normalized distance*. A full characterization of a decoder for this setup is given for a channel that deletes one or two symbols. For $t = 1$ it is shown that when the code is the entire space, the decoder is the *lazy decoder* which simply returns the channel output. Similarly, for $t = 2$ it is shown that the decoder acts as the lazy decoder in almost all cases and when the longest run is significantly long, it prolongs the longest run by one symbol.**

I. Introduction

Codes correcting insertions/deletions recently attracted considerable attention due to their relevance to the special error behavior in DNA-based data storage [4], [25], [28], [33], [35], [36], [48], [49]. These codes are relevant for other applications in communication models. For example, insertions/deletions happen during the synchronization of files and symbols of data streams [39] or due to over-sampling and under-sampling at the receiver side [14]. The algebraic concepts of codes correcting insertions/deletions date back to the 1960s when Varshamov and Tenengolts designed a class of binary codes, nowadays called *VT codes* [45]. These codes were originally designed to correct a single asymmetric error and later were proven to correct a single insertion/deletion [29]. Extensions for multiple deletions were recently proposed in several studies; see e.g. [5], [18], [40], [41]. However, while codes correcting substitution errors were widely studied and efficient capacity achieving codes are used conventionally, much less is known for codes correcting insertions/deletions. More than that, even the deletion channel capacity is far from being solved [7], [9], [10], [12], [31], [32], [34].

There are two main models which are studied for deletion errors. While in the first one, the goal is to correct a fixed number of deletions in the worst case, for the second one, which corresponds to the channel capacity of the deletion channel, one seeks to construct codes which correct a fraction of deletions with high probability [6], [9], [11], [13], [15], [17], [26], [27], [32], [44], [46]. This paper considers a combination of these two models. In this channel, referred as the *t-deletion channel*, t symbols of the transmitted word are deleted uniformly at random. Consider the case of $t = 1$, i.e., one of the n transmitted symbols is deleted, each with the same probability. In case the transmitted word belongs to a single-deletion-correcting code then clearly it is possible to successfully decode the transmitted word. However, if such error correction capability is not guaranteed in the worst case, two approaches can be of interest. In the first, one may output a list of all possible transmitted words, that is, *list decoding* for deletion errors as was studied recently in several works; see e.g. [20]–[24], [26], [30], [47]. The second one, which is taken in the present work, seeks to output a word that minimizes the decoding error probability. This channel was also studied in several previous works. In [19], the author studied the maximal length of words that can be uniquely reconstructed using a sufficient number of channel outputs of the t-deletion channel and calculated this maximal length explicitly for $n - t \leqslant 6$. In [2], the goal was to study the entropy of the set of all possible channel input words, given a corrupted word from a channel that deletes either one or two symbols. The minimum and maximum values of this entropy value were explored. Another variation of this channel was studied in [1]. More related problems to the setup considered in this paper were studied in [8], [37], [38], [42], [43] where the goal is to reconstruct a message from several noisy copies, while each one is transmitted through a deletion channel. However, the problem studied in this paper assumes only a single channel and a special case of the deletion channel which deletes a fixed number of symbols.

Mathematically speaking, assume S is a channel that is characterized by a conditional probability $\mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } | \boldsymbol{x} \text{ trans.}\}$, for every pair $(\boldsymbol{x}, \boldsymbol{y}) \in (\Sigma_q^*)^2$. A decoder for a code $\mathcal{C}$ with respect to the channel S is a function $\mathcal{D} : \Sigma_q^* \to \mathcal{C}$. Its *average decoding failure probability* is the probability that the decoder output is not the transmitted word. The *maximum-likelihood* (*ML*) *decoder* for $\mathcal{C}$ with respect to S, denoted by $\mathcal{D}_{\mathsf{ML}}$, outputs a codeword $\boldsymbol{c} \in \mathcal{C}$ that maximizes the probability $\mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } | \boldsymbol{c} \text{ trans.}\}$. This decoder minimizes the average decoding *failure* probability and thus it outputs only codewords. However, if one seeks to minimize the *expected normalized distance*, where the distance function depends upon the channel of interest, then the decoder should consider non-codewords as well. The goal of this work is to study the *ML** *decoder*, which outputs words that minimize the expected normalized distance for the t-deletion channel.

The rest of the paper is organized as follows. Section II presents the formal definition of channel transmission and maximum likelihood decoding in order to minimize the expected normalized distance. This section introduces also the t-deletion channel. Section III studies the 1-deletion channel. It introduces two types of decoders. The first one, referred as the *embedding number decoder*, maximizes the so-called *embedding number* between the channel output and all possible codewords. The second one is called the *lazy decoder* and it simply returns the channel output. The main result of this section states that if the code is the entire space then the ML* decoder is the lazy decoder. Similarly, Section IV studies the 2-deletion channel where it is shown that in almost all cases the ML* decoder should act as the lazy decoder and in the rest of the cases it returns a length-$(n-1)$ word which maximizes the embedding number. Due to the lack of space some of the proofs are omitted from this paper, however they can be found in the extended version of the paper [3].

II. Definitions and Preliminaries

Denote by $\Sigma_q = \{0, \ldots, q-1\}$ the alphabet of size q and $\Sigma_q^* \triangleq \bigcup_{\ell=0}^{\infty} \Sigma_q^{\ell}$. The length of $\boldsymbol{x} \in \Sigma^n$ is denoted by $|\boldsymbol{x}| = n$. The *Levenshtein distance* between two words $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_q^*$, denoted by $d_L(\boldsymbol{x}, \boldsymbol{y})$, is the minimum number of insertions and deletions required to transform $\boldsymbol{x}$ into $\boldsymbol{y}$, and $d_H(\boldsymbol{x}, \boldsymbol{y})$ denotes the *Hamming distance* between $\boldsymbol{x}$ and $\boldsymbol{y}$, when $|\boldsymbol{x}| = |\boldsymbol{y}|$. For a

[*]The three first authors contributed equally to this work.

978-1-5386-8209-8/21/$31.00 ©2021 IEEE

word $\boldsymbol{x} \in \Sigma_q^*$ and a set of indices $I \subseteq [|\boldsymbol{x}|]$, the word $\boldsymbol{x}_I$ is the *projection* of $\boldsymbol{x}$ on the indices of I which is the subsequence of $\boldsymbol{x}$ received by the symbols in the entries of I. For two words $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_q^*$, the number of times that $\boldsymbol{y}$ can be received as a subsequence of $\boldsymbol{x}$ is called the *embedding number of $\boldsymbol{y}$ in $\boldsymbol{x}$* [2], [16], [42], defined by $\mathsf{Emb}(\boldsymbol{x}; \boldsymbol{y}) = |\{I \subseteq [|\boldsymbol{x}|] \mid \boldsymbol{x}_I = \boldsymbol{y}\}|$. Note that if $\boldsymbol{y}$ is not a subsequence of $\boldsymbol{x}$ then $\mathsf{Emb}(\boldsymbol{x}; \boldsymbol{y}) = 0$.

The *radius-r insertion ball* of a word $\boldsymbol{x} \in \Sigma_q^*$, denoted by $I_r(\boldsymbol{x})$, is the set of all supersequences of $\boldsymbol{x}$ of length $|\boldsymbol{x}| + r$. From [29] it is known that $I_r(\boldsymbol{x}) = \sum_{i=0}^{r} \binom{|\boldsymbol{x}|+r}{i}$. Similarly, the *radius-r deletion ball* of a word $\boldsymbol{x} \in \Sigma_q^*$, denoted by $D_r(\boldsymbol{x})$, is the set of all subsequences of $\boldsymbol{x}$ of length $|\boldsymbol{x}| - r$.

