On Unique Error Patterns in the Levenshtein's Sequence Reconstruction Model

Ville Junnila, Tero Laihonen and Tuomo Lehtilä

***Abstract*—In the Levenshtein's sequence reconstruction problem a codeword is transmitted through N channels and in each channel a set of errors is introduced to the transmitted word. In previous works, the restriction that each channel provides a unique output word has been essential. In this work, we assume only that each channel introduces a unique set of errors to the transmitted word and hence, some output words can also be identical. As we will discuss, this interpretation is both natural and useful for deletion and insertion errors. We give properties, techniques and (optimal) results for this situation.**

Quaternary alphabets are relevant due to applications related to DNA-memories. Hence, we introduce an efficient Las Vegas style decoding algorithm for simultaneous insertion, deletion and substitution errors in q-ary Hamming spaces for $q \geq 4$.

Keywords: Information Retrieval, DNA-memory, Levenshtein's Sequence Reconstruction, Decoding Algorithm, Substitution Errors, Deletion Errors, Insertion Errors.

I. Introduction

We study *Levenshtein's sequence reconstruction problem* introduced in [2]. In particular, we consider insertion, deletion and substitution errors in q-ary Hamming spaces. The topic has been widely studied during recent years [3]–[11]. Levenshtein's original motivation came from molecular biology and chemistry, where adding redundancy was not feasible. Recently, Levenshtein's problem has returned to the limelight with the rise of advanced memory storage technologies such as associative memories [3], racetrack memories [12] and, especially, DNA-memories [4], where the information is stored to DNA-strands. In the information retrieval process from the DNA-memories, due to biotechnological limitations [4], multiple (possibly) erroneous strands are obtained, which makes Levenshtein's model suitable for this topic. Another interesting aspect, inspired by the DNA-memories, is the emphasis on information based on a quaternary alphabet, instead of the usual binary one, due to the four types of nucleotides in which the information is stored (see [13]–[17] for more information about DNA-memories).

A. Notation

For integers a and b with $a \leq b$, we will denote the set $\{a, a+1, \ldots, b\}$ by $[a, b]$. The cardinality of a set S is denoted by $|S|$. Note that when S is a multiset, $|S|$ is interpreted as the total number of elements in S including the multiplicities. We denote the *q-ary n-dimensional Hamming space* by $\mathbb{Z}_q^n$, where $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$. For a word $\mathbf{w} \in \mathbb{Z}_n^q$, we write $\mathbf{w} = w_1 w_2 \cdots w_n$ where each $w_i \in [0, q-1]$. The *support* of a word $\mathbf{w} = w_1 \cdots w_n \in \mathbb{Z}_q^n$ is defined as $\operatorname{supp}(\mathbf{w}) = \{i \mid w_i \neq 0\}$, the *weight* of $\mathbf{w}$ as $w(\mathbf{w}) = |\operatorname{supp}(\mathbf{w})|$ and the *Hamming distance* between $\mathbf{w}$ and $\mathbf{z}$ as $d(\mathbf{w}, \mathbf{z}) = w(\mathbf{w} - \mathbf{z})$. The *Hamming ball of radius t centered at $\mathbf{w}$* is denoted by $B_t(\mathbf{w}) = \{\mathbf{z} \in \mathbb{Z}_q^n \mid d(\mathbf{w}, \mathbf{z}) \leq t\}$. It is immediate that $|B_t(\mathbf{w})| = V_q(n, t) = \sum_{i=0}^{t} (q-1)^i \binom{n}{i}$. A *code* C is a nonempty subset of $\mathbb{Z}_q^n$. Moreover, the notation a^j means j consecutive symbols a. These notations can be concatenated in an intuitive manner; for example, $0^i 1 0^j$ denotes a binary word of length $i+1+j$ of weight 1 where the single symbol 1 is in the $(i+1)$th coordinate position. Finally, we denote the *all-zero word* $00\cdots0 \in \mathbb{Z}_q^n$ by $\mathbf{0}$ or 0^n.

B. Preliminaries

In this subsection, we introduce the traditional Levenshtein's channel model and some related terminology. In a *substitution error* a symbol in some coordinate position of a word $\mathbf{w} \in \mathbb{Z}_q^n$ is substituted with another symbol, in an *insertion error* a new symbol is inserted to the original word leading to a word of length $n+1$ and in a *deletion error* a symbol is deleted from the original word leading to a word of length $n-1$. Each of these three types of errors is relevant for DNA-memories [18].

For the rest of the paper, we assume the following: $C \subseteq \mathbb{Z}_q^n$ is a code, a *transmitted word* $\mathbf{x} \in C$ is sent through N channels in which insertion, deletion and substitution errors may occur and the number of each type of error is limited by some constant t_i, t_d or t_s, respectively. When the error type is clear from the context, we drop the subscript. Usually, in previous works, it has been assumed that each channel gives a different output word [2]–[4], [9], [19]. We refer to this model as a *traditional* Levenshtein's channel model. It originates from [2] and we define it next.

Definition 1. In the *traditional Levenshtein's model*, given a code $C \subseteq \mathbb{Z}_q^n$, we transmit a word $\mathbf{x} \in C$ through N channels and obtain an output word $\mathbf{y}_i$ from each channel for $i \in [1, N]$. The output of each channel is unique and for each error type (e.g., substitution, deletion and insertion), the number of errors which may occur in a channel is bounded by some parameters. The set of output words is denoted by $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$.

We denote by $T(Y) \subseteq C$ the set of words such that if $\mathbf{x}' \in T(Y)$, then the output set Y can be obtained when $\mathbf{x}'$ is sent through the N channels. Given C, the maximum possible size of $T(Y)$, over all possible transmitted words $\mathbf{x} \in C$, is denoted by $\mathcal{L}$.

The authors were funded in part by the Research Council of Finland grants 338797 and 358718. A shorter version of this article was presented in ISIT2023 [1].

The authors are with the Department of Mathematics and Statistics, University of Turku, Finland (e-mail: viljun@utu.fi, terolai@utu.fi, tualeh@utu.fi). Some of the work of the third author was performed at the Department of Computer Science, University of Helsinki, Finland.

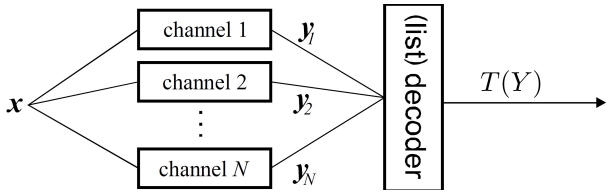

Fig. 1. The Levenshtein's sequence reconstruction.

Note that when the number of channels N increases, the size of $T(Y)$ decreases (or, more precisely, does not increase). In particular, we are especially interested in the required value for N (if it exists) to have $\mathcal{L} = 1$, that is, the number of channels N required for deducing the transmitted word $\mathbf{x}$ unambiguously. From the DNA-memory perspective, this number of channels N is the number of obtained DNA-strands which guarantees correctly deducing the stored DNA-strand. The channel model is illustrated in Figure 1.

C. Related research

For the traditional Levenshtein's model, when only substitution errors may occur in channels, the maximum size $\mathcal{L}$ of $T(Y)$ has been studied in [2], [3], [8], [9] and it is well-understood, for example, when $\mathcal{L}$ is a constant. When only deletion errors occur in channels, if $C = \mathbb{Z}_2^n$, $\mathcal{L} = 1$ and exactly t deletion errors occur in a channel, then the required number of channels is given in the following theorem, which combines Lemma 2 and Equation (36) of [19].

Theorem 2 ([19])**.** *Let exactly $t \leq n-2$ deletion errors occur in the traditional Levenshtein's channel model and $C = \mathbb{Z}_2^n$. Then $\mathcal{L} = 1$ if and only if $N \geq 2\sum_{i=0}^{t-1} \binom{n-t-1}{i} + 1$.*

The previous theorem gives the exact value for N when $q = 2$. However, Levenshtein gave the number of channels also for the cases with $q > 2$. When $q = 3$ an exact number is presented in [19] but when $q \geq 4$, we know only a recursive formulation. Note that the case with $q = 4$ has special relevance due to DNA related applications. In the following theorem, which is based on Equation (51) and Theorem 3 of [19], the same question is discussed in the case of insertion errors.

Theorem 3 ([19])**.** *Let exactly t insertion errors occur in the traditional Levenshtein's channel model and $C = \mathbb{Z}_q^n$. Then $\mathcal{L} = 1$ if and only if $N \geq \sum_{i=0}^{t-1} \binom{n+t}{i}(q-1)^i(1-(-1)^{t-i})+1$.*

The relationship between the number of channels and the list size $\mathcal{L}$ when either deletion or insertion errors occur in channels has been recently studied in [20] for $C = \mathbb{Z}_q^n$. Combinations of different error types have been studied in [4], [21] but they are not well-understood. Moreover, when only deletion or insertion errors occur, there are open problems, for example, when the code $C \neq \mathbb{Z}_q^n$ or the list size $\mathcal{L} > 1$, [4]–[7], [19]. Decoding algorithms for substitution errors have been studied in [2]–[4] and for deletion and insertion errors in [4], [6], [19].

A similar problem has also been considered, in some cases under the name *trace reconstruction*, when each deletion/insertion/substitution error has an independent probability to occur, that is, the maximum number of errors in a channel is not fixed unlike in the traditional model and our models, which will be introduced in Section II. For example, with deletion errors, this would mean that each symbol has an independent probability of ρ to be deleted in a channel. So, if only deletion errors occur, the chance to obtain the empty word is ρ^n; see, for example, [22]–[25].

D. Structure of the paper

The structure of this paper is as follows: In Section II, we introduce the reconstruction problem with two new models when a unique pattern of errors occurs in each channel. Then, in Section III, we consider the new models for deletion errors. In Section IV we continue by giving a fast online decoding algorithm when insertions, deletions and substitutions occur in channels. It requires only a linear time on the total length of (read) output words and does not always need to read every output word. It is likely that the algorithm returns a nonempty output, and the output of the algorithm is *always correct* when it is nonempty.

II. Error patterns

In this section, we introduce two new channel models and compare them to the traditional channel model. We start by discussing the challenges of the traditional model and then continue to introduce our first new model, called multiset model, after which we introduce our second new model, called non-multiset model.

A. Levenshtein's traditional model

The Levenshtein's traditional channel model has usually been considered under the assumption that every channel gives a unique output word [4], [9], [19]. In the case of deletion and insertion errors, this assumption of the model significantly restricts the number of channels (depending on the transmitted word). In what follows, our focus is on deletion errors, but the ideas also generalize to other error types. Furthermore, for deletion errors and any $n, t \geq 1$, there exist (as pointed out in [19]) cases in which we cannot deduce the transmitted word. This is also illustrated in the next example.

Example 4. Let $\mathbf{x} = 111100$ and the number of deletions be *exactly* $t = 1$. If we require that each channel gives a unique output word, then we can have only two channels and $Y \subseteq B_1^d(\mathbf{x}) \setminus \{\mathbf{x}\} = \{11100, 11110\}$ where $B_1^d(\mathbf{x})$ denotes the *deletion ball of radius* 1, that is, the set of words which can be obtained from $\mathbf{x}$ by removing at most 1 symbol. Moreover, if we consider a word $\mathbf{x}' = 111010$, then $Y' \subseteq B_1^d(\mathbf{x}') \setminus \{\mathbf{x}'\} = \{11010, 11110, 11100, 11101\}$. In particular, $B_1^d(\mathbf{x}) \setminus \{\mathbf{x}\} \subset B_1^d(\mathbf{x}') \setminus \{\mathbf{x}'\}$. Consequently, if we only know the set Y, then we cannot say in the traditional model with certainty whether the transmitted word is $\mathbf{x}$ or $\mathbf{x}'$!

Another challenge for the traditional model is, as we see above, that for deletion balls we may have $|B_1^d(\mathbf{x}) \setminus \{\mathbf{x}\}| \neq |B_1^d(\mathbf{x}') \setminus \{\mathbf{x}'\}|$. In general, the size of the deletion ball depends

on the choice of the central word [26]. In the following subsections, we will introduce two new models (called a multiset model and a non-multiset model) which will help us with the challenges of the traditional Levenshtein's model. We also discuss why these models are natural for a wide range of parameters.

B. Multiset model

Instead of assuming that every channel gives a unique output word, we will instead consider Levenshtein's channel model with the assumption that each channel introduces a unique set of errors, which we later call a *unique error pattern*, to the transmitted word $\mathbf{x} \in \mathbb{Z}_q^n$. In particular, we concentrate here on deletion errors but similar ideas and definitions work also for other types of errors (see Subsection II-E).

In the case of deletion errors, a unique error pattern is represented by a *deletion vector* $\mathbf{d} \in \mathbb{Z}_2^n$ in which $\text{supp}(\mathbf{d})$ indicates the coordinate positions, where the deletions occur in the transmitted word $\mathbf{x}$. Thus, the output word $\mathbf{y}$ of a channel belongs to $\mathbb{Z}_q^{n-w(\mathbf{d})}$. There are exactly $\binom{n}{t}$ possible deletion vectors when exactly t errors occur and $V_2(n,t)$ possible deletion vectors when at most t errors occur.

Definition 5. In the *multiset error pattern model for deletion errors*, given a code $C \subseteq \mathbb{Z}_q^n$, we transmit a word $\mathbf{x} \in C$ through N channels and in each channel a unique deletion vector of weight at most t is applied to the codeword $\mathbf{x}$. We obtain a multiset Y_m containing N output words $\mathbf{y}_i$, where $1 \leq i \leq N$.

We denote by $T(Y_m) \subseteq C$ the set of words such that if $\mathbf{x}' \in T(Y_m)$, then the output multiset Y_m can be obtained when $\mathbf{x}'$ is sent through the N channels. Given C, the maximum possible size of $T(Y_m)$, over all possible transmitted words $\mathbf{x} \in C$, is denoted by $\mathcal{L}_m$.

The case with a unique error pattern in each channel can be considered as a *generalization* of the traditional model with unique output words. Indeed, if each output word is unique, then we have applied a unique error pattern in every channel. Moreover, if we assume that we have an output (multi)set Y_m in which a different set of errors has occurred to every output word and two output words can be identical, then we could just prune this multiset Y_m into a non-multiset by removing the extra copies. In other words, the error pattern model also contains the information we have in the traditional model.

Compared to the situation of the traditional model in Example 4, the concept of unique deletion vectors gives new insights into the previously mentioned challenges as we can see in the following example.

Example 6. If we consider $\mathbf{x} = 111100$ and $\mathbf{x}' = 111010$ from Example 4 and apply different deletion vectors of weight exactly 1 to them, then the *multiset* of output words obtained from $\mathbf{x}$ is $Y = \{11100, 11100, 11100, 11100, 11110, 11110\}$ while the multiset of output words obtained from $\mathbf{x}'$ is $Y' = \{11010, 11010, 11010, 11110, 11100, 11101\}$. Since the multisets differ, we can now clearly distinguish between $\mathbf{x}$ and $\mathbf{x}'$ unlike in the traditional model. Moreover, it would be straightforward to verify that the multisets of output words Y and Y' are unique for $\mathbf{x}$ and $\mathbf{x}'$, respectively; in fact, this follows by Theorem 9 since their sizes are at least 5 and $V_2(6,1) - V(2,1) + 1 = 5$.

Furthermore, the unique deletion vectors in our multiset model seem more natural when we consider this problem from a probabilistic perspective. Indeed, if we only assume that each *unique* output word is equally likely to occur, then with $\mathbf{x} = 111100$ and $t = 1$ both output words 11100 and 11110 have equal probability of 50% to occur. However, if each deletion vector of weight 1 has equal probability, then 11100 has approximately 67% probability and 11110 has approximately 33% probability which seems a more natural result.

C. Non-multiset model

Besides the multiset model which requires that each error pattern occurring in a channel is unique, we also give another approach which does not cause challenges if some error patterns are identical as long as a predefined number of different error patterns occurs. We note that the probability to obtain a predefined number of different error patterns can be increased by increasing the number of channels (for more details, see Subsection II-F). We may assume that a unique set of errors occurs in each channel but instead of considering the multiset of output words we consider the set of output words. We call this model a *non-multiset error pattern* model which we define next for deletion errors (see Section II-E for other error types).

Definition 7. In the *non-multiset error pattern model for deletion errors*, given a code $C \subseteq \mathbb{Z}_q^n$, we transmit a word $\mathbf{x} \in C$ through N channels and in each channel a unique deletion vector of weight at most t is applied to word $\mathbf{x}$. We obtain a set Y containing $N_n \leq N$ words $\mathbf{y}_i$, $1 \leq i \leq N_n$.

We denote by $T(Y) \subseteq C$ the set of words such that if $\mathbf{x}' \in T(Y)$, then the output set Y can be obtained when $\mathbf{x}'$ is sent through the N channels. Given C, the maximum possible size of $T(Y)$, over all possible transmitted words $\mathbf{x} \in C$, is denoted by $\mathcal{L}$.

For the non-multiset model to work, we only require that in N' channels we have unique error patterns, that is, we may utilize $N_s \geq N'$ channels and have identical error patterns in some of the N_s channels as long as there are at least N' unique error patterns. Furthermore, we *do not* need to know which channels give the unique error patterns. Hence, we may use a probabilistic approximation which states that if we have at least N_s channels, then we are likely to have N' unique error patterns. We discuss these probabilities in greater detail in Subsection II-F and see that our models can be confidently deployed for a wide range of parameters. For example, when $n = N = 100$ and $t = 3$, we can expect roughly 100 (more precisely, 99.97...) unique error patterns for deletion errors by Equation (5).

Our different versions of the reconstruction model seem especially relevant for DNA-memories in which we may obtain multiple (erroneous) copies of stored information. The traditional Levenshtein's model, which considers only unique

copies, has its limitations as it is possible to obtain same output words from multiple channels. In our multiset model, the output words can be equal, but the applied error patterns are required to be different. Although the situation can be viewed as somewhat idealized, we show in Subsection II-F that the model seems to be reasonable for a wide range of parameters, when each error pattern is assumed to occur with equal probability. Our second model, the non-multiset one, allows some of the channels to have the same error pattern as long as there are enough different patterns. The probability of having different patterns can be increased by obtaining more output words from channels.

D. Comparing the three models

To summarize the three channel models, in the **traditional channel model** each channel gives a unique output word and we have $|Y| = N$. In the **multiset error pattern channel model**, a unique set of errors occurs in each channel and we have $|Y_m| = N$, where Y_m is a multiset of output words. In the **non-multiset error pattern channel model**, a unique set of errors occurs in each channel but we consider a non-multiset Y of output words. In particular, we may have $|Y| < N$. Notice that although in each of these cases we use the term '*channel*', the meaning of the term channel is different between different models.

In the following example, we compare the three channels models.

Example 8. Consider a situation with $\mathbf{x} = 11101$ and $\mathbf{x}' = 11011$ when exactly $t = 2$ deletion errors occur in a channel. The intersection of output words which can be obtained from both $\mathbf{x}$ and $\mathbf{x}'$ is $Y' = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\} = \{111, 110, 101\}$. We have presented these output words together with all the deletion vectors that may lead to them in Table I.

When we consider these words in the traditional Levenshtein's model, we notice that if $\mathbf{x}$ is transmitted, then we can never distinguish between $\mathbf{x}$ and $\mathbf{x}'$ since every output word that can be obtained from $\mathbf{x}$ can also be obtained from $\mathbf{x}'$. However, with the non-multiset and multiset error pattern models we can distinguish between these two words. The non-multiset model requires (in the worst case) 10 channels since we may obtain the output words in Y' with nine deletion vectors from $\mathbf{x}'$ while Y' can be obtained from $\mathbf{x}$ with 10 deletion vectors and thus, we require $9 + 1 = 10$ channels to distinguish between $\mathbf{x}$ and $\mathbf{x}'$. Furthermore, the multiset model requires (in the worst case) nine channels to distinguish between $\mathbf{x}$ and $\mathbf{x}'$ since we may obtain $\mathbf{y}_1$ with four deletion vectors, $\mathbf{y}_2$ with one deletion vector and $\mathbf{y}_3$ with three deletion vectors from both $\mathbf{x}$ and $\mathbf{x}'$, that is, in total with eight deletion vectors from both words.

Notice that in the multiset model we have more information available for us than in the non-multiset model. Hence, it is to be expected that the multiset model requires less channels than the non-multiset model.

We are especially interested in how many channels N are required to unambiguously decide the transmitted word $\mathbf{x}$ in the different models. In what follows, we give an exact formulation for this value in the case of deletion vectors. Let us denote by $D = \{\mathbf{d}_1, \ldots, \mathbf{d}_{|D|}\}$ and by $D' = \{\mathbf{d}'_1, \ldots, \mathbf{d}'_{|D'|}\}$ a set of deletion vectors we apply to $\mathbf{x}$ or $\mathbf{x}'$, respectively. Moreover, let $Y(\mathbf{x}, D)$ and $Y(\mathbf{x}', D')$ be the sets of output words and $Y_m(\mathbf{x}, D)$ and $Y_m(\mathbf{x}', D')$ be the multisets of output words we obtain when we apply deletion vector sets D and D' to $\mathbf{x}$ and $\mathbf{x}'$, respectively.

TABLE I
THE THREE OUTPUT WORDS $\mathbf{y}_i$ WHICH WE CAN OBTAIN FROM BOTH $\mathbf{x}$ AND $\mathbf{x}'$ WHEN EXACTLY TWO DELETION ERRORS OCCUR IN EACH CHANNEL TOGETHER WITH THE DELETION VECTORS THAT MAY LEAD TO THEM.

	$\mathbf{x} = 11101$		$\mathbf{x}' = 11011$	
$\mathbf{y}_1 = 111$	10010	01010	10100	01100
	00110	00011	00110	00101
$\mathbf{y}_2 = 110$	10001	01001	00011	
	00101			
$\mathbf{y}_3 = 101$	11000	10100	10010	10001
	01100		01010	01001

In the following, we give for each of the three channel models a formal definition for the maximum number of channels such that some words $\mathbf{x}$ and $\mathbf{x}'$ cannot be distinguished for suitable choices of D and D'. In the multiset error pattern model, when at most t deletion errors occur in each channel, that is, each deletion vector of D and D' is at most of weight t, such maximum number of channels N_m is equal to

$$\max_{\substack{\mathbf{x}\neq\mathbf{x}' \\ \mathbf{x},\mathbf{x}'\in C\subseteq\mathbb{Z}_q^n}} \max_{D,D'}\{|D| \mid Y_m(\mathbf{x}, D) = Y_m(\mathbf{x}', D')\}. \tag{1}$$

In the non-multiset error pattern model, when at most t deletion errors occur in each channel, the maximum number of channels N for which we might not be able to unambiguously decide between some codewords $\mathbf{x}$ and $\mathbf{x}'$ is equal to

$$\max_{\substack{\mathbf{x}\neq\mathbf{x}' \\ \mathbf{x},\mathbf{x}'\in C\subseteq\mathbb{Z}_q^n}} \max_{D,D'}\{|D| \mid Y(\mathbf{x}, D) = Y(\mathbf{x}', D'), \text{ and } |Y_m(\mathbf{x}, D)| = |Y_m(\mathbf{x}', D')|\}. \tag{2}$$

Note that in the previous equation, we could have replaced $|Y_m(\mathbf{x}, D)| = |Y_m(\mathbf{x}', D')|$ by $|D| = |D'|$. In the traditional model, the maximum number of channels N for which we might not be able to unambiguously decide between some two codewords when at most t deletions may occur in any channel can be expressed in a comparable way to Equations (1) and (2) as

$$\max_{\substack{\mathbf{x}\neq\mathbf{x}' \\ \mathbf{x},\mathbf{x}'\in C\subseteq\mathbb{Z}_q^n}} \max_{D,D'}\{|Y(\mathbf{x}, D)| \mid Y(\mathbf{x}, D) = Y(\mathbf{x}', D')\}. \tag{3}$$

As it will be seen in Section III, we note that when we increase the number of channels by 1 in Equations (1) and (2), we obtain the number of channels required to unambiguously determine the transmitted word. Note that in the case of Levenshtein's traditional channel model, this is not always possible in the case of deletion errors.

In Section III, we will consider *extremal word pairs* for deletion errors, that is, distinguishable word pairs which require the largest number of channels for distinguishing them.

This word pair $\mathbf{x}, \mathbf{x}'$ corresponds to the word pair attaining the maximum number of channels in Equations (1), (2) and (3). In particular, as long as $n \geq 2t+2$ (see Theorem 9), there *always* exists (unlike in the traditional model) a number of channels which is enough for distinguishing any word pairs in non-multiset and multiset error pattern models. As we will see in Section III, the set of extremal word pairs requiring the largest number of channels to distinguish them, differs in the case of deletion errors in the three models: the traditional Levenshtein's model, the multiset model and the non-multiset model (see Remark 19). From the viewpoint of worst-case analysis, this seems interesting. To separate between error patterns with and without multisets, we will use notations N_m for the number of channels and $\mathcal{L}_m$ for the list size when we consider the multiset case.

Later, in Section IV, we consider the situation where three types of errors substitutions, insertions and deletions can occur in the same channel from a probabilistic point of view. In our probabilistic perspective, we consider the probability for errors (rather than for output words, as one could expect in the case of traditional Levenshtein's model). As we have seen in Example 6, error pattern models and traditional Levenshtein's model differ also from the probabilistic perspective.

E. Substitutions and insertions

Similarly to deletion errors, we could also introduce substitution vectors for *substitution* errors. However, unlike in the case of deletion errors, two distinct substitution vectors would lead to distinct output words. In particular, if we know the transmitted word and the output word as well as that only substitution errors have occurred, then we can exactly deduce which substitution errors have occurred. Hence, in the case of substitution errors it does not matter if we assume that every channel gives a unique output or if a unique set of substitution errors occur in a channel since both approaches lead to the same conclusion.

With an *insertion vector* we mean an ordered set of $n+1$ (q-ary) words of total length at most t. We denote the empty word by ε. When an insertion error occurs, we insert, for each $1 \leq i \leq n+1$, the ith word of the insertion vector after the $(i-1)$th symbol of the word $\mathbf{x}$. Note that for $i=1$ by saying *after the* 0*th symbol* we mean before the first symbol. Insertion errors are not as problematic when we assume that each channel outputs a different word. In fact, the size of an error ball of radius t does not depend on the central word [19]. However, we still have some problems with the probabilities. Consider, for example, $\mathbf{x} = 000 \in \mathbb{Z}_2^3$ and exactly $t=1$ insertions. If we now assume (as in the traditional model) that each output word is unique, then $Y = \{0000, 1000, 0100, 0010, 0001\}$. However, if we assume that each channel has a unique error pattern, then $Y = \{0000, 0000, 0000, 0000, 1000, 0100, 0010, 0001\}$. As we can see, the probability that we output the word 0000 is 20% in the first case and 50% in the second case. Moreover, it seems natural that 0000 is more likely than the other words to be outputted since there are 4 different ways to obtain it while the other words have only 1. Let us denote by $B_t^I(\mathbf{x})$ the insertion ball for insertion vectors of radius t centered at a word $\mathbf{x} \in \mathbb{Z}_q^n$. We have

$$|B_t^I(\mathbf{x})| = \sum_{i=0}^{t} q^i \binom{n+i}{i}. \tag{4}$$

Indeed, let us consider the number of words which we can obtain from $\mathbf{x}$ with exactly j insertions such that $0 \leq j \leq t$. Insertion vector consists of $n+1$ (possibly empty) words with total length of j. The answer to the question asking how many combinations there are for possible locations of inserted symbols is given by a classical combinatorial technique of *stars and bars* [27]. Indeed, this problem can be considered as having $n+1$ boxes and j balls where the balls are inserted to the boxes. Thus, the technique of stars and bars tells that there are $\binom{n+j}{j}$ ways to insert balls into these boxes. Moreover, there are q^j ways to choose the inserted symbols once their locations are known. Notice that the cardinality in (4) differs from the cardinality of an insertion ball considered in [19].