We consider a channel S that is characterized by a conditional probability Pr_{S}, defined by $\mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{x} \text{ trans.}\}$, for all $(\boldsymbol{x}, \boldsymbol{y}) \in (\Sigma_q^*)^2$. Note that the lengths of the input and output words may not be the same as we consider deletions in this work. A decoder for a code $\mathcal{C}$ with respect to the channel S is a function $\mathcal{D} : \Sigma_q^* \to \mathcal{C}$. Its *average decoding failure probability* is defined by $\mathsf{P}_{\mathsf{fail}}(\mathsf{S}, \mathcal{C}, \mathcal{D}) = \frac{\sum_{\boldsymbol{c} \in \mathcal{C}} \mathsf{P}_{\mathsf{fail}}(\boldsymbol{c})}{|\mathcal{C}|}$, where

$$\mathsf{P}_{\mathsf{fail}}(\boldsymbol{c}) = \sum_{\boldsymbol{y}:\mathcal{D}(\boldsymbol{y}) \neq \boldsymbol{c}} \mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{c} \text{ trans.}\}.$$

We will mostly be interested in the *expected normalized distance* which is the average normalized distance between the transmitted word and the decoder's output. The distance will depend upon the channel. For example, for the BSC one should consider the Hamming distance, while for insertion/deletion channels, the Levenshtein distance will be of interest. Hence, for a channel S, distance function d, and a decoder $\mathcal{D}$, we let $\mathsf{P}_{\mathsf{err}}(\mathsf{S}, \mathcal{C}, \mathcal{D}, d) = \frac{\sum_{\boldsymbol{c} \in \mathcal{C}} \mathsf{P}_{\mathsf{err}}(\boldsymbol{c}, d)}{|\mathcal{C}|}$, where

$$\mathsf{P}_{\mathsf{err}}(\boldsymbol{c}, d) = \sum_{\boldsymbol{y}:\mathcal{D}(\boldsymbol{y}) \neq \boldsymbol{c}} \frac{d(\mathcal{D}(\boldsymbol{y}), \boldsymbol{c})}{|\boldsymbol{c}|} \cdot \mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{c} \text{ trans.}\}.$$

The *maximum-likelihood* (*ML*) *decoder* for $\mathcal{C}$ with respect to a channel S, denoted by $\mathcal{D}_{\mathsf{ML}}$, outputs a codeword $\boldsymbol{c} \in \mathcal{C}$ that maximizes the probability $\mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{c} \text{ trans.}\}$. That is, for $\boldsymbol{y} \in \Sigma_q^*$, $\mathcal{D}_{\mathsf{ML}}(\boldsymbol{y}) = \arg\max_{\boldsymbol{c} \in \mathcal{C}} \{\mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{c} \text{ trans.}\}\}$. It is well known that for the BSC, the ML decoder chooses the closest codeword with respect to the Hamming distance.

Note that channels which introduce deletions or insertions change the sequence's length. If the goal is to minimize the average decoding *failure* probability then clearly the decoder's output should be a codeword as there is no point in outputting a non-codeword. However, if one seeks to minimize the average decoding *error* probability, then the decoder should consider non-codewords as well. Therefore, we present here the ML* decoder, which is an alternative definition of the ML decoder that takes into account non-codewords and in particular words with different length than the code length. The *maximum-likelihood** (*ML**) *decoder* for $\mathcal{C}$ with respect to a channel S, denoted by $\mathcal{D}_{\mathsf{ML}^*}$, should output words that minimize the expected normalized distance $\mathsf{P}_{\mathsf{err}}(\mathsf{S}, \mathcal{C}, \mathcal{D}, d)$:

$$\begin{aligned}
\mathsf{P}_{\mathsf{err}}(\mathsf{S}, \mathcal{C}, \mathcal{D}, d) &= \frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{c} \in \mathcal{C}} \mathsf{P}_{\mathsf{err}}(\boldsymbol{c}, d) \\
&= \frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{c} \in \mathcal{C}} \sum_{\boldsymbol{y}:\mathcal{D}(\boldsymbol{y}) \neq \boldsymbol{c}} \frac{d(\mathcal{D}(\boldsymbol{y}), \boldsymbol{c})}{|\boldsymbol{c}|} \cdot \mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{c} \text{ trans.}\} \\
&\overset{(a)}{=} \frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{y} \in \Sigma_q^*} \sum_{\boldsymbol{c}:\mathcal{D}(\boldsymbol{y}) \neq \boldsymbol{c}} \frac{d(\mathcal{D}(\boldsymbol{y}), \boldsymbol{c})}{|\boldsymbol{c}|} \mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{c} \text{ trans.}\},
\end{aligned}$$

where in (a) we switched the summation order, while taking into account all possible channel's outputs. For every $\boldsymbol{y} \in \Sigma_q^*$, denote the value $\sum_{\boldsymbol{c}:\mathcal{D}(\boldsymbol{y}) \neq \boldsymbol{c}} \frac{d(\mathcal{D}(\boldsymbol{y}), \boldsymbol{c})}{|\boldsymbol{c}|} \mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{c} \text{ trans.}\}$ by $f_{\boldsymbol{y}}(\mathcal{D}(\boldsymbol{y}))$ and if $\mathcal{D}(\boldsymbol{y})$ is some arbitrary value $\boldsymbol{x}$, this value is denoted by $f_{\boldsymbol{y}}(\boldsymbol{x})$. Thus, the ML* decoder is defined as

$$\mathcal{D}_{\mathsf{ML}^*}(\boldsymbol{y}) = \underset{\boldsymbol{x} \in \Sigma_q^*}{\operatorname{argmin}} \{f_{\boldsymbol{y}}(\boldsymbol{x})\}.$$

In this paper we study the ML* decoder for a special case of the deletion channel that is denoted by t-Del and is referred as the *t-deletion channel*. In this channel, defined also in [19], exactly t symbols of the transmitted word are deleted. The t symbols are selected randomly and independently out of the $\binom{n}{t}$ options to delete t out of the n symbols, where n is the word length. Note that it may happen that different deletion patterns will still result with the same output. In this work, whenever the set $\arg\min_{\boldsymbol{x} \in \Sigma_q^*}\{f_{\boldsymbol{y}}(\boldsymbol{x})\}$ contains more than one word, we assume that $\mathcal{D}_{\mathsf{ML}^*}(\boldsymbol{y})$ returns a word of minimum length. Section III is dedicated to the case of $t = 1$, while in Section IV the $t = 2$ case is solved. In both cases we provide a full characterization of the ML* decoder and its expected normalized distance when the code is Σ_2^n. In the analysis to follow in this paper, when the channel being discussed is clear from the context, the conditional probability $\mathsf{Pr}_{\mathsf{S}}\{\boldsymbol{y} \text{ rec. } |\boldsymbol{c} \text{ trans.}\}$ will be denoted by $p(\boldsymbol{y}|\boldsymbol{c})$.

III. The 1-Deletion Channel

In this section we consider the 1-deletion channel which deletes one symbol randomly. Given a single-deletion-correcting code, any channel output can be easily decoded, and therefore for the rest of this section we assume that the given code is not a single-deletion-correcting code. We start by examining two types of decoders for this channel. The first decoder, referred as the *embedding number decoder* and denoted by $\mathcal{D}_{EN}$, returns for a channel output $\boldsymbol{y}$ the word $\mathcal{D}_{EN}(\boldsymbol{y})$ which is a codeword in the code $\mathcal{C}$ that maximizes the embedding number of $\boldsymbol{y}$ in $\mathcal{D}_{EN}(\boldsymbol{y})$. That is,

$$\mathcal{D}_{EN}(\boldsymbol{y}) = \underset{\boldsymbol{c} \in \mathcal{C}}{\arg\max}\{\mathsf{Emb}(\boldsymbol{c}; \boldsymbol{y})\},$$

where, for now, if there is more than one such a word, the decoder chooses one of them arbitrarily. The second decoder, referred as the *lazy decoder*, is denoted by $\mathcal{D}_{Lazy}$. For a channel output $\boldsymbol{y}$, $\mathcal{D}_{Lazy}$ simply returns $\boldsymbol{y}$ as the output, i.e., $\mathcal{D}_{Lazy}(\boldsymbol{y}) = \boldsymbol{y}$. Note that the lazy decoder does not return a codeword. Additionally, $d_L(\mathcal{D}_{Lazy}(\boldsymbol{y}), \boldsymbol{c}) = 1$ since $\boldsymbol{y} \in D_1(\boldsymbol{c})$ and hence, the average decoding error probability of the lazy decoder is $\frac{1}{n}$, when n is the code length.

In the main result of this section, presented in Theorem 7, we prove for $\mathsf{S} = 1$-Del and $\mathcal{C} = \Sigma_2^n$, that $\mathcal{D}_{Lazy}$ performs at least as good as any other decoder, and hence $\mathcal{D}_{Lazy} = \mathcal{D}_{\mathsf{ML}^*}$.

For the rest of this section it is assumed that $\mathcal{C} \subseteq \Sigma_2^n$ and $\mathsf{S} = 1$-Del. The following lemma states that $\mathcal{D}_{Lazy}$ is preferable over any other decoder that outputs words of length $n - 1$.

Lemma 1. *Let $\mathcal{D} : \Sigma_2^{n-1} \to \Sigma_2^{n-1}$ be a general decoder that preserves the length of the channel output. Then,*

$$P_{\text{err}}(1\text{-}\mathsf{Del}, \mathcal{C}, \mathcal{D}, d_L) \geqslant P_{\text{err}}(1\text{-}\mathsf{Del}, \mathcal{C}, \mathcal{D}_{\text{Lazy}}, d_L) = 1/n,$$

and for $\mathcal{C} = \Sigma_2^n$ equality is obtained if and only if $\mathcal{D} = \mathcal{D}_{\text{Lazy}}$.

Next, we discuss decoders that return only words of length n and for the rest of this section it is assumed that $\mathcal{C} = \Sigma_2^n$. Note that a decoder that prolongs an arbitrary run of maximal length within its input word is equivalent to the embedding number decoder. This observation holds since their embedding numbers are equal. Therefore, we can define the *embedding number decoder* to be the decoder that, given a channel output $\boldsymbol{y} \in \Sigma_2^{n-1}$, prolongs the first run of maximal length in $\boldsymbol{y}$ by one. A decoder $\mathcal{D}$ that prolongs one of the runs of maximal length in $\boldsymbol{y}$ by one is said to be *equivalent* to the embedding number decoder, and is denoted by $\mathcal{D} \equiv \mathcal{D}_{\text{EN}}$.

Lemma 2. *For every $\boldsymbol{c} \in \mathcal{C} = \Sigma_2^n$, the decoder $\mathcal{D}_{EN}$ satisfies*

$$P_{\text{err}}(\boldsymbol{c}, d_L) = \frac{2}{n} \cdot \sum_{\boldsymbol{y} \in D_1(\boldsymbol{c})} \frac{\mathsf{Emb}(\boldsymbol{c}; \boldsymbol{y})}{n} \cdot \mathbb{I}\{\mathcal{D}_{\text{EN}}(\boldsymbol{y}) \neq \boldsymbol{c}\}.$$

For $\boldsymbol{y} \in D_1(\boldsymbol{c})$, we get $\mathcal{D}_{\text{EN}}(\boldsymbol{y}) = \boldsymbol{c}$ if and only if the deletion occurred within the run which is corresponding to the first run of maximal length in $\boldsymbol{y}$. Hence, the embedding number decoder will fail at least for any deletion occurring outside of the first run of maximal length in $\boldsymbol{c}$. This observation will be used in the proof of the Lemma 3. Before presenting this proof, one more definition is introduced. For a word $\boldsymbol{x} \in \Sigma_2^n$, we denote by $\tau(\boldsymbol{x})$ the length of its maximal run. For example $\tau(00111010) = 3$ and $\tau(01010101) = 1$. For a code $\mathcal{C} \subseteq \Sigma_2^n$, we denote by $\tau(\mathcal{C})$ the average length of the maximal runs of its codewords. That is, $\tau(\mathcal{C}) = \frac{\sum_{\boldsymbol{c}\in\mathcal{C}} \tau(\boldsymbol{c})}{|\mathcal{C}|}$. Furthermore, if $N(r)$, for $1 \leqslant r \leqslant n$ denotes the number of codewords in $\mathcal{C}$ in which the length of their maximal run is r, then $\tau(\mathcal{C}) = \frac{\sum_{r=1}^{n} r\cdot N(r)}{|\mathcal{C}|}$. We are now ready to present a lower bound on the average decoding error probability of $\mathcal{D}_{EN}$.

Lemma 3. *The average decoding error probability of the embedding number decoder* $\mathcal{D}_{\mathrm{EN}}$ *satisfies*

$$P_{\mathrm{err}}(\text{1-Del}, \mathcal{C} = \Sigma_2^n, \mathcal{D}_{\mathrm{EN}}, d_L) \geqslant \frac{2}{n} \cdot \left(1 - \frac{\tau(\mathcal{C})}{n}\right).$$

Proof: Let $\mathcal{C}_r \subseteq \mathcal{C}$ be the subset of codewords with maximal run of length r, and let its size be denoted by $N(r)$. For any $\boldsymbol{c} \in \mathcal{C}$, any deletion outside of the first run of maximal length will result in a decoding failure. The summation $\sum_{\boldsymbol{y}\in D_1(\boldsymbol{c})} \frac{\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y})}{n} \cdot \mathbb{I}\{\mathcal{D}_{\mathrm{EN}}(\boldsymbol{y}) \neq \boldsymbol{c}\}$ is equivalent to counting the indices in $\boldsymbol{c}$ in which a deletion will result in a decoding failure. Hence, by Lemma 2 we get that for every $\boldsymbol{c} \in \mathcal{C}_r$, $P_{\mathrm{err}}(\boldsymbol{c}, d_L) \geqslant \frac{2}{n} \cdot \frac{n-r}{n}$, which implies,

$$P_{\mathrm{err}}(\text{1-Del}, \mathcal{C}, \mathcal{D}_{\mathrm{EN}}, d_L) = \sum_{\boldsymbol{c}\in\mathcal{C}} \frac{P_{\mathrm{err}}(\boldsymbol{c}, d_L)}{|\mathcal{C}|} = \frac{1}{|\mathcal{C}|} \sum_{r=1}^{n} \sum_{\boldsymbol{c}\in\mathcal{C}_r} P_{\mathrm{err}}(\boldsymbol{c}, d_L)$$
$$\geqslant \frac{1}{|\mathcal{C}|} \sum_{r=1}^{n} \sum_{\boldsymbol{c}\in\mathcal{C}_r} \frac{2}{n} \cdot \frac{n-r}{n} = \frac{1}{|\mathcal{C}|} \frac{2}{n} \sum_{r=1}^{n} N(r) \left(1 - \frac{r}{n}\right)$$
$$= \frac{2}{n} \left(1 - \frac{1}{n} \frac{\sum_{r=1}^{n} r \cdot N(r)}{|\mathcal{C}|}\right) = \frac{2}{n} \cdot \left(1 - \frac{\tau(\mathcal{C})}{n}\right).$$

■

The next lemma states that $\mathcal{D}_{EN}$ is preferable over any other decoder that outputs a word of length n.

Lemma 4. *Let* $\mathcal{D} : \Sigma_2^{n-1} \to \Sigma_2^n$ *be a general decoder that prolongs the input length by one. It follows that*

$$P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}, d_L) \geqslant P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}_{\mathrm{EN}}, d_L), \tag{1}$$

and equality is obtained if and only if $\mathcal{D} \equiv \mathcal{D}_{\mathrm{EN}}$.

By Lemma 3 and since the average length of the longest run of words in Σ_2^n is at most $2\log_2(n)$ [3], the lazy decoder is strictly preferable over the embedding number decoder.

Lemma 5. *For all* $n \geqslant 17$ *it holds that*

$$P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}_{\mathrm{EN}}, d_L) > P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}_{\mathrm{Lazy}}, d_L).$$

For the rest of this section we assume that $n \geqslant 17$. The next lemma states that $\mathcal{D}_{Lazy}$ is preferable over a *hybrid decoder*, which returns words of length either $n-1$ or n.

Lemma 6. *Let* $\mathcal{D} : \Sigma_2^{n-1} \to \Sigma_2^{n-1} \cup \Sigma_2^n$ *be a decoder that either preserves the word length or prolongs it by one symbol. Then,*

$$P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}, d_L) \geqslant P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}_{\mathrm{Lazy}}, d_L).$$

Proof: Since $|\mathcal{D}(\boldsymbol{y})| = n$, it holds that

$$P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}, d_L) = \frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{y}\in\Sigma_2^{n-1}} \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} \frac{d_L(\mathcal{D}(\boldsymbol{y}), \boldsymbol{c})}{|\boldsymbol{c}|} p(\boldsymbol{y}|\boldsymbol{c})$$
$$\geqslant \frac{2}{n|\mathcal{C}|} \sum_{\substack{\boldsymbol{y}\in\Sigma_2^{n-1} \\ |\mathcal{D}(\boldsymbol{y})|=n}} \left(\sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}) - p(\boldsymbol{y}|\mathcal{D}(\boldsymbol{y})) \right)$$
$$+ \frac{1}{n|\mathcal{C}|} \sum_{\substack{\boldsymbol{y}\in\Sigma_2^{n-1} \\ |\mathcal{D}(\boldsymbol{y})|=n-1}} \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}). \tag{2}$$

We show that for $\boldsymbol{y} \in \Sigma_2^{n-1}$ such that $|\mathcal{D}(\boldsymbol{y})| = n$ it holds

$$2 \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}) - 2p(\boldsymbol{y}|\mathcal{D}(\boldsymbol{y})) \geqslant \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}). \tag{3}$$