F. Probabilities

In this section, we briefly consider the probabilities on how many unique error patterns we might obtain when each error pattern is equally probable. Consider a setup in which we have a collection of m distinct coupons and each time we draw one coupon, it is replaced with a new one. In a well-known *Coupon Collector Problem* [28], we are asked how many coupons we need to buy randomly to collect at least one copy of every coupon. Furthermore, in the *Partial Coupon Collector Problem* PCCP(j, m), we are asked how many coupons we need to buy randomly to obtain j different coupons from m total coupon types. This setup corresponds to our problem with distinct error patterns in channels assuming that all error patterns are equally likely. Asking: "Through how many channels do we need to transmit word $\mathbf{x}$ to obtain j distinct error patterns" is the same as PCCP(j, m). In [28], the expected value for PCCP(j, m) has been presented as:

$$E[PCCP(j,m)] = m(H_m - H_{m-j}) \approx m \ln \frac{m}{m-j},$$

where H_m is the mth harmonic number $\sum_{i=1}^{m} \frac{1}{i}$. The approximation follows from $\gamma + 1/(2m+2) + \ln m < H_m < \gamma + 1/(2m) + \ln m$, where γ is the Euler-Mascheroni constant, see [29]. When we consider deletion vectors and at most t_d deletions occur in any channel, the value m is $V_2(n, t_d)$. For insertion errors with at most t_i insertions in a channel, the value m is $|B_{t_i}^I(\mathbf{x})| = \sum_{j=0}^{t_i} q^j \binom{n+j}{j}$ by Equation (4).

Another way to consider this problem is: If we transmit the word $\mathbf{x} \in \mathbb{Z}_q^n$ through N channels and 1 of m equally likely error patterns may occur in any of them, what is the expected number of unique error patterns occurring in these channels? Observe that the likelihood of any single error pattern not occurring in any of the N channels is $(\frac{m-1}{m})^N$. Thus, a particular error pattern occurs in at least one of the channels with probability $1-(\frac{m-1}{m})^N$ and hence, the expected number of unique error patterns is

$$m\left(1 - \left(\frac{m-1}{m}\right)^N\right). \tag{5}$$

III. Deletion vectors

In this section, we consider how many channels we may require to ensure that we can uniquely determine the transmitted word, that is $\mathcal{L} = 1$, when we have deletion vectors of weight at most (or exactly) t. This number of channels can be obtained by adding 1 to the value of Equations (1), (2) and (3) (depending on the considered model). We give two types of results. First we consider the non-multiset error pattern model and give the exact minimum number of different error patterns, that is, the minimum value for N which guarantees different *sets* of output words from two different transmitted words, that is, cases where $\mathcal{L} = 1$. Then, we consider the same problem for the *multiset* model. In the following theorem, we provide a number of channels, which guarantees that $\mathcal{L} = 1$ in the non-multiset error pattern case. However, observe that Theorem 9 also holds for the multiset model and gives $\mathcal{L}_m = 1$, since if the sets of output words are different, then also multisets of the output words are different. In this section, we restrict our considerations to $C = \mathbb{Z}_q^n$. In the following theorem we concentrate on the case with $q = 2$. However, the same result holds also for larger q (see the discussion after Theorem 9).

Theorem 9. *Let t, n and q be integers such that $t \geq 1$, $n \geq 2t + 2 \geq 4$ and $q = 2$.*

(i) If at most t deletion errors occur in a channel and the number N of channels satisfies $N \geq V_2(n,t) - V_2(\lceil n/2 \rceil - 1, t) + 1$, then the output word (multi)set Y is unique for any transmitted word and $\mathcal{L} = 1$.

(ii) If exactly t deletion errors occur in a channel and the number N of channels satisfies $N \geq \binom{n}{t} - \binom{\lceil n/2 \rceil - 1}{t} + 1$, then the output word (multi)set Y is unique for any transmitted word and $\mathcal{L} = 1$.

Proof. In this proof, we only consider Case (i). However, the proof for (ii) follows by replacing in the following proof each $V_2(a,b)$ by $\binom{a}{b}$ and by changing every weight constraint $w(\mathbf{d}) \leq t$ to $w(\mathbf{d}) = t$ for a deletion vector $\mathbf{d}$.

Let $N \geq V_2(n,t) - V_2(\lceil n/2 \rceil - 1, t) + 1$. Note that we may apply $V_2(n,t) - V_2(\lceil n/2 \rceil - 1, t) + 1$ unique deletion vectors on any word of length n as the number of all such vectors is equal to $V_2(n,t)$. Suppose on the contrary that there exists a set of output words Y which can be obtained by applying a set D of N deletion vectors to $\mathbf{x}$ and also by applying a set D' of N deletion vectors to $\mathbf{x}'$. We first show that both $\mathbf{x}$ and $\mathbf{x}'$ have the same weight.

Claim 1: We have $w(\mathbf{x}) = w(\mathbf{x}')$.

Proof of Claim 1. Let us suppose on the contrary, without loss of generality, that $w(\mathbf{x}) > w(\mathbf{x}')$. Moreover, let us first assume that $w(\mathbf{x}) > n/2$. We denote $m = w(\mathbf{x}) - w(\mathbf{x}')$.

Let us denote by D'' a set of deletion vectors $\mathbf{d}''$ with $w(\mathbf{d}'') \leq t$, where we can have $d''_i = 1$ if $x'_i = 1$ or for at most $m - 1$ indices i for which $x'_i = 0$ (in other words, deletion vectors delete some 1's and at most $m - 1$ symbols 0 from $\mathbf{x}'$). In other words,

$$D'' = \{\mathbf{d}'' \in \mathbb{Z}_2^n \mid w(\mathbf{d}'') \leq t, \text{ and } |\text{supp}(\mathbf{d}'') \setminus \text{supp}(\mathbf{x}')| \leq m - 1\}.$$

Then, we have $|D''| \geq V_2(m - 1 + w(\mathbf{x}'), t) = V_2(w(\mathbf{x}) - 1, t) \geq V_2(\lfloor n/2 \rfloor, t)$. Furthermore, we can observe that if we obtain $\mathbf{y}'$ from $\mathbf{x}'$ with a deletion vector in D'', then $\mathbf{y}'$ has at least $n - w(\mathbf{x}') - (m - 1) = n - w(\mathbf{x}) + 1$ symbols 0 and we cannot obtain $\mathbf{y}'$ from $\mathbf{x}$ with any deletion vector since $\mathbf{x}$ has $n - w(\mathbf{x})$ symbols 0. Thus,

$$\begin{aligned} |D'| &\leq V_2(n,t) - |D''| \leq V_2(n,t) - V_2(\lfloor n/2 \rfloor, t) \\ &< V_2(n,t) - V_2(\lceil n/2 \rceil - 1, t) + 1 \leq N, \end{aligned} \tag{6}$$

a contradiction.

Moreover, the case $w(\mathbf{x}) \leq n/2$ is similar. Indeed, in this case we may swap 1's and 0's in $\mathbf{x}$ and $\mathbf{x}'$. Now $w(\mathbf{x}') > w(\mathbf{x})$ and $w(\mathbf{x}') > n/2$. After this, we could apply above proof by swapping $\mathbf{x}$ and $\mathbf{x}'$, by switching D' to D and m to $m' = w(\mathbf{x}') - w(\mathbf{x})$. Thus, Claim 1 follows.

Let us now assume that h is the smallest index for which $x_h \neq x'_h$. Furthermore, without loss of generality, assume that $x_h = 0$ and $x'_h = 1$. Moreover, let us notate $w_1(\mathbf{w}) = |\{i \mid i \leq h, i \in \text{supp}(\mathbf{w})\}|$ and $w_2(\mathbf{w}) = |\{i \mid i > h, i \in \text{supp}(\mathbf{w})\}|$. We have $w_1(\mathbf{x}) = w_1(\mathbf{x}') - 1$ and $w_2(\mathbf{x}) = w_2(\mathbf{x}') + 1$, since $w(\mathbf{x}) = w(\mathbf{x}')$. There are $w_1(\mathbf{x})$ symbols 1 and $h - w_1(\mathbf{x})$ symbols 0 in $\mathbf{x}$ before the $(h+1)$th coordinate. Moreover, in $\mathbf{x}'$, before the $(h+1)$th coordinate, there are $w_1(\mathbf{x}) + 1$ symbols 1 and $h - w_1(\mathbf{x}) - 1$ symbols 0.

Let us consider the following deletion vector sets $D_1 \subseteq \mathbb{Z}_2^n$ and $D_2 \subseteq \mathbb{Z}_2^n$:

$$\begin{aligned} D_1 = \{&\mathbf{d} \in \mathbb{Z}_2^n \mid w(\mathbf{d}) \leq t, \text{supp}(\mathbf{d}) \cap [1,h] \subseteq \text{supp}(\mathbf{x}), \\ &\text{and } (\text{supp}(\mathbf{d}) \cap [h+1,n]) \cap \text{supp}(\mathbf{x}) = \emptyset\} \text{ and} \\ D_2 = \{&\mathbf{d} \in \mathbb{Z}_2^n \mid w(\mathbf{d}) \leq t, \text{supp}(\mathbf{d}) \cap [h+1,n] \subseteq \text{supp}(\mathbf{x}'), \\ &\text{and } (\text{supp}(\mathbf{d}) \cap [1,h]) \cap \text{supp}(\mathbf{x}') = \emptyset\}. \end{aligned}$$

In other words, we have $\mathbf{d} \in D_1$ if and only if $w(\mathbf{d}) \leq t$ and its support is such that for $d_i = 1$ we require that $x_i = 1$ and $i \leq h$, or $x_i = 0$ and $i > h$. Similarly we have $\mathbf{d}' \in D_2$ if and only if $w(\mathbf{d}') \leq t$ and its support is such that for $d'_i = 1$ we require that $x'_i = 0$ and $i \leq h$, or $x'_i = 1$ and $i > h$.

Claim 2: If we obtain $\mathbf{y}$ from $\mathbf{x}$ with a deletion vector $\mathbf{d} \in D_1$, then $\mathbf{y}$ cannot be obtained with any deletion vector from $\mathbf{x}'$.

Proof of Claim 2. Let us assume that $\mathbf{y}$ is obtained from $\mathbf{x}$ with a deletion vector $\mathbf{d} \in D_1$. Suppose on the contrary that we can obtain $\mathbf{y}$ from $\mathbf{x}'$ with some deletion vector $\mathbf{d}'$ of weight at most t. We have $w(\mathbf{d}) = w(\mathbf{d}')$ as the original words $\mathbf{x}$ and $\mathbf{x}'$ are of equal length.

Let us first consider how $\mathbf{y}$ can be obtained from $\mathbf{x}$ with $\mathbf{d}$. Consider the symbol 0 in the hth coordinate (note that it is not deleted by $\mathbf{d}$). In $\mathbf{x}$, there are $h - w_1(\mathbf{x}) - 1$ symbols 0 before it and $w_1(\mathbf{x})$ symbols 1. Notice that $\mathbf{d}$ deletes symbols 0 from $\mathbf{x}$ only after the symbol 0 in the coordinate h. Hence, when we consider the symbol 0 in $\mathbf{y}$ which has $h - w_1(\mathbf{x}) - 1$ symbols 0 before it, we notice that it has exactly $w_1(\mathbf{x}) - w_1(\mathbf{d})$ symbols 1 before it.

Let us then consider how $\mathbf{y}$ can be obtained from $\mathbf{x}'$ with $\mathbf{d}' \in \mathbb{Z}_2^n$. Consider the first symbol 0 after the hth coordinate in $\mathbf{x}'$. This symbol exists because $w(\mathbf{x}) = w(\mathbf{x}')$. In $\mathbf{x}'$, there are $h - w_1(\mathbf{x}) - 1$ symbols 0 before it and at least $w_1(\mathbf{x}) + 1$ symbols 1. Since $w(\mathbf{x}) = w(\mathbf{x}')$ and $\mathbf{d}$ deletes exactly $w_1(\mathbf{d})$

symbols 1, we can delete at most $w_1(\mathbf{d})$ symbols 1 from $\mathbf{x}'$ with $\mathbf{d}'$. Thus, there are at least $w_1(\mathbf{x})+1-w_1(\mathbf{d})$ symbols 1 before the symbol 0 which has $h-w_1(\mathbf{x})-1$ symbols 0 before it in $\mathbf{y}$ obtained from $\mathbf{x}'$. Notice that this kind of symbol 0 must exist also in $\mathbf{y}$ when we obtain it from $\mathbf{x}'$ since $\mathbf{y}$ has at least $h-w_1(\mathbf{x})$ symbols 0 when we obtain it from $\mathbf{x}$. Thus, word $\mathbf{y}$, which we obtained from $\mathbf{x}$, is not identical with word $\mathbf{y}$ which we obtained from $\mathbf{x}'$, a contradiction which proves Claim 2.

Since D_1 and D_2 are constructed in symmetrical ways, Claim 2 also holds for D_2, that is, if we obtain $\mathbf{y}$ from $\mathbf{x}'$ with $\mathbf{d}' \in D_2$, then $\mathbf{y}$ cannot be obtained with any deletion vector from $\mathbf{x}$. Indeed, to prove this, we consider the $w_1(\mathbf{x}')-1$ symbols 1 in $\mathbf{x}'$ before the hth coordinate and then we construct $\mathbf{y}$ which has $h-w_1(\mathbf{x}')-w_1(\mathbf{d}')$ symbols 0 before the $w_1(\mathbf{x}')$th symbol 1. When we try to obtain this $\mathbf{y}$ from $\mathbf{x}$, we notice that there are always at least $h-w_1(\mathbf{x}')-w_1(\mathbf{d}')+1$ symbols 0 before the $w_1(\mathbf{x}')$th symbol 1 (if it exists) since $\mathbf{d}$ can delete at most $w_1(\mathbf{d}')$ symbols 0 from $\mathbf{x}$ (recall that $w(\mathbf{x})=w(\mathbf{x}')$) and there are, in $\mathbf{x}$, $h-w_1(\mathbf{x}')+1$ symbols 0 before the $w_1(\mathbf{x}')$th symbol 1.