This is proved by verifying that

$$2 \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}) - 2p(\boldsymbol{y}|\mathcal{D}(\boldsymbol{y})) - \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}) = \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}) - 2p(\boldsymbol{y}|\mathcal{D}(\boldsymbol{y}))$$
$$\overset{(a)}{=} \sum_{\substack{\boldsymbol{c}\in I_1(\boldsymbol{y}) \\ \boldsymbol{c}\neq\mathcal{D}(\boldsymbol{y})}} p(\boldsymbol{y}|\boldsymbol{c}) + p(\boldsymbol{y}|\mathcal{D}(\boldsymbol{y})) - 2p(\boldsymbol{y}|\mathcal{D}(\boldsymbol{y}))$$
$$\overset{(b)}{\geqslant} \sum_{\substack{\boldsymbol{c}\in I_1(\boldsymbol{y}) \\ \boldsymbol{c}\neq\mathcal{D}(\boldsymbol{y})}} \frac{1}{n} - p(\boldsymbol{y}|\mathcal{D}(\boldsymbol{y})) \overset{(c)}{\geqslant} 1 - p(\boldsymbol{y}|\mathcal{D}(\boldsymbol{y})) \geqslant 0,$$

where in (a) we split $I_1(\boldsymbol{y})$ into two parts when $\mathcal{D}(\boldsymbol{y}) \in I_1(\boldsymbol{y})$, and note that this equality holds also when $\mathcal{D}(\boldsymbol{y}) \notin I_1(\boldsymbol{y})$. (b) holds since $p(\boldsymbol{y}|\boldsymbol{c}) \geqslant 1/n$ when $\boldsymbol{c} \in I_1(\boldsymbol{y})$. Lastly, (c) holds since $|I_1(\boldsymbol{y}) \setminus \{\mathcal{D}(\boldsymbol{y})\}|$ is at least n since $|I_1(\boldsymbol{y})| = n+1$.

Lastly, combining (2) and (3) and remembering that $d_L(\boldsymbol{c}, \mathcal{D}_{\mathrm{Lazy}}(\boldsymbol{y})) = 1$ we have that

$$P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}, d_L) \geqslant P_{\mathrm{err}}(\text{1-Del}, (\Sigma_2)^n, \mathcal{D}_{\mathrm{Lazy}}, d_L).$$

■

Finally, it is shown that the lazy decoder is at least as good as any other type of decoder that returns words of any length. In particular, it is superior to any decoder that returns words of shorter length and therefore it is the ML* decoder.

Theorem 7. *For any decoder* $\mathcal{D} : \Sigma_2^{n-1} \to \Sigma_2^*$,

$$P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}, d_L) \geqslant P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}_{\mathrm{Lazy}}, d_L).$$

Thus, for the 1-Del *channel with* Σ_2^n, *the ML** *decoder is* $\mathcal{D}_{\mathrm{Lazy}}$.

Proof: Let $\mathcal{D} : \Sigma_2^{n-1} \to \Sigma_2^*$. By Lemma 6, the theorem holds for any hybrid decoder and therefore we can assume that $\mathcal{D}$ is not a hybrid decoder. Hence, there exists at least one channel output $\boldsymbol{y}'$, such that, $\mathcal{D}(\boldsymbol{y}')$ is neither of length n, nor of length $n-1$. We consider the following two cases.

Case 1: $|\mathcal{D}(\boldsymbol{y}')| \neq n+1$. Thus, $d_L(\mathcal{D}(\boldsymbol{y}'), \boldsymbol{c}) \geqslant 2$ and the inequality can be proved using similar arguments to Lemma 6.

Case 2: $|\mathcal{D}(\boldsymbol{y}')| = n+1$. If $\mathcal{D}(\boldsymbol{y}')$ is not the alternating word, then $|D_1(\mathcal{D}(\boldsymbol{y}'))| \leqslant n$, i.e., there are at most n words of length n that have distance 1 from $\mathcal{D}(\boldsymbol{y}')$. Since $|I_1(\boldsymbol{y}')| = n+1$, there is at least one word $\boldsymbol{c} \in I_1(\boldsymbol{y}')$ such that $d_L(\mathcal{D}(\boldsymbol{y}'), \boldsymbol{c}) > 1$. Using this observation we derive that

$$P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}, d_L) = \frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{y}\in\Sigma_2^{n-1}} \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} \frac{d_L(\mathcal{D}(\boldsymbol{y}), \boldsymbol{c})}{|\boldsymbol{c}|} p(\boldsymbol{y}|\boldsymbol{c})$$
$$\geqslant \frac{1}{n|\mathcal{C}|} \sum_{\substack{\boldsymbol{y}\in\Sigma_2^{n-1} \\ |\mathcal{D}(\boldsymbol{y})|=n-1}} \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}) + \frac{1}{|\mathcal{C}|} \sum_{\substack{\boldsymbol{y}\in\Sigma_2^{n-1} \\ |\mathcal{D}(\boldsymbol{y})|=n}} \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} \frac{d_L(\mathcal{D}(\boldsymbol{y}), \boldsymbol{c})}{|\boldsymbol{c}|} p(\boldsymbol{y}|\boldsymbol{c})$$
$$+ \frac{1}{|\mathcal{C}|} \sum_{\substack{\boldsymbol{y}\in\Sigma_2^{n-1} \\ |\mathcal{D}(\boldsymbol{y})|=n+1}} \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} \frac{d_L(\mathcal{D}(\boldsymbol{y}), \boldsymbol{c})}{|\boldsymbol{c}|} p(\boldsymbol{y}|\boldsymbol{c}) \overset{(a)}{>} \frac{1}{n|\mathcal{C}|} \sum_{\boldsymbol{y}\in\Sigma_2^{n-1}} \sum_{\boldsymbol{c}\in I_1(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{c}) = \frac{1}{n},$$

where the last inequality results from the words $\boldsymbol{y}', \boldsymbol{c}$ which satisfy $d_L(\mathcal{D}(\boldsymbol{y}'), \boldsymbol{c}) > 1$. That is, it is concluded that

$$P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}, d_L) > \frac{1}{n} = P_{\mathrm{err}}(\text{1-Del}, \Sigma_2^n, \mathcal{D}_{\mathrm{Lazy}}, d_L).$$

Note that, for the special case where $\mathcal{D}(\boldsymbol{y}')$ is the alternating sequence of length $n+1$, $|I_1(\boldsymbol{y}')| = |D_1(\mathcal{D}(\boldsymbol{y}'))| = n+1$, which implies that inequality (a) is a weak inequality.

Since $\mathcal{D}_{Lazy}$ minimizes the average decoding error probability when $\mathcal{C} = \Sigma_2^n$, and since the ML* decoder returns a word of minimal length, it follows that $\mathcal{D}_{Lazy}$ is the ML* decoder for the 1-deletion channel. ■

IV. The 2-Deletion Channel

In this section we consider the case of the 2-deletion channel when the code is the entire space, i.e., $\mathcal{C} = \Sigma_2^n$. In this setup, exactly two symbols are selected randomly and deleted from the transmitted word $\boldsymbol{x} \in \Sigma_2^n$. We construct a decoder that is based on $\mathcal{D}_{Lazy}$ and on a variant of $\mathcal{D}_{EN}$ and prove that it minimizes the average decoding error probability. That is, we explicitly find the ML* decoder for the 2-deletion channel.

Before we continue, two more families of decoders are introduced. The *maximum likelihood* of length m*, denoted by $\mathcal{D}^m_{\mathsf{ML}^*}$, is the decoder that for any given channel output $\boldsymbol{y}$ returns a word $\boldsymbol{x}$ of length m that minimizes $f_{\boldsymbol{y}}(\boldsymbol{x})$. That is,

$$\mathcal{D}^m_{\mathsf{ML}^*}(\boldsymbol{y}) = \underset{\boldsymbol{x}\in\Sigma_2^m}{\arg\min}\{f_{\boldsymbol{y}}(\boldsymbol{x})\}.$$

The *embedding number decoder of length m*, denoted by $\mathcal{D}^m_{EN}$, is the decoder that for any given channel output $\boldsymbol{y}$ returns a word $\boldsymbol{x}$ of length m that maximizes the embedding number of $\boldsymbol{y}$ in $\boldsymbol{x}$. That is, for $m \geqslant |\boldsymbol{y}|$,

$$\mathcal{D}^m_{EN}(\boldsymbol{y}) = \underset{\boldsymbol{x}\in\Sigma_2^m}{\arg\max}\{\mathsf{Emb}(\boldsymbol{x};\boldsymbol{y})\}.$$

Similarly to the analysis of the 1-Del channel in Section III, any embedding number decoder prolongs existing runs in the word $\boldsymbol{y}$. It can be shown that any embedding number decoder of length $m > |\boldsymbol{y}|$ prolongs at least one of the longest runs in $\boldsymbol{y}$ by at least one symbol. For simplicity, we assume that in the case where there are two or more longest runs in $\boldsymbol{y}$, $\mathcal{D}^m_{EN}$ for $m > |\boldsymbol{y}|$ necessarily chooses to prolong the first ones. Moreover, if there is more than one option that maximizes the embedding number, $\mathcal{D}^m_{EN}$ will choose the one that prolongs the least number of runs.

In the rest of this section we prove several properties on the ML* decoder for a single 2-deletion channel and lastly in Theorem 14 we construct this decoder explicitly. Unless specified otherwise, we assume that $\mathcal{D}_{\mathsf{ML}^*}$ returns a word with minimum length that minimizes $f_{\boldsymbol{y}}(\mathcal{D}(\boldsymbol{y}))$.

Lemma 8. *For any channel output $\boldsymbol{y} \in \Sigma_2^{n-2}$, it holds that*

$$n-2 \leqslant |\mathcal{D}_{\mathsf{ML}^*}(\boldsymbol{y})| \leqslant n+1.$$

For any channel output $\boldsymbol{y} \in \Sigma_2^{n-2}$, Lemma 8 implies that $|\mathcal{D}_{\mathsf{ML}^*}(\boldsymbol{y})| \in \{n-2, n-1, n, n+1\}$. The following lemma states that $\mathcal{D}^n_{EN}$ is the ML* decoder of length n.

Lemma 9. *For all $n \geqslant 3$, it holds that $\mathcal{D}^n_{\mathsf{ML}^*} = \mathcal{D}^n_{EN}$.*

Let $\rho(\boldsymbol{y}) = k$ be the number of runs in a channel output $\boldsymbol{y}$, and let r_j denote the length of the j-th run in $\boldsymbol{y}$. Additionally, let the i-th, the i'-th run be the first two longest runs in $\boldsymbol{y}$ such that $r_i \geqslant r_{i'}$. It can be verfied that if $r_i \geqslant 2r_{i'}$, then $\mathcal{D}^n_{EN}$ prolongs the i-th run by two symbols. Otherwise, it prolongs the i-th and the i'-th run, each by one symbol [3]. It can be also shown that $\mathcal{D}^{n-2}_{EN} = \mathcal{D}_{Lazy}$ and that $\mathcal{D}^{n-1}_{EN}$ prolongs the i-th run of $\boldsymbol{y}$ by one symbol. This leads to the next lemma.

Lemma 10. *For all $n \geqslant 3$, it holds that*

$$\mathcal{D}^{n-2}_{\mathsf{ML}^*} = \mathcal{D}_{Lazy},\ \ \mathcal{D}^{n-1}_{\mathsf{ML}^*} = \mathcal{D}^{n-1}_{EN}.$$

The next lemma eliminates outputting words of length $n+1$.

Lemma 11. *Let $\boldsymbol{y} \in \Sigma_2^{n-2}$ be a channel output. For any decoder $\mathcal{D}$, such that $\mathcal{D}(\boldsymbol{y})$ is a supersequence of $\boldsymbol{y}$ and $|\mathcal{D}(\boldsymbol{y})| = n+1$, it holds that $f_{\boldsymbol{y}}(\mathcal{D}(\boldsymbol{y})) \geqslant f_{\boldsymbol{y}}(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))$.*

Proof: Note that it is enough to prove that

$$\sum_{\boldsymbol{c}\in I_2(\boldsymbol{y})} \mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) \left(d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) - d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c})\right) \geqslant 0. \quad (4)$$

Let $\rho(\boldsymbol{y}) = k$ be the number of runs in $\boldsymbol{y}$, r_j be the length of the j-th run for $1 \leqslant j \leqslant k$, and assume that the i-th run of $\boldsymbol{y}$ is the first longest run of $\boldsymbol{y}$. For a transmitted word $\boldsymbol{c}$, it holds that $d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c}) \in \{1,3\}$, and $d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) \in \{1,3,5\}$. Recall that $I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y})) \subseteq I_2(\boldsymbol{y})$. $\mathcal{D}(\boldsymbol{y})$ is a supersequence of $\boldsymbol{y}$, and hence $\mathcal{D}(\boldsymbol{y})$ is obtained from $\boldsymbol{y}$ by prolonging existing runs or by creating new runs in $\boldsymbol{y}$. Hence, for every word $\boldsymbol{c} \in I_2(\boldsymbol{y})$ such that $\boldsymbol{c} \notin I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y})) \cup D_1(\mathcal{D}(\boldsymbol{y}))$, it holds that $d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) \geqslant 3$ while $d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c}) = 3$. Additionally, any $\boldsymbol{c} \in I_2(\boldsymbol{y})$ such that $\boldsymbol{c} \in I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y})) \cap D_1(\mathcal{D}(\boldsymbol{y}))$, satisfies $d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c}) = d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) = 1$. Hence, for these words it holds that $d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) - d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c}) \geqslant 0$ and they can be eliminated from inequality (4). To complete the proof, the words $\boldsymbol{c} \in I_2(\boldsymbol{y})$ such that $\boldsymbol{c} \in I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))$, $\boldsymbol{c} \notin D_1(\mathcal{D}(\boldsymbol{y}))$ and the words $\boldsymbol{c} \in I_2(\boldsymbol{y})$ such that $\boldsymbol{c} \notin I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))$, $\boldsymbol{c} \in D_1(\mathcal{D}(\boldsymbol{y}))$ should be considered. For words in the first case it holds that $d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c}) = 1$ and $d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) \geqslant 3$, while for words in the latter case, $d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c}) = 3$ and $d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) \geqslant 1$. Hence,

$$\begin{aligned}&\sum_{\boldsymbol{c}\in I_2(\boldsymbol{y})} \mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) \left(d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) - d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c})\right)\\ &\geqslant 2\sum_{\substack{\boldsymbol{c}\in I_2(\boldsymbol{y})\\ \boldsymbol{c}\in I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))\\ \boldsymbol{c}\notin D_1(\mathcal{D}(\boldsymbol{y}))}} \mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) - 2\sum_{\substack{\boldsymbol{c}\in I_2(\boldsymbol{y})\\ \boldsymbol{c}\notin I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))\\ \boldsymbol{c}\in D_1(\mathcal{D}(\boldsymbol{y}))}} \mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}).\end{aligned}$$

Denote by *Diff* the right hand side of the last inequality.

We first assume that $\mathcal{D}(\boldsymbol{y})$ is obtained from $\boldsymbol{y}$ by prolonging the i-th run by exactly one symbol. Let $\boldsymbol{c} \in I_2(\boldsymbol{y})$ and consider the cases mentioned above.

Case 1: $\boldsymbol{c} \in I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))$ and $\boldsymbol{c} \notin D_1(\mathcal{D}(\boldsymbol{y}))$. Recall that both decoders return supersequences of $\boldsymbol{y}$. By the assumption $\mathcal{D}(\boldsymbol{y})$ is obtained from $\boldsymbol{y}$ by prolonging the i-th run by one symbol and then performing two more insertions to the obtained word. Since $\boldsymbol{c} \in I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))$, $\boldsymbol{c}$ must be obtained from $\boldsymbol{y}$ by prolonging the i-th run and performing one more insertion. $\boldsymbol{c} \notin D_1(\mathcal{D}(\boldsymbol{y}))$, and therefore the number of such words equals to $|I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))| - |I_2(\boldsymbol{y}) \cap I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y})) \cap D_1(\mathcal{D}(\boldsymbol{y}))|$. Note that the size of the right intersection is at most 2, since the words in the this intersection are the words that are obtained from $\boldsymbol{y}$ by prolonging the i-th run by one symbol and then performing one of the two other insertions performed to receive $\mathcal{D}(\boldsymbol{y})$. Hence, there are at least $|I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))| - 2 = n-1$ such words in this case and for each of them $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) \geqslant r_i + 1$. Recall that these words satisfy $d(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c}) = 1$ and $d(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) \geqslant 3$.