Claim 3: We have $|D_i| \geq V_2(\lceil n/2 \rceil -1,t)$ for $i=1$ or $i=2$.

Proof of Claim 3. There are $w_1(\mathbf{x})$ symbols 1 and $n-h-w_2(\mathbf{x})$ symbols 0 which we can remove with a deletion vector $\mathbf{d} \in D_1$. Thus, $|D_1| = V_2(n-h-w_2(\mathbf{x})+w_1(\mathbf{x}),t)$. Similarly, there are $h-w_1(\mathbf{x}')=h-w_1(\mathbf{x})-1$ symbols 0 and $w_2(\mathbf{x}')=w_2(\mathbf{x})-1$ symbols 1 which we can remove with deletion vector $\mathbf{d}' \in D_2$. Thus, $|D_2| = V_2(h-w_1(\mathbf{x})-1+w_2(\mathbf{x})-1,t)$.

We split the proof between Cases A) $|D_1| \geq |D_2|$ and B) $|D_2| \geq |D_1|$. Consider first Case A). Hence, $n-h-w_2(\mathbf{x})+w_1(\mathbf{x}) \geq h-w_1(\mathbf{x})-1+w_2(\mathbf{x})-1$. Notice that $n-h-w_2(\mathbf{x})+w_1(\mathbf{x})+(h-w_1(\mathbf{x})-1+w_2(\mathbf{x})-1)=n-2$. Since $|D_1| \geq |D_2|$, we have $n-h-w_2(\mathbf{x})+w_1(\mathbf{x}) \geq \lceil \frac{n-2}{2} \rceil$. Thus,

$$|D_1| = V_2(n-h-w_2(\mathbf{x})+w_1(\mathbf{x}),t) \geq V_2(\lceil n/2 \rceil -1,t)$$

as claimed.

Case B) is similar. We have $n-h-w_2(\mathbf{x})+w_1(\mathbf{x}) \leq h-w_1(\mathbf{x})-1+w_2(\mathbf{x})-1$ and $n-h-w_2(\mathbf{x})+w_1(\mathbf{x})+(h-w_1(\mathbf{x})-1+w_2(\mathbf{x})-1)=n-2$. Since $|D_2| \geq |D_1|$, we have $h-w_1(\mathbf{x})-1+w_2(\mathbf{x})-1 \geq \lceil \frac{n-2}{2} \rceil$. Thus,

$$|D_2| = V_2(h-w_1(\mathbf{x})-1+w_2(\mathbf{x})-1,t) \geq V_2(\lceil n/2 \rceil -1,t)$$

as claimed. Now, Claim 3 follows.

Let $\{v,w\}=\{1,2\}$ and $|D_v| \geq |D_w|$. If $v=1$, then $\mathbf{x}_v=\mathbf{x}$ and $\mathbf{x}_w=\mathbf{x}'$. If $v=2$, then $\mathbf{x}_w=\mathbf{x}$ and $\mathbf{x}_v=\mathbf{x}'$. We have $|D_v| \geq V_2(\lceil n/2 \rceil -1,t)$ by Claim 3 and by Claim 2 we cannot obtain the output words $\mathbf{y}$, which are obtained from $\mathbf{x}_v$ with $\mathbf{d} \in D_v$, with any deletion vector from $\mathbf{x}_w$. Thus, we have $N=|D| \leq V_2(n,t)-|D_v| \leq V_2(n,t)-V_2(\lceil n/2 \rceil -1,t)$. Therefore, $\mathcal{L}=1$ when $N \geq V_2(n,t)-V_2(\lceil n/2 \rceil -1,t)+1$. □

Observe that although we consider the binary case in the previous theorem, it is easy to see that the result applies also for larger alphabets. Indeed, consider distinct words $\mathbf{x},\mathbf{x}' \in \mathbb{Z}_q^n$ for $q>2$, and let $i \in [1,n]$ be such that $x_i \neq x'_i$. We partition $\mathbb{Z}_q$ into nonempty sets A and B such that $x_i \in A$ and $x'_i \in B$, and let $f:\mathbb{Z}_q^n \to \mathbb{Z}_2^n$ transform any q-ary word to binary word by changing all symbols from A to 0's and symbols from B to 1's. For example, we may choose $A=\{x_i\}$ and $B=[0,q-1]\setminus\{x_i\}$. Note that $f(\mathbf{x}) \neq f(\mathbf{x}')$. Now, we may observe that if a deletion vector $\mathbf{d}$ turns $\mathbf{x}$ and $\mathbf{x}'$ into the same word $\mathbf{y}$, then $\mathbf{d}$ transforms both $f(\mathbf{x})$ and $f(\mathbf{x}')$ into $f(\mathbf{y})$. Hence, the lower bound of Theorem 9 applies also for larger alphabets.

In the subsequent theorem, we see that the lower bounds of Theorem 9 are tight (also for larger alphabets) in the non-multiset error pattern model. This is done by showing that a suitably chosen pair $\mathbf{x},\mathbf{x}'$ form an extremal word pair for the non-multiset model.

Theorem 10. *Let $n \geq 2t+2 \geq 4$ and $C=\mathbb{Z}_q^n$. Consider words $\mathbf{x}=0^{\lceil n/2 \rceil -1}10^{\lfloor n/2 \rfloor}$ and $\mathbf{x}'=0^{\lceil n/2 \rceil}10^{\lfloor n/2 \rfloor -1}$.*

(i) If at most t deletion errors occur in a channel and the number N of channels satisfies $N \leq V_2(n,t)-V_2(\lceil n/2 \rceil -1,t)$, then we can obtain the same output word set Y from N channels with input words $\mathbf{x}$ and $\mathbf{x}'$.

(ii) If exactly t deletion errors occur in a channel and the number N of channels satisfies $N \leq \binom{n}{t}-\binom{\lceil n/2 \rceil -1}{t}$, then we can obtain the same output word set Y from N channels with input words $\mathbf{x}$ and $\mathbf{x}'$.

Proof. Again, in this proof, we only consider Case (i) and Case (ii) can be shown by replacing in the following proof each $V_2(a,b)$ by $\binom{a}{b}$ and by changing every weight constraint $w(\mathbf{d}) \leq t$ to $w(\mathbf{d})=t$ for a deletion vector $\mathbf{d}$.

Let us transmit a word $\mathbf{x} \in \mathbb{Z}_q^n$ with $\mathrm{supp}(\mathbf{x})=\{\lceil n/2 \rceil\}$ and $x_{\lceil n/2 \rceil}=1$ through N channels. It is enough to consider $N=V_2(n,t)-V_2(\lceil n/2 \rceil -1,t)$ channels. Let $\mathbf{x}' \in \mathbb{Z}_q^n$ be a word such that $\mathrm{supp}(\mathbf{x}')=\{\lceil n/2 \rceil +1\}$ and $x'_{\lceil n/2 \rceil +1}=1$. Let us consider following sets of deletion vectors:

$$D=\{\mathbf{d} \mid w(\mathbf{d}) \leq t, \mathrm{supp}(\mathbf{d}) \cap [\lceil n/2 \rceil, n] \neq \emptyset\}$$

and (notice that $\lceil n/2 \rceil - \lfloor n/2 \rfloor$ is 0 or 1 depending on the parity of n)

$$D'=\{\mathbf{d} \mid w(\mathbf{d}) \leq t, \mathrm{supp}(\mathbf{d}) \cap [1+(\lceil n/2 \rceil - \lfloor n/2 \rfloor), \text{ and } \lceil n/2 \rceil +1] \neq \emptyset\}.$$

Observe that $|D|=|D'|=V_2(n,t)-V_2(\lceil n/2 \rceil -1,t)$. Indeed, let us consider first the set D. There are $V_2(n,t)$ different vectors of weight at most t. Moreover, those vectors belong to D unless their support is within the set $[1,\lceil n/2 \rceil -1]$ and there are $V_2(\lceil n/2 \rceil -1,t)$ such vectors. Similarly, it can be shown that $|D'|=V_2(n,t)-V_2(\lceil n/2 \rceil -1,t)$.

Let us first consider the set of output words Y which we obtain from $\mathbf{x}$ with D. We have

$$Y=\{\mathbf{y} \in \{0,1\}^h \mid n-t \leq h \leq n-1, w(\mathbf{y}) \leq 1, \text{ and } \mathrm{supp}(\mathbf{y}) \subseteq [h+1-\lfloor n/2 \rfloor, \lceil n/2 \rceil]\}.$$

Indeed, we delete at least 1 and at most t symbols and hence, $\mathbf{y} \in \{0,1\}^h$ for $n-t \leq h \leq n-1$. Moreover, $w(\mathbf{y}) \leq 1$, since it is possible that we delete the only symbol 1 in $\mathbf{x}$. Finally, if $w(\mathbf{y})=1$, then we deleted at least 1 symbol 0 after the symbol 1 in $\mathbf{x}$ and hence, $\mathrm{supp}(\mathbf{y}) \subseteq [h-(\lfloor n/2 \rfloor -1), \lceil n/2 \rceil]$. In other

words, there are at most $\lfloor n/2 \rfloor - 1$ symbols 0 after the symbol 1.

Let us then consider the set of output words Y' which we obtain from $\mathbf{x}$ with D'. Similarly to previous case, it is easy to check that $Y' = Y$. Thus, the claim follows. □

Let us next consider deletion vectors in the *multiset* model. Now we use multisets of output words to distinguish between two possible transmitted words. Recall that the lower bounds of Theorem 9 hold also for the multiset model. Theorem 9(i) is tight in the case of multisets when $t = 1$.

Proposition 11. *Let $t = 1$ and $q \geq 2$. If $N_m \leq \lceil n/2 \rceil + 1 = V_2(n, 1) - V_2(\lceil n/2 \rceil - 1, 1)$, then $\mathcal{L}_m \geq 2$.*

Proof. Let us transmit the words $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_q^n$ with $\text{supp}(\mathbf{x}) = \{\lceil n/2 \rceil\}$, $x_{\lceil n/2 \rceil} = 1$ and $\text{supp}(\mathbf{x}') = \{\lceil n/2 \rceil + 1\}$, $x'_{\lceil n/2 \rceil + 1} = 1$ through $N_m = V_2(n, t) - V_2(\lceil n/2 \rceil - 1, t) = \lfloor n/2 \rfloor + 1$ channels. The claim follows by applying the sets of deletion vectors $D = \{\mathbf{d}_i \mid \text{supp}(\mathbf{d}_i) = \{n + 1 - i\}, i \in [1, \lfloor n/2 \rfloor + 1]\}$ and $D' = \{\mathbf{d}'_i \mid \text{supp}(\mathbf{d}'_i) = \{i + \lceil n/2 \rceil - \lfloor n/2 \rfloor, i \in [1, \lfloor n/2 \rfloor + 1]\}\}$ to $\mathbf{x}$ and $\mathbf{x}'$, respectively. Indeed, we get the same multiset Y_m (consisting of $\lfloor n/2 \rfloor$ times the word $0^{\lceil n/2 \rceil - 1}10^{\lfloor n/2 \rfloor - 1}$ and once the word 0^{n-1}) in both cases. □

Next we provide further results for the multiset model and compare it with the non-multiset one. We have decided to focus on even n in the upcoming considerations to avoid extra complications, as the behaviour of the odd n seems to be somewhat different. Note that, in the following, we sometimes concentrate on the case in which exactly t errors occur in channels instead of a case in which at most t errors occur.

As we can see in the following proposition, corollaries and especially in Remark 16, the construction which gives a tight bound for the case with $t = 1$ *does not* give a tight bound for $t = 2$ for the multiset model.

Proposition 12. *Let $t \geq 1$, $q \geq 2$ and $n \geq t + 1$. Let us assume that exactly t deletion errors occur in a channel in the multiset model. We can distinguish between $\mathbf{x} = 0^{a-1}10^{n-a}$ and $\mathbf{x}' = 0^a 10^{n-a-1}$ with $N = \binom{n}{t} - \binom{a-1}{\lfloor \frac{at+a}{n} \rfloor}\binom{n-a-1}{t-\lfloor \frac{at+a}{n} \rfloor} + 1$ channels while $N - 1$ is not enough.*

Proof. Let us transmit the words $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_q^n$, for which $\text{supp}(\mathbf{x}) = \{a\}$, $\text{supp}(\mathbf{x}') = \{a + 1\}$ and $x_a = x'_{a+1} = 1$, through $N_m = \binom{n}{t} - \binom{a-1}{\lfloor \frac{at+a}{n} \rfloor}\binom{n-a-1}{t-\lfloor \frac{at+a}{n} \rfloor}$ channels.

First of all, we notice that we can obtain the word $\mathbf{0} \in \mathbb{Z}_q^{n-t}$ from both $\mathbf{x}$ and $\mathbf{x}'$ with $\binom{n-1}{t-1}$ deletion vectors each deleting the single 1. Furthermore, we can obtain a word $\mathbf{y}_1$ with $\text{supp}(\mathbf{y}_1) = \{a - i\}$ from word $\mathbf{x}$ with $\binom{a-1}{i}\binom{n-a}{t-i}$ deletion vectors for $i \in [0, t-1]$ and from $\mathbf{x}'$ with $\binom{a}{i+1}\binom{n-a}{t-i-1}$ deletion vectors for $i \in [0, t-1]$. Note that the upper bound $t - 1$ for i is due to the fact that we cannot obtain the word $\mathbf{y}_1$ with $\text{supp}\{a - t\}$ from $\mathbf{x}'$. Together, these mean that we may obtain the same multiset of output words from

$$\binom{n-1}{t-1} + \sum_{i=0}^{t-1} \min\left\{\binom{a}{i+1}\binom{n-a-1}{t-i-1}, \binom{a-1}{i}\binom{n-a}{t-i}\right\}$$

channels.

Let us consider $\binom{a}{i+1}\binom{n-a-1}{t-i-1} / \left(\binom{a-1}{i}\binom{n-a}{t-i}\right)$ for $i \in [0, t-1]$ to determine when one of these is smaller. We have

$$\begin{aligned} &\frac{\binom{a}{i+1}\binom{n-a-1}{t-i-1}}{\binom{a-1}{i}\binom{n-a}{t-i}} \geq 1 \\ \Leftrightarrow \quad &\frac{a(t-i)}{(i+1)(n-a)} \geq 1 \\ \Leftrightarrow \quad &\frac{at+a}{n} - 1 \geq i. \end{aligned}$$

Let us denote $A = \lfloor \frac{at+a}{n} \rfloor - 1$. In the following, we use the well-known binomial identities, of which the first is Vandermonde's identity, $\sum_{i=0}^{k}\binom{r}{i}\binom{p}{k-i} = \binom{p+r}{k}$ and $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$. Now we have

$$\begin{aligned} &\binom{n-1}{t-1} + \sum_{i=0}^{t-1} \min\left\{\binom{a}{i+1}\binom{n-a-1}{t-i-1}, \binom{a-1}{i}\binom{n-a}{t-i}\right\} \\ =&\binom{n-1}{t-1} + \sum_{i=0}^{A}\binom{a-1}{i}\binom{n-a}{t-i} + \sum_{i=A+1}^{t-1}\binom{a}{i+1}\binom{n-a-1}{t-i-1} \\ =&\binom{n-1}{t-1} + \sum_{i=0}^{A}\binom{a-1}{i}\binom{n-a-1}{t-i} + \sum_{i=0}^{A}\binom{a-1}{i}\binom{n-a-1}{t-i-1} \\ &+ \sum_{i=A+1}^{t-1}\binom{a-1}{i+1}\binom{n-a-1}{t-i-1} + \sum_{i=A+1}^{t-1}\binom{a-1}{i}\binom{n-a-1}{t-i-1} \\ =&\binom{n-1}{t-1} + \sum_{i=0}^{t-1}\binom{a-1}{i}\binom{n-a-1}{t-i-1} + \sum_{i=0}^{A}\binom{a-1}{i}\binom{n-a-1}{t-i} \\ &+ \sum_{i=A+2}^{t}\binom{a-1}{i}\binom{n-a-1}{t-i} \\ =&\binom{n-1}{t-1} + \binom{n-2}{t-1} + \sum_{i=0}^{A}\binom{a-1}{i}\binom{n-a-1}{t-i} \\ &+ \sum_{i=A+1}^{t}\binom{a-1}{i}\binom{n-a-1}{t-i} - \binom{a-1}{A+1}\binom{n-a-1}{t-A-1} \end{aligned}$$

$$
\begin{aligned}
=&\binom{n-1}{t-1}+\binom{n-2}{t-1}+\sum_{i=0}^{t}\binom{a-1}{i}\binom{n-a-1}{t-i}\\
&-\binom{a-1}{A+1}\binom{n-a-1}{t-A-1}\\
=&\binom{n-1}{t-1}+\binom{n-2}{t-1}+\binom{n-2}{t}\\
&-\binom{a-1}{A+1}\binom{n-a-1}{t-A-1}\\
=&\binom{n}{t}-\binom{a-1}{A+1}\binom{n-a-1}{t-A-1}.
\end{aligned}
$$

Hence, we require exactly $\binom{n}{t}-\binom{a-1}{A+1}\binom{n-a-1}{t-A-1}+1$ channels to distinguish between $\mathbf{x}$ and $\mathbf{x}'$ in the multiset model. □

Corollaries 13, 14 and 15 follow from Proposition 12. In these corollaries we establish how many channels are exactly required for distinguishing between some interesting word pairs $\mathbf{x}_1, \mathbf{x}_2$. These word pairs are interesting since at least for some values of n and t, they are extremal (see the discussion in Remark 19). However, it is possible that for some values of n and t, these are not extremal word pairs.

Corollary 13. *Let $n = h(2t+2)$, positive integers $h \geq 1$ and $q, t \geq 2$. If exactly t deletions occur in each channel, then in the multiset model exactly*

$$\binom{n}{t}-\binom{ht-1}{\lfloor t/2\rfloor}\binom{h(t+2)-1}{\lceil t/2\rceil}+1$$

channels are enough for distinguishing between $\mathbf{x}_1 = 0^{ht-1}10^{h(t+2)}$ and $\mathbf{x}_2 = 0^{ht}10^{h(t+2)-1}$.

Corollary 14. *Let $n \geq 2t+2$ be even, $t \geq 1$, $q \geq 2$. If exactly t deletions occur in each channel, then in the multiset model exactly*

$$\binom{n}{t}-\binom{n/2-1}{\lfloor t/2\rfloor}\binom{n/2-1}{\lceil t/2\rceil}+1$$

channels are enough for distinguishing between $\mathbf{x}_1 = 0^{n/2-1}10^{n/2}$ and $\mathbf{x}_2 = 0^{n/2}10^{n/2-1}$.

Next we consider the case of *at most* t deletion errors in a channel.

Corollary 15. *Let $n \geq 2t+2$ be even, $q, t \geq 2$. If* at most *t deletions occur in each channel, then in the multiset model for $a = n/2$ exactly*

$$V(n,t)-\sum_{i=0}^{t}\binom{n/2-1}{\lfloor \frac{i}{2}\rfloor}\binom{n/2-1}{\lceil \frac{i}{2}\rceil}+1$$

channels are enough for distinguishing between $\mathbf{x}_1 = 0^{n/2-1}10^{n/2}$ and $\mathbf{x}_2 = 0^{n/2}10^{n/2-1}$.

Proof. Let us denote by N_i, for $0 \leq i \leq t$, the number of channels we require to distinguish between $\mathbf{x}_1$ and $\mathbf{x}_2$ when exactly i errors occur. Then, we require $\sum_{i=0}^{t}(N_i-1)+1$ channels to distinguish between $\mathbf{x}_1$ and $\mathbf{x}_2$ when at most t errors occur in a channel since obtaining an output word $\mathbf{y} \in \mathbb{Z}_q^{n-i}$ from both $\mathbf{x}_1$ and $\mathbf{x}_2$ requires that exactly i deletions occur to both $\mathbf{x}_1$ and $\mathbf{x}_2$. Notice that $N_0 = 1 = \binom{n}{0}-\binom{n/2-1}{\lfloor 0/2\rfloor}\binom{n/2-1}{\lceil 0/2\rceil}+1$ and we obtain the other values for N_i from Corollary 14. Hence, we have

$$
\begin{aligned}
&\sum_{i=0}^{t}(N_i-1)+1\\
=&\sum_{i=0}^{t}\left(\binom{n}{i}-\binom{n/2-1}{\lfloor \frac{i}{2}\rfloor}\binom{n/2-1}{\lceil \frac{i}{2}\rceil}\right)+1\\
=&V(n,t)-\sum_{i=0}^{t}\binom{n/2-1}{\lfloor \frac{i}{2}\rfloor}\binom{n/2-1}{\lceil \frac{i}{2}\rceil}+1.
\end{aligned}
$$
□

In the following remark we observe some differences in the behaviour of extremal word pairs between the multiset and non-multiset models.

Remark 16. Let $t = 2$ and $n = 6h$ for an integer $h \geq 2$. Let at most t deletions occur in any channel. Then, by Corollary 15, the exact minimum number of channels required in the multiset model for distinguishing between $\mathbf{x}_1 = 0^{n/2-1}10^{n/2}$ and $\mathbf{x}_2 = 0^{n/2}10^{n/2-1}$ is $N = V(n,2)-\sum_{i=0}^{2}\binom{n/2-1}{\lfloor \frac{i}{2}\rfloor}\binom{n/2-1}{\lceil \frac{i}{2}\rceil}+1 = \binom{n}{2}+n+1-(n/2-1)-(n/2-1)^2 = n^2/4+n+1$. On the other hand, by considering Corollary 13 with $t = 2$ and Proposition 12 with $t = 1$, the exact minimum number of channels required in the multiset model for distinguishing between $\mathbf{x} = 0^{2h-1}10^{4h}$ and $\mathbf{x}' = 0^{2h}10^{4h-1}$ is

$$
\begin{aligned}
&\binom{n}{2}-(2h-1)(h(2+2)-1)\\
&+\left(n-\binom{2h-1}{\lfloor \frac{4h}{n}\rfloor}\binom{n-2h-1}{1-\lfloor \frac{4h}{n}\rfloor}\right)+1\\
=&\binom{n}{2}-(8h^2-6h+1)+(n-(n-2h-1))+1\\
=&\binom{n}{2}-2n^2/9+4n/3+1 = 5n^2/18+5n/6+1.
\end{aligned}
$$

Hence, we require $n^2/36-n/6$ more channels to distinguish between $\mathbf{x}$ and $\mathbf{x}'$ than we need for distinguishing between $\mathbf{x}_1$ and $\mathbf{x}_2$. Recall that by Theorems 9 and 10, the word pair $\mathbf{x}_1, \mathbf{x}_2$ requires the most channels for the non-multiset model. Thus, the set of extremal word pairs differs between the multiset and non-multiset models for deletion errors for some parameters of n and t.

In the following lemma, we consider how many channels we may require for distinguishing two words whose weights differ by b. Furthermore, we present a pair of words attaining the presented bound.

Lemma 17. *Let exactly t deletions occur in any channel in the multiset model. If $w(\mathbf{x}_1) = w(\mathbf{x}_2)+b$ for two binary words on symbols 0 and 1 and $b \geq 1$, then the number of channels N for which we may obtain the same output word set from both $\mathbf{x}_1$ and $\mathbf{x}_2$ is at most*

$$
\begin{aligned}
N = \sum_{i=b}^{t}\min\Bigg\{&\binom{w(\mathbf{x}_1)}{i}\binom{n-w(\mathbf{x}_1)}{t-i},\\
&\binom{w(\mathbf{x}_1)-b}{i-b}\binom{n-w(\mathbf{x}_1)+b}{t+b-i}\Bigg\}
\end{aligned}
$$

and the value is tight for words $\mathbf{x}_1 = 1^{w(\mathbf{x}_1)}0^{n-w(\mathbf{x}_1)}$ *and* $\mathbf{x}_2 = 1^{w(\mathbf{x}_1)-b}0^{n+b-w(\mathbf{x}_1)}$.

Proof. Let $w(\mathbf{x}_1) = w(\mathbf{x}_2)+b$ for some $b \geq 1$ for two binary words on symbols 0 and 1. Let us consider the multiset of output words we can obtain from both of these words. We can observe that we need to delete at least b symbols 1 from $\mathbf{x}_1$ and at least b symbols 0 from $\mathbf{x}_2$. Clearly, $b \leq t$. Moreover, if we delete exactly i symbols 1 from $\mathbf{x}_1$ and exactly $t-i$ symbols 0 from $\mathbf{x}_1$ to obtain some output word, then to obtain the resulting output word we need to delete exactly $i-b$ symbols 1 and exactly $t+b-i$ symbols 0 from $\mathbf{x}_2$. Hence, we may obtain these output words from at most $\binom{w(\mathbf{x}_1)}{i}\binom{n-w(\mathbf{x}_1)}{t-i}$ channels from $\mathbf{x}_1$ and from at most $\binom{w(\mathbf{x}_1)-b}{i-b}\binom{n-w(\mathbf{x}_1)+b}{t+b-i}$ channels from $\mathbf{x}_2$. In other words, we can obtain them from at most

$$N = \sum_{i=b}^{t} \min\left\{\binom{w(\mathbf{x}_1)}{i}\binom{n-w(\mathbf{x}_1)}{t-i}, \binom{w(\mathbf{x}_1)-b}{i-b}\binom{n-w(\mathbf{x}_1)+b}{t+b-i}\right\}$$

channels, as claimed. Finally, we may observe that if $\mathbf{x}_1 = 1^{w(\mathbf{x}_1)}0^{n-w(\mathbf{x}_1)}$ and $\mathbf{x}_2 = 1^{w(\mathbf{x}_1)-b}0^{n+b-w(\mathbf{x}_1)}$, then we have the same output word multiset for N channels so the upper bound is tight. □

In the following proposition, we examine more closely the case from the previous lemma with $b = 1$.