Case 2: $\boldsymbol{c} \notin I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))$ and $\boldsymbol{c} \in D_1(\mathcal{D}(\boldsymbol{y}))$. By the assumption, $\mathcal{D}$ prolongs the i-th run by one symbol and performs two more insertions into the obtained word and $\mathcal{D}^{n-1}_{EN}$ prolongs the i-th run by one symbol. Hence, the words $\boldsymbol{c} \in I_2(\boldsymbol{y})$ such that $\boldsymbol{c} \notin I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))$ and $\boldsymbol{c} \in D_1(\mathcal{D}(\boldsymbol{y}))$ can not be obtained from $\boldsymbol{y}$ by prolonging the i-th run. Therefore, it implies that $\boldsymbol{c}$ is the unique word obtained from $\mathcal{D}(\boldsymbol{y})$ by deleting the symbol that was inserted to the i-th run of $\boldsymbol{y}$. It holds that $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) \leqslant (r_i+1)^2$, $d_L(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}),\boldsymbol{c}) = 3$ and $d_L(\mathcal{D}(\boldsymbol{y}),\boldsymbol{c}) = 1$. Since r_i is the length of the i-th run of $\boldsymbol{y}$, $r_i \leqslant |\boldsymbol{y}| = n-2$. Thus, *Diff* $\geqslant 2(r_i+1)^2 - 2(r_i+1)^2 \geqslant 0$.

Second, assume that $\mathcal{D}(\boldsymbol{y})$ is obtained from $\boldsymbol{y}$ by prolonging the i-th run by at least two symbols. It holds that $(D_1(\mathcal{D}(\boldsymbol{y})) \cap I_2(\boldsymbol{y})) \subseteq I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))$, which implies that $|\{\boldsymbol{c} : \boldsymbol{c} \in I_2(\boldsymbol{y}) \cap D_1(\mathcal{D}(\boldsymbol{y})), \boldsymbol{c} \notin I_1(\mathcal{D}^{n-1}_{EN}(\boldsymbol{y}))\}| = 0$, and therefore, *Diff* $\geqslant 0$.

Lastly, assume that $\mathcal{D}(\boldsymbol{y})$ is obtained from $\boldsymbol{y}$ by three insertions such that neither of these insertions prolongs the i-th run. It holds that, $|I_2(\boldsymbol{y}) \cap I_1(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y})) \cap D_1(\mathcal{D}(\boldsymbol{y}))| = 0$. Therefore the number of words $\boldsymbol{c} \in I_2(\boldsymbol{y})$ such that $\boldsymbol{c} \in I_1(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}))$ and $\boldsymbol{c} \notin D_1(\mathcal{D}(\boldsymbol{y}))$ equals to $|I_1(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}))| = n+1$. For any such word $\boldsymbol{c}$ it holds that $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) \geqslant r_i + 1$. Furthermore, $|D_1(\mathcal{D}(\boldsymbol{y}))|$ equals to the number of runs in $\mathcal{D}(\boldsymbol{y})$ [29] and any $\boldsymbol{c} \in D_1(\mathcal{D}(\boldsymbol{y})) \cap I_2(\boldsymbol{y})$ is obtained from $\mathcal{D}(\boldsymbol{y})$ by deleting one symbol of the three insertions to $\boldsymbol{y}$ in order to obtain $\mathcal{D}(\boldsymbol{y})$. Hence, there are at most three such words, and each is obtained by deleting one of the three inserted symbols. Let $\boldsymbol{c}$ be one of those words. If the two remaining symbols belong to the same run, then $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) = \binom{m}{2}$ where m is the length of this run in $\boldsymbol{c}$ and $m \leqslant r_i + 2$. In this case consider the word $\boldsymbol{c}'$ that is obtained by prolonging the i-th run of $\boldsymbol{y}$ by two symbols. It holds that, $\mathsf{Emb}(\boldsymbol{c}';\boldsymbol{y}) = \binom{r_i+2}{2} \geqslant \binom{m}{2} = \mathsf{Emb}(\boldsymbol{c};\boldsymbol{y})$. Otherwise, $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) = m_1 m_2$ where m_1 and m_2 are the lengths of the runs that include the remaining inserted symbols and $m_1, m_2 \leqslant r_i + 1$. Let $\boldsymbol{c}'$ be the word that is obtained from $\boldsymbol{y}$ by prolonging the i-th run and the run of length $\max\{m_1 - 1, m_2 - 1\}$ that is prolonged by $\mathcal{D}$. In this case, $\mathsf{Emb}(\boldsymbol{c}';\boldsymbol{y}) = m_1(r_i+1) \geqslant m_1 m_2 = \mathsf{Emb}(\boldsymbol{c};\boldsymbol{y})$. Note that there is at most one such word $\boldsymbol{c}$ that is obtained by prolonging the same run with two symbols, which implies that there is always a selection of words $\boldsymbol{c}'$ such that *Diff* $\geqslant 0$.

We proved that for any decoder $\mathcal{D}$ such that $\mathcal{D}(\boldsymbol{y})$ is a supersequence $\boldsymbol{y}$ and $|\mathcal{D}(\boldsymbol{y})| = n+1$, *Diff* $\geqslant 0$, and thus, $f_{\boldsymbol{y}}(\mathcal{D}(\boldsymbol{y})) - f_{\boldsymbol{y}}(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y})) \geqslant 0$. ■

Additionally, we show in [3] that for any channel output $\boldsymbol{y} \in \Sigma_2^{n-2}$, when $|\mathcal{D}(\boldsymbol{y})| = n$, $f_{\boldsymbol{y}}(\mathcal{D}_{Lazy}(\boldsymbol{y})) \leqslant f_{\boldsymbol{y}}(\mathcal{D}(\boldsymbol{y}))$ and therefore $\mathcal{D}_{\mathsf{ML}^*}(\boldsymbol{y})$ cannot be a word of length n. Lemma 11 states that $\mathcal{D}_{EN}^{n-1}$ is preferable over any decoder that returns only words of length $n+1$ and therefore an immediate conclusion is given in the following corollary.

Corollary 12. *The ML* decoder for the 2-deletion channel, returns words of length either $n-2$ or $n-1$.*

In the following result we define a condition on the length of the longest run in $\boldsymbol{y}$ to decide whether prolonging it by one symbol can minimize the average decoding error probability. This result proves Theorem 14 which determines the ML* decoder for the case of a single 2-deletion channel.

Lemma 13. *Let $\boldsymbol{y} \in \Sigma_2^{n-2}$ be a channel output, such that the number of runs in $\boldsymbol{y}$ is $\rho(\boldsymbol{y}) = k$, and the first longest run in $\boldsymbol{y}$ is the i-th run. Denote by r_j the length of the j-th run in $\boldsymbol{y}$ for $1 \leqslant j \leqslant k$. It holds that $f_{\boldsymbol{y}}(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y})) \geqslant f_{\boldsymbol{y}}(\mathcal{D}_{Lazy}(\boldsymbol{y}))$ if and only if $2n^2 - 4nr_i - 6n + r_i^2 + 3r_i + k + 1 \geqslant 0$.*

Proof: The decoder $\mathcal{D}_{EN}^{n-1}$ prolongs the i-th run of $\boldsymbol{y}$ by one symbol. Thus, the Levenshtein distance between $\mathcal{D}_{EN}^{n-1}(\boldsymbol{y})$ and the transmitted word $\boldsymbol{c}$ can be either 1 or 3, and hence,

$$f_{\boldsymbol{y}}(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y})) - f_{\boldsymbol{y}}(\mathcal{D}_{Lazy}(\boldsymbol{y})) = \sum_{\substack{\boldsymbol{c} \in I_2(\boldsymbol{y}) \\ d_L(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}),\boldsymbol{c})=3}} \frac{p(\boldsymbol{y}|\boldsymbol{c})}{n} - \sum_{\substack{\boldsymbol{c} \in I_2(\boldsymbol{y}) \\ d_L(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}),\boldsymbol{c})=1}} \frac{p(\boldsymbol{y}|\boldsymbol{c})}{n}.$$

Denote

$$\mathcal{S}um_3 \triangleq \sum_{\substack{\boldsymbol{c} \in I_2(\boldsymbol{y}) \\ d_L(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}),\boldsymbol{c})=3}} p(\boldsymbol{y}|\boldsymbol{c}), \qquad \mathcal{S}um_1 \triangleq \sum_{\substack{\boldsymbol{c} \in I_2(\boldsymbol{y}) \\ d_L(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}),\boldsymbol{c})=1}} p(\boldsymbol{y}|\boldsymbol{c}).$$

Let us prove that $2n^2 - 4nr_i - 6n + r_i^2 + 3r_i + k + 1 \geqslant 0$ is a necessary and sufficient condition for the inequality $\mathcal{S}um_3 \geqslant \mathcal{S}um_1$ to hold. First, we count the number of words $\boldsymbol{c} \in I_2(\boldsymbol{y})$ such that $d_L(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}),\boldsymbol{c}) = 1$. Each such $\boldsymbol{c}$ is a supersequence of $\mathcal{D}_{EN}^{n-1}(\boldsymbol{y})$ and therefore $\boldsymbol{c}$ can be obtained from $\boldsymbol{y}$ only by one of the three following options. The first option is by prolonging the i-th run and the j-th of $\boldsymbol{y}$ for $j \neq i$, each by one symbol. The number of such words is $k-1$. The second option is by prolonging the i-th run in $\boldsymbol{y}$ by one symbol and creating a new run in $\boldsymbol{y}$. The number of options to create a new run in $\boldsymbol{y}$ is $n-k+1$ and therefore, there are $n-k+1$ such words. The third option is by prolonging the i-th run by two symbols and there is only one such a word. Hence, the total number of words $\boldsymbol{c} \in I_2(\boldsymbol{y})$ such that $d_L(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}),\boldsymbol{c}) = 1$ is $n+1 = |I_1(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}))|$. Among them, the $k-1$ words that are obtained by the first option have an embedding number of $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) = (r_i+1)(r_j+1)$. Similarly the $n-k+1$ words that are obtained from $\boldsymbol{y}$ using the second option satisfy $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) = r_i + 1$. Lastly, for the word $\boldsymbol{c}$ that is obtained by prolonging the i-th run of $\boldsymbol{y}$ by two symbols, it holds that $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) = \binom{r_i+2}{2}$. Hence,

$$\begin{aligned}\mathcal{S}um_1 &= \sum_{\substack{\boldsymbol{c} \in I_2(\boldsymbol{y}) \\ d_L(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}),\boldsymbol{c})=1}} p(\boldsymbol{y}|\boldsymbol{c}) = \frac{\binom{r_i+2}{2}}{\binom{n}{2}} + \sum_{\substack{1 \leqslant j \leqslant k \\ j \neq i}} \frac{(r_i+1)(r_j+1)}{\binom{n}{2}} + \sum_{j=1}^{n-k+1} \frac{(r_i+1)}{\binom{n}{2}} \\ &= \frac{(2n - \frac{r_i}{2} - 1) \cdot (r_i+1)}{\binom{n}{2}} = \frac{(4n - r_i - 2) \cdot (r_i + 1)}{n \cdot (n-1)},\end{aligned}$$

Next, let us evaluate the summation $\mathcal{S}um_3$. Note that if $d_L(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}),\boldsymbol{c}) = 3$ then $\boldsymbol{c}$ is not in a supersequence of $\mathcal{D}_{EN}^{n-1}(\boldsymbol{y})$, and hence $\boldsymbol{c} \notin I_1(\mathcal{D}_{EN}^{n-1}(\boldsymbol{y}))$. The words that contribute to the summation $\mathcal{S}um_3$ can be divided into three different types of words $\boldsymbol{c} \in I_2(\boldsymbol{y})$.

Case 1: Let $\mathcal{C}_1 \subseteq I_2(\boldsymbol{y})$ be the set of words that includes additional run(s) that do not appear in $\boldsymbol{y}$. Such additional runs can be either one run of length 2, or two runs of length 1 each. The number of words such that the length of the new run is two is $n-k$, and the number of words with two additional runs is $\binom{n-k}{2}$. Additionally, for $\boldsymbol{c} \in \mathcal{C}_1$, $\mathsf{Emb}(\boldsymbol{c};\boldsymbol{y}) = 1$, which implies, $\sum_{\boldsymbol{c} \in \mathcal{C}_1} p(\boldsymbol{y}|\boldsymbol{c}) = \frac{|\mathcal{C}_1|}{\binom{n}{2}} = \dfrac{(n-k)(n-k+1)}{n(n-1)}$.

Case 2: Let $\mathcal{C}_2 \subseteq I_2(\boldsymbol{y})$ be the set of words that are obtained from $\boldsymbol{y}$ by prolonging the j-th run and by creating a new run in $\boldsymbol{y}$. By using similar arguments as in the previous case it can be shown that, $\sum_{\boldsymbol{c} \in \mathcal{C}_2} p(\boldsymbol{y}|\boldsymbol{c}) = \dfrac{2(n-k+1)}{n(n-1)}(n - r_i + k - 3)$.

Case 3: Let $\mathcal{C}_3 \subseteq I_2(\boldsymbol{y})$ be the set of words that are obtained from $\boldsymbol{y}$ by prolonging one or two existing runs in $\boldsymbol{y}$ (other than the i-th run). It can be shown that $\sum_{\boldsymbol{c} \in \mathcal{C}_3} p(\boldsymbol{y}|\boldsymbol{c}) = \dfrac{(n - r_i + k - 3)(n - r_i + k - 2)}{n(n-1)}$.

Thus, $\mathcal{S}um_3 = \dfrac{4n^2 - 4nr_i - 8n + r_i^2 + 3r_i + 2k}{n(n-1)}$, and $\mathcal{S}um_3 - \mathcal{S}um_1 \geqslant 0$ if and only if

$$2n^2 - 4nr_i - 6n + r_i^2 + 3r_i + k + 1 \geqslant 0. \qquad ■$$

Using this result we can explicitly define the ML* decoder $\mathcal{D}_{\mathsf{ML}^*}$. This decoder works as follows. For each word $\boldsymbol{y}$ it calculates $\rho(\boldsymbol{y}) = k$ and r_i and then checks if

$$2n^2 - 4nr_i - 6n + r_i^2 + 3r_i + k + 1 \geqslant 0. \tag{5}$$

If (5) holds, the decoder works as $\mathcal{D}_{Lazy}$. Otherwise, it acts like $\mathcal{D}_{EN}^{n-1}$ and prolongs the first longest run by one. This result is summarized in the following theorem.

Theorem 14. *The ML* decoder $\mathcal{D}_{\mathsf{ML}^*}$ for the* 2-Del *channel performs as the lazy decoder if inequality (5) holds and otherwise as the embedding number decoder of length $n-1$, i.e.,*

$$\mathcal{D}_{\mathsf{ML}^*}(\boldsymbol{y}) = \begin{cases} \mathcal{D}_{Lazy}(\boldsymbol{y}) & \textit{inequality (5) holds}, \\ \mathcal{D}_{EN}^{n-1}(\boldsymbol{y}) & \textit{otherwise.} \end{cases}$$

By Theorem 14 if $\mathcal{D}_{\mathsf{ML}^*}(\boldsymbol{y}) = \mathcal{D}_{EN}^{n-1}(\boldsymbol{y})$ then (5) does not hold, and it can be shown that $r_i \geqslant (2 - \sqrt{2})n \approx 0.586n$. Thus, in almost all cases $\mathcal{D}_{\mathsf{ML}^*}(\boldsymbol{y}) = \mathcal{D}_{Lazy}(\boldsymbol{y})$.

REFERENCES

[1] M. Abroshan, R. Venkataramanan, L. Dolecek, and A. G. i Fàbregas. "Coding for deletion channels with multiple traces," *IEEE International Symposium on Information Theory*, pp. 1372-1376, 2019.

[2] A. Atashpendar, M. Beunardeau, A. Connolly, R. Géraud, D. Mestel, A. W. Roscoe, and P. Y. A. Ryan. "From clustering supersequences to entropy minimizing subsequences for single and double deletions," *arXiv preprint:1802.00703*, 2018.

[3] D. Bar-Lev, Y. Gershon, O. Sabary, and E. Yaakobi. "The error probability of maximum likelihood decoding for the t-deletion channel," https://www.omersabary.com/publication/2021MLdectdel, 2021.

[4] M. Blawat, K. Gaedke, I. Hütter, X.-M. Chen, B. Turczyk, S. Inverso, B.W. Pruitt, and G.M. Church, "Forward error correction for DNA data storage," *Int. Conf. on Computational Science*, vol. 80, pp. 1011–1022, 2016.

[5] J. Brakensiek, V. Guruswami, and S. Zbarsky, "Efficient low-redundancy codes for correcting multiple deletions," *Proc. of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1884–1892, Philadelphia, PA, USA, 2016.

[6] J.A. Briffa, V. Buttigieg, and S. Wesemeyer, "Time-varying block codes for synchronization errors: MAP decoder and practical issues," arXiv:1802.08595, Feb. 2018.