Proposition 18. *Let n be even and let exactly t deletions occur in any channel in the multiset model. If $w(\mathbf{x}_1) = w(\mathbf{x}_2)+1$ for two binary words on symbols 0 and 1, then there exists a pair $\mathbf{x}$ and $\mathbf{x}'$ of binary words on symbols 0 and 1 with $w(\mathbf{x}) = w(\mathbf{x}') = 1$ such that we require at least as many channels for distinguishing between $\mathbf{x}$ and $\mathbf{x}'$ as we need for distinguishing between $\mathbf{x}_1$ and $\mathbf{x}_2$.*

Proof. Let $w(\mathbf{x}_1) = w(\mathbf{x}_2)+1$. By Lemma 17 with $b = 1$, we may obtain the same output word multiset from both $\mathbf{x}_1$ and $\mathbf{x}_2$ when

$$N = \sum_{i=1}^{t} \min\left\{\binom{w(\mathbf{x}_1)}{i}\binom{n-w(\mathbf{x}_1)}{t-i} \binom{w(\mathbf{x}_1)-1}{i-1}\binom{n-w(\mathbf{x}_1)+1}{t+1-i}\right\}. \tag{7}$$

Moreover, this is attained by the pair $\mathbf{x}_1 = 1^{w(\mathbf{x}_1)}0^{n-w(\mathbf{x}_1)}$ and $\mathbf{x}_2 = 1^{w(\mathbf{x}_1)-1}0^{n+1-w(\mathbf{x}_1)}$. Since we are interested in the case, where we require the largest number of channels for distinguishing between two input words, we assume from now on that $\mathbf{x}_1$ and $\mathbf{x}_2$ are as in the previous sentence. Furthermore, observe that we may assume without loss of generality that $w(\mathbf{x}_1) \geq n/2+1$ and denote $w(\mathbf{x}_1) = w$. Indeed, one of the two words has either more than $n/2$ symbols 0 or 1 and if necessary, we could swap the roles of 0's and 1's. Consider the words $\mathbf{x} = 0^{w-1}10^{n-w}$ and $\mathbf{x}' = 0^{w-2}10^{n+1-w}$. We show that the multiset of output words which can be obtained from $\mathbf{x}$ and $\mathbf{x}'$ is at least as large as the multiset of output words which can be obtained from $\mathbf{x}_1$ and $\mathbf{x}_2$. Let $D = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$ and $D' = \{\mathbf{d}'_1, \ldots, \mathbf{d}'_N\}$ be the sets of deletion vectors of weight t such that, for each $i \in [1, N]$, if we obtain an output word $\mathbf{y}$ from $\mathbf{x}_1$ with $\mathbf{d}_i \in D$, then we also obtain it from $\mathbf{x}_2$ with $\mathbf{d}'_i \in D'$. Furthermore, we make an observation that if $w(\mathbf{x}_1) \in \mathrm{supp}(\mathbf{d})$ for $\mathbf{d} \in D$ and applying $\mathbf{d}$ to $\mathbf{x}_1$ gives an output word $\mathbf{y}$, then applying $\mathbf{d}$ to $\mathbf{x}_2$ also gives the same output word $\mathbf{y}$. Hence, we may assume that D and D' contain every deletion vector of weight t which have $w(\mathbf{x}_1)$ in their supports. There are $\binom{n-1}{t-1}$ such deletion vectors. Similarly, for $\mathbf{x}$ and $\mathbf{x}'$, we know that deletion vectors which contain w or $w-1$, respectively, in their supports lead to the same output words containing only 0's and there are $\binom{n-1}{t-1}$ such deletion vectors. Thus, we omit these deletion vectors from all the following considerations and calculations. Moreover, let us denote by $D_{\mathbf{y}} \subseteq D$ the set of all deletion vectors of D which result to $\mathbf{y}$ after applying them to $\mathbf{x}_1$. Similarly, we denote by $D'_{\mathbf{y}} \subseteq D'$ the set of all deletion vectors of D' which result to $\mathbf{y}$ after applying them to $\mathbf{x}_2$. Recall that if applying $\mathbf{d}_i \in D$ to $\mathbf{x}_1$ leads to $\mathbf{y}$, then applying $\mathbf{d}'_i \in D'$ to $\mathbf{x}_2$ leads to $\mathbf{y}$. In particular, we have $|D_{\mathbf{y}}| = |D'_{\mathbf{y}}|$ for each $\mathbf{y}$ (since we consider the multiset model). Notice that sets $D_{\mathbf{y}}$ partition D and sets $D'_{\mathbf{y}}$ partition D'.

We show that for each $D_{\mathbf{y}}$ (where $\mathbf{y}$ does not belong to the above omitted output words), we can injectively link another output word $\mathbf{y}'$ which can be attained with at least $|D_{\mathbf{y}}|$ deletion vectors from both $\mathbf{x}$ and $\mathbf{x}'$. Let $\mathbf{y} = 1^{w-i}0^{n+i-w-t}$ and $\mathbf{y}' = 0^{w-1-i}10^{n+i-w-t}$ for $i \in [1, w-1]$ (note that since $w(\mathbf{y}') > 0$, it was not omitted above). We may assume that $i \leq w-1$, due to the previous omissions. Clearly, we also have $i \leq t$. Let us use following notation:

$$\begin{aligned} m_1 &= \binom{w-1}{i}\binom{n-w}{t-i}, \\ m_2 &= \binom{w-1}{i-1}\binom{n-w}{t+1-i}, \\ m &= \binom{w-1}{i}\binom{n-w}{t-i}, \\ m' &= \binom{w-2}{i-1}\binom{n+1-w}{t+1-i}. \end{aligned}$$

We can obtain $\mathbf{y}$ from $\mathbf{x}_1$ with m_1 deletion vectors and from $\mathbf{x}_2$ with m_2 deletion vectors. Moreover, we may obtain $\mathbf{y}'$ from $\mathbf{x}$ with m deletion vectors and from $\mathbf{x}'$ with m' deletion vectors.

Notice that $m_1 = m$. Next, we consider the values of $i \in [1, \min\{w-1, t\}]$ for which $m' \geq m_2$. Recall that $\binom{a}{b} = 0$ if $b < 0$ or $b > a$. Note that both m' and m_2 obtain value 0 with the same values of i, with the possible exception that $m_2 = 0$ and $m' \geq 1$ when $i = w+t-n$. Further note that, since $i \in [1, w-1]$, the left binomial coefficient in each of four parameters is always positive. When both m' and m_2 obtain positive values, we have

$$\begin{aligned} \frac{m'}{m_2} = \frac{(n+1-w)(w-i)}{(w-1)(n+i-w-t)} &\geq 1 \\ \Leftrightarrow \quad n-t+wt &\geq ni \\ \Leftrightarrow \quad 1+\frac{t(w-1)}{n} &\geq i. \end{aligned}$$

Denote above $P = 1+\frac{t(w-1)}{n}$. Let us next consider when we use m and when m'. Recall that for each i, we are interested in the one that is smaller. Notice that both m and m' obtain value 0 for same values of i. Now for nonzero values

$$\begin{aligned}\frac{m}{m'} = &\frac{(w-1)(t+1-i)}{(n+1-w)i} \geq 1\\ \Leftrightarrow &\ wt+w-t-1 \geq ni\\ \Leftrightarrow &\frac{(w-1)(t+1)}{n} \geq i.\end{aligned}$$

Denote $Q = \frac{(w-1)(t+1)}{n}$. Furthermore, let us compare values m (or m_1) and m_2. Notice that m and m_2 obtain value 0 for the same values of i with the possible exception for $i = t+w-n$ for which we may have $m \geq 1$ and $m_2 = 0$. Now for nonzero values

$$\begin{aligned}\frac{m}{m_2} = &\frac{(w-i)(t+1-i)}{i(n+i-w-t)} \geq 1\\ \Leftrightarrow &\ wt+w \geq ni+i\\ \Leftrightarrow &\frac{w(t+1)}{n+1} \geq i.\end{aligned}$$

Denote $R = \frac{w(t+1)}{n+1}$. Next, we show that $P \geq R \geq Q$ (since $w, t \leq n$). We have

$$\begin{aligned}P-R =&1+\frac{(n+1)t(w-1)-nw(t+1)}{n^2+n}\\ =&1+\frac{tw-nt-t-nw}{n^2+n}\\ \geq&1+\frac{tn-nt-t-n^2}{n^2+n} \geq 0\end{aligned}$$

and

$$\begin{aligned}R-Q =&\frac{nw(t+1)-(n+1)(w-1)(t+1)}{n^2+n}\\ =&\frac{(n+1-w)(t+1)}{n^2+n} > 0.\end{aligned}$$

We note that for $i \in [1, \min\{t, w-1\}]$ when $m = 0$, we also have $m_1, m_2 = 0$ and also when $m' = 0$, we have $m_1, m_2 = 0$, as we can see from above together with the equality $m = m_1$. Consider next the cases with $i \geq P$ and $P > i \geq R$. In both of these cases $m_2 \geq m = m_1$. Now, we obtain $\mathbf{y}$ and $\mathbf{y}'$ with $m = m_1$ ways since $m' \geq m$ and $m_2 \geq m_1$. If $R > i \geq Q$, then $m' \geq m = m_1 \geq m_2$ and we obtain $\mathbf{y}$ with m_2 deletion vectors and $\mathbf{y}'$ with $m \geq m_2$ deletion vectors. Finally, if $i < Q$, then $m = m_1 \geq m' \geq m_2$ and we obtain $\mathbf{y}$ with m_2 deletion vectors and $\mathbf{y}'$ with $m' \geq m_2$ deletion vectors. Thus, in all three cases we can obtain $\mathbf{y}'$ in at least as many ways as we can obtain $\mathbf{y}$. Therefore, for any transmitted word pair $\mathbf{x}_1, \mathbf{x}_2$ with difference of exactly 1 in their weights, there exists another word pair $\mathbf{x}$ and $\mathbf{x}'$ with equal weights of 1 such that we require at least as many channels for distinguishing between $\mathbf{x}$ and $\mathbf{x}'$ as we require for distinguishing between $\mathbf{x}_1$ and $\mathbf{x}_2$. □

Remark 19. In this remark we discuss word pairs leading to the largest channel numbers in the different models when exactly t deletion errors occur.

1) In the traditional model, for even n and $q = 2$, the extremal word pair (up to permutation of symbols) is (see [19, proof of Lemma 1]) the pair $\mathbf{x} = 01010101\cdots01$, $\mathbf{x}' = 10010101\cdots01$ and for $n = 2+hq$, $q \geq 2$, $h \in \mathbb{N}$, the extremal word pair is $\mathbf{x} = 0123\cdots(q-1)012\cdots(q-1)$, $\mathbf{x}' = 1023\cdots(q-1)012\cdots(q-1)$, that is the only difference is in the first two symbols and the words continue afterwards as alternating words.
2) In the deletion pattern model with non-multisets and even n, an extremal word pair is $\mathbf{x} = 0^{n/2}10^{n/2-1}$, $\mathbf{x}' = 0^{n/2-1}10^{n/2}$ by Theorems 9(ii) and 10(ii).
3) In the deletion pattern model with multisets, even n and odd t, the word pair $\mathbf{x} = 0^{n/2}10^{n/2-1}$, $\mathbf{x}' = 0^{n/2-1}10^{n/2}$, which is given in Corollary 14 (up to a permutation of symbols), seems to require the largest number of channels. Indeed by the proof of Proposition 11, it is an extremal word pair for $t = 1$. Furthermore, it is easy to check by computer, using a brute-force method finding every extremal word pair, that this pair actually belongs to the set of *extremal* word pairs when $t = 3$ and $n \in \{8, 10\}$.
4) In the deletion pattern model with multisets, $n = 2h(t+1)$ and even t, the word pair which seems to be requiring the largest number of channels, which is presented in Corollary 13 (up to permutation of symbols), is $\mathbf{x} = 0^{ht}10^{h(t+2)-1}$, $\mathbf{x}' = 0^{ht-1}10^{h(t+2)}$. It is easy to check by computer with a brute-force method that this pair belongs to the set of extremal word pairs when $t = 2$ and $n \in \{6, 12\}$.

In the subsequent lemma, we give a tool for comparing the number of channels required in the worst case of the non-multiset deletion vector version compared to the multiset version.

Lemma 20. *Let $n \geq 2t+2$ and t be even positive integers. We have*

$$\frac{\binom{n/2-1}{t/2}^2}{\binom{n/2-1}{t}} \overset{n\to\infty}{\longrightarrow} \binom{t}{t/2}.$$

Proof. We have

$$\begin{aligned}\frac{\binom{n/2-1}{t/2}^2}{\binom{n/2-1}{t}} =&\frac{(n/2-1)!t!(n/2-1-t)!}{(n/2-1-t/2)!(n/2-1-t/2)!(t/2)!(t/2)!}\\ =&\binom{t}{t/2}\frac{(n/2-1)\cdots(n/2-t/2)}{(n/2-1-t/2)\cdots(n/2-t)}\\ \overset{n\to\infty}{\longrightarrow}&\binom{t}{t/2}.\end{aligned}$$

□

For an even t when exactly t deletions occur, by Theorems 9(ii) and 10 we require $\binom{n}{t} - \binom{n/2-1}{t} + 1$ channels in the non-multiset model to separate extremal words presented in the case 2) of Remark 19. The same word pair is also mentioned in Remark 19 for the multiset model with even t and by Corollary 14, we require $\binom{n}{t} - \binom{n/2-1}{t/2}^2 + 1$ channels to distinguish between these words. By Lemma 20 when n is large, we

require roughly

$$\left(\binom{n}{t}-\binom{n/2-1}{t}+1\right)-\left(\binom{n}{t}-\binom{n/2-1}{t/2}^2+1\right)$$
$$\approx\left(\binom{t}{t/2}-1\right)\binom{n/2-1}{t}$$

more channels in the non-multiset model compared to the multiset case.

IV. Decoding

In this section, we consider channels with insertion, deletion and substitution errors using an underlying code containing almost all words of $\mathbb{Z}_q^n$. We assume that each insertion vector is applied to the word of length n. Then deletion vectors and substitutions are applied to original non-inserted symbols and no deletion affects the substituted symbols. We assume that each error pattern has the same probability. Unlike in the previous section, in this section we allow multiple channels to have the same error patterns. In particular, if only substitution errors occur, then each possible output word has the same probability to be outputted as we have seen in the beginning of Section II. However, in the case of deletion and insertion errors, some output words are more likely. For the rest of the section, we focus on $q \geq 4$. Notice that the presented technique cannot be expanded to the cases with $q < 4$ as will be seen in Remark 26. Moreover, the case with $q = 4$ is a natural size of alphabet for DNA-storage. The case with $q = 4$ is presented in the conference version of this article [1] without a proof.

For channels with insertion, deletion and substitution errors, we introduce, for a code with minor restrictions, a decoding algorithm with complexity $O(Nn)$, where N is the number of output words read at the point in which the algorithm halts (see Algorithm 1). Our algorithm never gives an incorrect result. However, for some output sets Y it only outputs an empty word. When we discuss about complexities, we assume q to be constant. The code we are using has only minor restrictions on how common the two most common symbols in any codeword can be. Moreover, similar restrictions have been used for example in [24]. Besides giving verifiability properties and solving all three types of errors simultaneously, the novelty of our technique is that we do not use majority decoding which has been an essential part of most earlier techniques. We note that due to this property, our algorithm can solve systematic errors as long as they only affect on some portion of words in Y. Indeed, our algorithm only requires a small subset of well-chosen output words based on which the original word is reconstructed.

Algorithm 1 is an online algorithm in the sense that the output words of the channels can be viewed to be fed to the algorithm one by one (instead of giving all the outputs at once). In this context, the number N of channels is assumed to denote the number of outputs required before the algorithm stops. Moreover, the algorithm is sort of a randomized one in style of a Las Vegas algorithm, although technically the randomization occurs outside of the algorithm in obtaining the output words of the channels. However, in Las Vegas style, if the number of the output words is unrestricted, then the algorithm is not guaranteed to halt (although it is highly likely), but if the algorithm halts, then it always gives a correct result.