[7] B. Bukh, and V. Guruswami and J. Håstad, "An improved bound on the fraction of correctable deletions," *IEEE Trans. on Inform. Theory*, vol. 63, no. 1, pp. 93–103, Jan. 2017.

[8] V. Bhardwaj, P. A. Pevzner, C. Rashtchian and Y. Safonova, "Trace Reconstruction Problems in Computational Biology," *IEEE Transactions on Information Theory*, doi: 10.1109/TIT.2020.3030569, 2020.

[9] J. Castiglione and A. Kavcic, "Trellis based lower bounds on capacities of channels with synchronization errors, *Information Theory Workshop*, Jeju, South Korea, pp. 24–28, 2015.

[10] M. Cheraghchi, "Capacity upper bounds for deletion-type channels," *Journal of the ACM*, vol. 66, no. 2, p. 9, 2019.

[11] R. Con and A. Shpilka, "Explicit and efficient constructions of coding schemes for the binary deletion channel and the Poisson repeat channel," arXiv:1909.10177, Sep 2019.

[12] M. Dalai, "A new bound on the capacity of the binary deletion channel with high deletion probabilities," *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, pp. 499–502, Jun. 2011.

[13] S. Diggavi and M. Grossglauser, "On information transmission over a finite buffer channel," IEEE Transactions on Information Theory, vol. 52, no. 3, pp. 1226–1237, March 2006.

[14] L. Dolecek and V. Anantharam, "Using Reed?Muller RM $(1, m)$ codes over channels with synchronization and substitution errors,'" *IEEE Trans. on Inform. Theory*, vol. 53, no. 4, pp. 1430–1443, Apr. 2007.

[15] E. Drinea and M. Mitzenmacher, "Improved lower bounds for the capacity of iid deletion and duplication channels," IEEE Transactions on Information Theory, vol. 53, no. 8, pp. 2693–2714, 2007.

[16] C. Elzinga, S. Rahmann, and H. Wang. Algorithms for subsequence combinatorics. *Theoretical Computer Science*, 409(3):394–404, 2008.

[17] D. Fertonani and T. M. Duman, "Novel bounds on the capacity of the binary deletion channel," IEEE Transactions on Information Theory, vol. 56, no. 6, pp. 2753–2765, 2010.

[18] R. Gabrys and F. Sala, "Codes correcting two deletions," *IEEE Trans. on Inform. Theory*, vol. 65, no. 2, pp. 965–974, Feb. 2018.

[19] B. Graham. A Binary Deletion Channel With a Fixed Number of Deletions. Combinatorics, Probability and Computing, 24(3), 486-489.2018.

[20] V. Guruswami and J. Hastad, "Explicit two-deletion codes with redundancy matching the existential bound," *IEEE Transactions on Information Theory*, doi: 10.1109/TIT.2021.3069446.

[21] V. Guruswami, B. Haeupler, and A. Shahrasbi, "Optimally resilient codes for list-decoding from insertions and deletions," *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 524–537, 2020.

[22] V. Guruswami and C. Wang, "Deletion codes in the high-noise and high-rate regimes," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 1961–1970, 2017.

[23] B. Haeupler, A. Shahrasbi, and M. Sudan, "Synchronization strings: List decoding for insertions and deletions," https://arxiv.org/abs/1802.08663, 2018.

[24] T. Hayashi and K. Yasunaga, "On the list decodability of insertions and deletions," *Int. Symp. Inform. Theory*, pp. 86–90, 2018.

[25] R. Heckel, G. Mikutis, and R.N. Grass, "A characterization of the DNA data storage channel," arxiv.org/pdf/1803.03322.pdf, 2018.

[26] S. Kas Hanna and S. El Rouayheb, "List decoding of deletions using guess & check codes," *Int. Symp. Inform. Theory*, pp. 2374–2378, 2019.

[27] A. Kirsch and E. Drinea, "Directly lower bounding the information capacity for channels with i.i.d. deletions and duplications," IEEE Transactions on Information Theory, vol. 56, no. 1, pp. 86–102, Jan. 2010.

[28] S. Kosuri and G.M. Church, "Large-scale de novo DNA synthesis: technologies and applications," *Nature Methods*, vol. 11, no. 5, pp. 499–507, May 2014.

[29] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, 10(8):707–710, 1966.

[30] S. Liu, I. Tjuawinata, and C. Xing, "On list decoding of insertion and deletion errors," https://arxiv.org/abs/1906.09705, 2019.

[31] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probability Surveys*, vol. 6, pp. 1–33, 2009.

[32] M. Mitzenmacher and E. Drinea, "A simple lower bound for the capacity of the deletion channel," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4657–4660, 2006.

[33] L. Organick, S.D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M.Z. Racz, G. Kamath, P. Gopalan, B. Nguyen, C. Takahashi, S. Newman, H.-Y. Parker, C. Rashtchian, K. Stewart, G. Gupta, R. Carlson, J. Mulligan, D. Carmean, G. Seelig, L. Ceze, and K. Strauss, "Scaling up DNA data storage and random access retrieval," *bioRxiv*, Mar. 2017.

[34] M. Rahmati and T. M. Duman, "Upper bounds on the capacity of deletion channels using channel fragmentation," *IEEE Trans. Inform. Theory*, vol. 61, no. 1, pp. 146–156, 2015.

[35] M.G. Ross, C. Russ, M. Costello, A. Hollinger, N.J. Lennon, R. Hegarty, N. Nusbaum, and D.B. Jaffe, "Characterizing and measuring bias in sequence data," *Genome Biol.*, vol. 14, no. 5, R51, 2013.

[36] O. Sabary, Y. Orlev, R. Shafir L. Anavy, E. Yaakobi, Z.Yakhini, "SOLQC: Synthetic oligo library quality control Tool," *bioRxiv*, Nov. 2019.

[37] O. Sabary, E. Yaakobi and A. Yucovich, "The Error Probability of Maximum-Likelihood Decoding over Two Deletion/Insertion Channels," *IEEE International Symposium on Information Theory (ISIT)*, pp. 763-768, doi: 10.1109/ISIT44484.2020.9174488, 2020.

[38] O. Sabary, A. Yucovich, G. Shapira, E. Yaakobi, "Reconstruction Algorithms for DNA-Storage Systems," *bioRxiv* 2020.09.16.300186, 2020.

[39] F. Sala, C. Schoeny, N. Bitouzé, and L. Dolecek, "Synchronizing files from a large number of insertions and deletions," *IEEE Trans. on Comm.*, vol. 64, no. 6, pp. 2258–2273, June 2016.

[40] J. Sima and J. Bruck, "Optimal k-deletion correcting codes," *Proc. IEEE Int. Symp. Inf. Theory*, Paris, France, pp. 847–851, Jul. 2019.

[41] J. Sima, N. Raviv, and J. Bruck, "On coding over sliced information," http://arxiv.org/abs/1809.02716, 2018.

[42] S. R. Srinivasavaradhan, M. Du, S. Diggavi, and C. Fragouli, "On maximum likelihood reconstruction over multiple deletion channels." *IEEE International Symposium on Information Theory (ISIT)*, pp. 436–440, 2018.

[43] S. R. Srinivasavaradhan, M. Du, S. Diggavi, and C. Fragouli, "Algorithms for reconstruction over single and multiple deletion channels," *IEEE Transactions on Information Theory*, doi: 10.1109/TIT.2020.3033513, 2020.

[44] I. Tal, H. D. Pfister, A. Fazeli and A. Vardy, "Polar codes for the deletion channel: weak and strong polarization," https://arxiv.org/abs/1904.13385.

[45] R. R. Varshamov and G. M. Tenenholtz. A code for correcting a single asymmetric error. *Automatica i Telemekhanika*, 26(2):288–292, 1965.

[46] R. Venkataramanan, S. Tatikonda, and K. Ramchandran, "Achievable rates for channels with deletions and insertions," IEEE Transactions on Information Theory, vol. 59, no. 11, pp. 6990–7013, 2013.

[47] A. Wachter-Zeh, "List decoding of insertions and deletions," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6297–6304, 2017.

[48] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific Reports*, vol. 7, sp. 5011, Jul. 2017.

[49] A.K.-Y. Yim, A.C.-S. Yu, J.-W. Li, A.I.-C. Wong, J.F.C. Loo, K.M. Chan, S.K. Kong, and T.-F. Chan, "The Eesential component in DNA-based information storage system: Robust error-uolerating module," *Frontiers in Bioengineering and Biotechnology* vol. 2, pp. 1–5, Nov. 2014.