Probabilistic decoders have been previously mostly considered for a setup, where each error to a single coordinate has an independent chance to occur, under the name *trace reconstruction*; see, for example, [22] in the case of deletion channels and [24] in the case of simultaneous insertion, deletion and substitution errors. Unlike in these setups, we limit the maximum number of errors which may occur in a channel, as has been done, for example, by Levenshtein in [19]. That allows our algorithm to have verifiability, that is, although the algorithm is probabilistic, it is likely that the algorithm halts (see Lemma 24), and the output is always correct if the algorithm halts (see Lemma 23).

Let code $C \subseteq \mathbb{Z}_q^n$ contain all the words of $\mathbb{Z}_q^n$ except for those in which the two most common symbols appear together in total in at least $\lceil (p-1)n/p \rceil$ positions with $p = 2^4/e$. Observe that there are

$$\binom{q}{2}\left(\sum_{i=\lceil\frac{(p-1)n}{p}\rceil}^{n}\binom{n}{i}2^i(q-2)^{n-i}\right)$$

such words. Due to these restrictions on C, the third most common symbol (and also second most common symbol) in any codeword occurs in at least $\lceil n/((q-2)p) \rceil$ coordinates by the pigeonhole principle. Notice that p is irrational and hence, $(p-1)n/p$ is not an integer. Our results in this section require that we are using code C (or some sub-code of C). We next show that C is *large* when q is fixed and n is large. In order to estimate the cardinality of C, we first consider the case with $q = 4$. We have

$$\begin{aligned}
|C| \geq& 4^n - \binom{4}{2}\left(\sum_{i=\lceil\frac{(p-1)n}{p}\rceil}^{n}\binom{n}{i}2^i(4-2)^{n-i}\right)\\
=& 4^n - 6\cdot 4^{n/2}\sum_{i=\lceil\frac{(p-1)n}{p}\rceil}^{n}\binom{n}{i}\\
\geq& 4^n - 6\cdot 4^{n/2}(ep)^{n/p} \qquad (*)\\
=& 4^n - 6\cdot 4^{n/2}4^{2en/2^4}\\
>& 4^n - 6\cdot 4^{7n/8} \in \Theta(q^n). \qquad (8)
\end{aligned}$$

In Inequality (*), we use the following modification of a well-known upper bound for partial binomial sums:

$$\sum_{i=0}^{\lfloor h\rfloor}\binom{K}{i} \leq \left(\frac{en}{h}\right)^h,$$

where $K \in \mathbb{Z}$ and $K \geq h > 0$. Indeed, this upper bound holds since

$$\begin{aligned}
\sum_{i=0}^{\lfloor h\rfloor}\binom{K}{i} \leq \sum_{i=0}^{\lfloor h\rfloor}\frac{h^i}{i!}\cdot\left(\frac{K}{h}\right)^i \leq \left(\frac{K}{h}\right)^{\lfloor h\rfloor}\sum_{i=0}^{\lfloor h\rfloor}\frac{h^i}{i!}\\
< \left(\frac{K}{h}\right)^{\lfloor h\rfloor}e^h \leq \left(\frac{eK}{h}\right)^h.
\end{aligned}$$

In particular, for $\sum_{i=\lceil \frac{(p-1)n}{p} \rceil}^{n} \binom{n}{i}$ it gives:

$$\sum_{i=\lceil \frac{(p-1)n}{p} \rceil}^{n} \binom{n}{i} = \sum_{i=0}^{\lfloor n/p \rfloor} \binom{n}{i} \leq \left(\frac{en}{n/p}\right)^{n/p} = (ep)^{n/p}.$$

We can use similar arguments for the case with $q \geq 5$. Let $q = 2^b \geq 5$, $b = b' + \log_2 5$ where $b' \geq 0$. We have

$$\begin{aligned}
|C| =& q^n - \binom{q}{2} \left(\sum_{i=\lceil \frac{(p-1)n}{p} \rceil}^{n} \binom{n}{i} 2^i (q-2)^{n-i} \right) \\
>& q^n - \frac{q^2}{2} \cdot 2^n \cdot q^{n/p} \sum_{i=\lceil \frac{(p-1)n}{p} \rceil}^{n} \binom{n}{i} \\
>& q^n - q^2 \cdot 2^n \cdot 2^{bn/p} (ep)^{n/p} \\
\geq& q^n - 2^{2b} \cdot 2^n \cdot 2^{ebn/16} \cdot 2^{4en/16} \\
=& q^n - 2^{2b} \cdot 2^{n + e\log_2(5)n/16 + 4en/16 + eb'n/16} \\
>& q^n - 2^{2b} \cdot 2^{2.08n + b'n/5} \in \Theta(q^n). \qquad (9)
\end{aligned}$$

The inclusion (9) follows from $q^n = 2^{bn} = 2^{\log_2 5 n + b'n} > 2^{2.32n + b'n}$ and the facts that $2.32 > 2.08$ and $b' > b'/5$. By (8) and (9), we have $|C| \in \Theta(q^n)$ for all integers $q \geq 4$. Note that code C excludes, in particular, the extremal words presented in Remark 19 points 2 to 4.

We denote by t_s, t_i and t_d the number of substitution, insertion and deletion errors, respectively, which may occur in a channel. When we discuss about the complexity of our algorithm, these values are assumed to be constants. Moreover, Lemma 24 gives them some minor constraints. Recall that for our Las Vegas algorithm, the underlying code $C \subset \mathbb{Z}_q^n$ is required to be such that in each codeword the two most common symbols appear in total in at most $(p-1)n/p \approx 0.83n$ positions. When $p \geq 2^4/e \approx 5.9$, we have $|C| \in \Theta(q^n)$.

Remark 21. Observe that if we increase the value of p from $2^4/e$, then that will increase the size of the code C. However, we have a trade-off later in the proof of Lemma 24; the larger p is the less likely Algorithm 1 is to stop.

We denote $t_m = t_d + t_i + 2t_s$ and for a word $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ we denote

$$M_i(\mathbf{w}) = |\{j \mid w_j = i \in \mathbb{Z}_q\}|$$

and

$$M_{a,b,c}(\mathbf{w}) = |\{j \mid w_j \notin \{a,b,c\}\}|.$$

The useful observation behind Algorithm 1 is that $M_i(\mathbf{y}) + t_m \geq M_i(\mathbf{y}')$ for every i and any two output words $\mathbf{y}, \mathbf{y}' \in Y$ and this bound can be attained when $M_i(\mathbf{y}) \geq t_d + t_s$. This observation is further discussed in the proof of the following lemma.

Lemma 22. *Let a, b and c be distinct symbols of $\mathbb{Z}_q$.*

1) *If $\mathbf{y}_1, \mathbf{y}_2 \in Y$ are such that $M_a(\mathbf{y}_1) = M_a(\mathbf{y}_2) + t_m$, then $\mathbf{y}_1$ is formed from the transmitted word $\mathbf{x}$ by inserting t_i symbols a and substituting t_s symbols by a, and $\mathbf{y}_2$ is formed from $\mathbf{x}$ by deleting t_d symbols a and substituting t_s symbols a with other symbols.*
2) *If $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3 \in Y$ are such that $M_a(\mathbf{y}_1) = M_a(\mathbf{y}_2) + t_m$ and $M_b(\mathbf{y}_3) = M_b(\mathbf{y}_1) + t_m$, then $\mathbf{y}_1$ is formed from $\mathbf{x}$ by inserting t_i symbols a, substituting t_s symbols b by a and deleting t_d symbols b.*
3) *If $\mathbf{y}_1, \mathbf{y}_2 \in Y$ are such that $M_a(\mathbf{y}_1) = M_a(\mathbf{y}_2) + t_m$ and $M_{a,b,c}(\mathbf{y}_2) = M_{a,b,c}(\mathbf{y}_1) + t_i + t_s$, then $\mathbf{y}_2$ is formed from $\mathbf{x}$ by inserting t_i symbols other than a, b or c, substituting t_s symbols a by symbols other than a, b or c and deleting t_d symbols a.*

Proof. Recall that $t_m = t_i + t_d + 2t_s$. Let us first prove Claim 1. Observe that we have $M_a(\mathbf{y}_1) \leq M_a(\mathbf{x}) + t_i + t_s$ since only insertions and substitutions may increase the number of symbols a in an output word and that the equality holds only when all insertions and substitutions increase the number of symbols a. Furthermore, we have $M_a(\mathbf{y}_2) \geq M_a(\mathbf{x}) - t_s - t_d$ since only deletions and substitutions may decrease the number of symbols a in an output word, and the equality holds only when all deletions and substitutions decrease the number of symbols a. Thus, $M_a(\mathbf{y}_1) \leq M_a(\mathbf{y}_2) + t_m$ and the equality holds only when all insertions and substitutions increase the number of symbols a in $\mathbf{y}_1$ and all deletions and substitutions decrease the number of symbols a in $\mathbf{y}_2$. Hence, the Claim follows.

Claim 2. is a direct corollary from Claim 1.

Claim 3. follows from Claim 1. Indeed, by Claim 1, each symbol deleted or substituted out of $\mathbf{x}$ to form $\mathbf{y}_2$ is a. Moreover, word $\mathbf{y}_2$ has $t_i + t_s$ more symbols in total in the set $\mathbb{Z}_q \setminus \{a,b,c\}$ than word $\mathbf{y}_1$. Thus, each symbol which we insert or substitute to $\mathbf{x}$ to form $\mathbf{y}_2$ is in $\mathbb{Z}_q \setminus \{a,b,c\}$. □

In the following lemmas, we first show that Algorithm 1 never gives an incorrect output and that it is efficient. Then we show that we are likely to find the set Y_6.

Lemma 23. *If, after reading exactly N inputs, we find output words $\mathbf{y}_i \in Y_6$, $i \in [1,6]$, defined in Step 17 of Algorithm 1, then $\mathbf{c} = \mathbf{x}$ in Algorithm 1, and the algorithm halts in $O(Nn)$ time.*

Proof. Let $\mathbf{y}_i$ and $\mathbf{z}_i$, $i \in [1,6]$, be as in Algorithm 1. Consider first the output words $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$. Observe that $M_{i_1}(\mathbf{y}_1) = M_{i_1}(\mathbf{y}_2) + t_m$ and $M_{i_3}(\mathbf{y}_3) = M_{i_3}(\mathbf{y}_1) + t_m$. Therefore, by Lemma 22, each of t_i inserted symbols in $\mathbf{y}_1$ is i_1, each of t_d deleted symbols is i_3 and all t_s substitutions change symbols i_3 to symbols i_1. Consequently, the output word $\mathbf{y}_1$ is obtained from the (unknown) transmitted word $\mathbf{x}$ by modifying only the symbols i_1 and i_3. Similarly (due to the three first equations in Step 17) modifications to $\mathbf{x}$ in obtaining $\mathbf{y}_2$ affect only the symbols i_1 and i_2 and modifications to $\mathbf{x}$ in obtaining $\mathbf{y}_3$ affect only the symbols i_2 and i_3. Observe that at this point we know the exact number of each symbol in $\mathbf{x}$ (but not their order). In particular,

$$M_{i_1}(\mathbf{x}) = M_{i_1}(\mathbf{y}_3), M_{i_2}(\mathbf{x}) = M_{i_2}(\mathbf{y}_1), M_{i_3}(\mathbf{x}) = M_{i_3}(\mathbf{y}_2)$$

and

$$M_{i_4}(\mathbf{x}) = M_{i_4}(\mathbf{y}_1) = M_{i_4}(\mathbf{y}_2) = M_{i_4}(\mathbf{y}_3)$$

for any $i_4 \notin \{i_1, i_2, i_3\}$. Let us then consider the output words $\mathbf{y}_4, \mathbf{y}_5$ and $\mathbf{y}_6$.

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2025.3568068

Algorithm 1 Decoding in $\mathbb{Z}_q^n$

Input: At least six output words $Y = \{\mathbf{y}_i \mid i \in \mathbb{Z}_+\}$
Output: Transmitted word $\mathbf{c}(= \mathbf{x})$ or empty word ε

1: Let $i = 1$, $\mathbf{c} = \varepsilon$, collection $Y_2 = \{\mathbf{y}_{a,b} \mid a \neq b, a, b \in \mathbb{Z}_q\}$, collection $Y_3 = \{\mathbf{y}_{a,b,c} \mid a \neq b \neq c \neq a, a, b, c \in \mathbb{Z}_q\}$, $Y_6 = \emptyset$ and $\mathbf{y}_{a,b,c} = \mathbf{y}_{a,b} = \mathbf{y}_1$ for each $a \neq b \neq c \neq a$
2: **while** $Y_6 = \emptyset$ and $i \leq |Y|$ **do**
3: Read $\mathbf{y}_i \in Y$
4: **for** each $j, j', j'' \in [0, q-1]$ with $j \neq j' \neq j'' \neq j$ **do**
5: calculate $M_j(\mathbf{y}_i)$ and $M_{j,j',j''}(\mathbf{y}_i)$
6: **end for**
7: **for** each $\mathbf{y}_{a,b} \in Y_2$ **do**
8: **if** $M_a(\mathbf{y}_i) \leq M_a(\mathbf{y}_{a,b})$ and $M_b(\mathbf{y}_i) \geq M_b(\mathbf{y}_{a,b})$ **then**
9: Set $\mathbf{y}_{a,b} := \mathbf{y}_i$ and store $M_a(\mathbf{y}_i)$ and $M_b(\mathbf{y}_i)$
10: **end if**
11: **end for**
12: **for** each $\mathbf{y}_{a,b,c} \in Y_3$ **do**
13: **if** $M_a(\mathbf{y}_i) \leq M_a(\mathbf{y}_{a,b,c})$ and $M_{a,b,c}(\mathbf{y}_i) \geq M_{a,b,c}(\mathbf{y}_{a,b,c})$ **then**
14: Set $\mathbf{y}_{a,b,c} := \mathbf{y}_i$ and store $M_a(\mathbf{y}_i)$ and $M_{a,b,c}(\mathbf{y}_i)$
15: **end if**
16: **end for**
17: **if** there exist in Y_2 words $\mathbf{y}_1 = \mathbf{y}_{i_3,i_1}$, $\mathbf{y}_2 = \mathbf{y}_{i_1,i_2}$, $\mathbf{y}_3 = \mathbf{y}_{i_2,i_3}$ and in Y_3 words $\mathbf{y}_4 = \mathbf{y}_{i_1,i_2,i_3}$, $\mathbf{y}_5 = \mathbf{y}_{i_2,i_1,i_3}$ and $\mathbf{y}_6 = \mathbf{y}_{i_3,i_1,i_2}$ such that $i_j \neq i_h$ for all distinct j, h as well as

$$M_{i_1}(\mathbf{y}_1) = M_{i_1}(\mathbf{y}_2) + t_m,\ M_{i_1}(\mathbf{y}_2) = M_{i_1}(\mathbf{y}_4),$$
$$M_{i_2}(\mathbf{y}_2) = M_{i_2}(\mathbf{y}_3) + t_m,\ M_{i_2}(\mathbf{y}_3) = M_{i_2}(\mathbf{y}_5),$$
$$M_{i_3}(\mathbf{y}_3) = M_{i_3}(\mathbf{y}_1) + t_m,\ M_{i_3}(\mathbf{y}_1) = M_{i_3}(\mathbf{y}_6) \text{ and}$$
$$M_{i_1,i_2,i_3}(\mathbf{y}_4) = M_{i_1,i_2,i_3}(\mathbf{y}_5) = M_{i_1,i_2,i_3}(\mathbf{y}_6) = M_{i_1,i_2,i_3}(\mathbf{y}_1) + t_i + t_s$$

18: **then**
19: Set $Y_6 = \{\mathbf{y}_j \mid j \in [1, 6]\}$
20: **end if**
21: Set $i = i + 1$
22: **end while**
23: **if** $Y_6 = \emptyset$ **then**
24: **return** empty word
25: **end if**
26: Delete each i_1 and i_3 from $\mathbf{y}_1(= \mathbf{y}_{i_3,i_1})$ to construct $\mathbf{z}_1$
27: Delete each i_1 and i_2 from $\mathbf{y}_2(= \mathbf{y}_{i_1,i_2})$ to construct $\mathbf{z}_2$
28: Delete each i_2 and i_3 from $\mathbf{y}_3(= \mathbf{y}_{i_2,i_3})$ to construct $\mathbf{z}_3$
29: Delete everything except each i_2 and i_3 from $\mathbf{y}_4(= \mathbf{y}_{i_1,i_2,i_3})$ to construct $\mathbf{z}_4$
30: Delete everything except each i_1 and i_3 from $\mathbf{y}_5(= \mathbf{y}_{i_2,i_1,i_3})$ to construct $\mathbf{z}_5$
31: Delete everything except each i_1 and i_2 from $\mathbf{y}_6(= \mathbf{y}_{i_3,i_1,i_2})$ to construct $\mathbf{z}_6$
32: **while** there exists an index j such that $\mathbf{z}_j \neq \varepsilon$ **do**
33: **if** exactly three different words $\mathbf{z}_i, \mathbf{z}_j$ and $\mathbf{z}_h$ start with the same symbol a **then**
34: Concatenate $\mathbf{c}$ from right with a
35: Remove the first symbol of $\mathbf{z}_i, \mathbf{z}_j$ and $\mathbf{z}_h$
36: **end if**
37: **end while**
38: **return** $\mathbf{c}$

By the previous observations, we first obtain $M_{i_1}(\mathbf{y}_1) = M_{i_1}(\mathbf{y}_2) + t_m = M_{i_1}(\mathbf{y}_4) + t_m$. Therefore, as $M_{i_1,i_2,i_3}(\mathbf{y}_4) = M_{i_1,i_2,i_3}(\mathbf{y}_1) + t_i + t_s$, we obtain by Lemma 16(3) that $\mathbf{y}_4$ is formed from $\mathbf{x}$ by adding $t_i + t_s$ symbols (with insertions or substitutions) other than i_1, i_2 or i_3 and by removing $t_d + t_s$ symbols i_1 (with deletions or substitutions). Similarly, we obtain that the symbols added to $\mathbf{y}_5$ and $\mathbf{y}_6$ are other that i_1, i_2 or i_3 and the removed symbols are i_2 and i_3, respectively.

Consequently, if we consider the four symbol types examined above, namely i_1, i_2, i_3 and $\mathbb{Z}_q \setminus \{i_1, i_2, i_3\}$. The modifications within each word $\mathbf{y}_i$, $i \in [1, 6]$, with respect to $\mathbf{x}$ are restricted to symbols in two of the examined types. Thus, we know that symbols in $\mathbf{z}_i$ ($i = [1, 6]$) are ordered in the same way as in the transmitted word $\mathbf{x}$, since we have removed all modified symbols from $\mathbf{y}_i$ when we have formed $\mathbf{z}_i$. Furthermore, we have $\binom{4}{2} = 6$ different words $\mathbf{z}_i$ and for each pair of the missing symbol types, we have a word $\mathbf{z}_i$ from which exactly those types are missing.

Next we show that we obtain the transmitted codeword $\mathbf{c} = \mathbf{x}$ in Algorithm 1 during Steps 32–37. If, for example, $x_1 = i_1$, then the first symbol of $\mathbf{z}_3, \mathbf{z}_5$ and $\mathbf{z}_6$ is x_1. Moreover, $\mathbf{z}_1, \mathbf{z}_2$ and $\mathbf{z}_4$ cannot share a common first symbol. The same is true for $x_1 = i_j$ for any $i_j \in \{i_1, i_2, i_3\}$ since words $\mathbf{z}_i$ go through all $\binom{4}{2} = 6$ combinations of missing symbol type pairs among the four examined symbol types. Furthermore, if $x_1 \in \mathbb{Z}_q \setminus \{i_1, i_2, i_3\}$, then x_1 is equal to the first symbol of z_1, z_2 and z_3. Therefore, in all cases, we have $c_1 = x_1$. As we go on, we remove the first symbol from those $\mathbf{z}_i$'s which shared the same symbol. By iteratively applying these arguments, we obtain the rest of the symbols of $\mathbf{x}$.

Let us then consider the complexity of the algorithm. Here, we assume that q is a constant on n. We observe that in the first while loop between Steps 2 and 22, we only do simple coordinatewise comparison operations and the loop lasts at most N rounds. Between Steps 26 and 31, we again make only simple modifications to the words of length $n + t_i - t_d$. Finally, all operations in the final while loop occur to words of length at most $n - t_d$ and the operations are simple. Hence, the complexity of the algorithm is in $O(Nn)$. □

Lemma 24. *As N increases, the probability for obtaining output words $\mathbf{y}_i \in Y_6$, $i \in [1, 6]$, in Step 17 of Algorithm 1 approaches 1 for any $n \geq (q-1)p(t_d + t_s)$.*

Proof. Consider the set Y_6 in Algorithm 1. Recall from the proof of Lemma 23 the separation of symbols into four types i_1, i_2, i_3 and $\mathbb{Z}_q \setminus \{i_1, i_2, i_3\}$. Moreover, we see (as in Lemma 23) from the equations in Step 17 of Algorithm 1 that Y_6 has six words and each of them can be obtained from $\mathbf{x}$ by modifying the symbols of exactly two symbol types. In particular, we observe that for each symbol pair i_j, i_h ($j \neq h$ and

$j,h \in \{1,2,3\}$) there exists a word $\mathbf{y}_i \in Y_6$ which is formed from $\mathbf{x}$ by focusing all the modifications to these two symbols. Moreover, the symbols of $\mathbb{Z}_q \setminus \{i_1,i_2,i_3\}$ are such that they are never removed from $\mathbf{x}$ to form these output words. Moreover, there are multiple possible ways (regarding the symbols) in which we can form the subset Y_6 from Y_2 and Y_3 and it is enough for our claim that we find at least one of these ways. Furthermore, if a set of words in Y satisfies the conditions set for Y_6 in Step 17, then those words are found in Steps 7 to 16. Let us assume without loss of generality that i_1, i_2 and i_3 are the three most common symbols in $\mathbf{x} \in C \subseteq \mathbb{Z}_q^n$ and $M_{i_3}(\mathbf{x}) \leq M_{i_2}(\mathbf{x}) \leq M_{i_1}(\mathbf{x})$. Recall, that our restrictions on code C guarantee, that $M_{i_3}(\mathbf{x}) \geq \lceil n/((q-2)p) \rceil$ by the pigeonhole principle.

Thus, here we consider only the case where we remove symbols i_1, i_2 and i_3. Notice that the likelihood of obtaining exactly this kind set Y_6 is less than the likelihood of obtaining any suitable set Y_6. Now, the least likely case is the one where we remove symbols i_3 from $\mathbf{x}$ since i_3 is the least common among $\{i_1,i_2,i_3\}$. We denote that word by $\mathbf{y}_1$ and the symbol we insert to it is assumed to be i_1 (all symbols have equal probability to be inserted). Notice that since $n \geq (q-1)p(t_d+t_s)$, we have $M_{i_3}(\mathbf{x}) \geq \lceil n/((q-2)p) \rceil \geq t_d + t_s$.

In the subsequent approximations, we will need the following well-known lower bound. If K, h be such non-negative integers that $K \geq 3h-1$, then we have

$$\begin{aligned}
2\binom{K}{h} &= \binom{K}{h} + \frac{K!}{h!(K-h)!} \\
&= \binom{K}{h} + \frac{K!}{(h-1)!(K-h+1)!} \cdot \frac{K-h+1}{h} \\
&\geq \binom{K}{h} + 2\binom{K}{h-1} \qquad (10) \\
&\geq \binom{K}{h} + \binom{K}{h-1} + 2\binom{K}{h-2} \\
&\geq \cdots \geq V_2(K,h).
\end{aligned}$$

Let us first consider the probability to obtain the word $\mathbf{y}_1$. To obtain it, t_i insertions occur and each insertion contains only symbol i_1. Recall that the likelihood of any specific insertion is $1/|B^I_{t_i}(\mathbf{x})|$. First the probability that exactly t_i (for a positive t_i) insertions occur is at least $\frac{1}{t_i+1}$. Indeed, by Equation (4) we have

$$\frac{|B^I_{t_i}(\mathbf{x})| - |B^I_{t_i-1}(\mathbf{x})|}{|B^I_{t_i}(\mathbf{x})|} \geq \frac{q^{t_i}\binom{n+t_i}{t_i}}{(t_i+1)q^{t_i}\binom{n+t_i}{t_i}} = \frac{1}{t_i+1}.$$

Probability that each newly inserted symbol is i_1 is $\left(\frac{1}{q}\right)^{t_i}$.

Next, we give a lower bound for the probability that each deletion and substitution modifies symbol i_3 and that there occurs exactly t_s substitutions and exactly t_d deletions. We assume here that we cannot substitute and delete the same symbol or any inserted symbol. In particular, there are at least $(q-1)^{t_s}\binom{\lceil n/((q-2)p)\rceil}{t_s+t_d}\binom{t_s+t_d}{t_s}$ ways in which the t_s+t_d deletions and substitutions may occur. Moreover, we may apply $i \leq t_s$ substitutions and $j \leq t_d$ deletions to $\mathbf{x}$ in $(q-1)^i\binom{n}{i+j}\binom{i+j}{i}$ different ways. Hence, for the lower bound of the considered probability, we have

$$\begin{aligned}
&\frac{(q-1)^{t_s}\binom{\lceil n/((q-2)p)\rceil}{t_s+t_d}\binom{t_s+t_d}{t_s}}{\sum_{j=0}^{t_d}\sum_{i=0}^{t_s}(q-1)^i\binom{n}{i+j}\binom{i+j}{i}} \\
\geq &\frac{\binom{\lceil n/((q-2)p)\rceil}{t_s+t_d}\binom{t_s+t_d}{t_s}}{\sum_{j=0}^{t_d}\sum_{i=0}^{t_s}\binom{n}{i+j}\binom{i+t_d}{i}} \\
\geq &\frac{\binom{\lceil n/((q-2)p)\rceil}{t_s+t_d}}{\sum_{j=0}^{t_d}\sum_{i=0}^{t_s}\binom{n}{i+j}} \\
\geq &\frac{\binom{\lceil n/((q-2)p)\rceil}{t_s+t_d}}{\sum_{j=0}^{t_d}V_2(n,j+t_s)} \\
\geq &\frac{\binom{\lceil n/((q-2)p)\rceil}{t_s+t_d}}{2\sum_{j=0}^{t_d}\binom{n}{j+t_s}} \qquad (11) \\
\geq &\frac{\binom{\lceil n/((q-2)p)\rceil}{t_s+t_d}}{2V_2(n,t_d+t_s)} \\
\geq &\frac{\binom{\lceil n/((q-2)p)\rceil}{t_s+t_d}}{4\binom{n}{t_d+t_s}} \qquad (12) \\
= &\frac{\lceil n/((q-2)p)\rceil!(n-t_d-t_s)!}{4(\lceil n/((q-2)p)\rceil - t_d - t_s)!n!} \\
\geq &\frac{1}{4}\cdot\left(\frac{\lceil n/((q-2)p)\rceil + 1 - t_d - t_s}{n}\right)^{t_d+t_s} \\
\geq &\frac{1}{4}\cdot\left(\frac{1}{(q-2)p} - \frac{t_d+t_s}{n}\right)^{t_d+t_s} \\
\geq &\frac{1}{4}\cdot\left(\frac{1}{(q-2)(q-1)p}\right)^{t_d+t_s}.
\end{aligned}$$

Inequalities (11) and (12) are due to Inequality (10). Observe that the condition $K \geq 3h-1$ in Inequality (10) is satisfied since $n > 3(t_d+t_s)$.

Finally, the probability that each substitution produces i_1 is $\left(\frac{1}{q-1}\right)^{t_s}$.

Observe that each of these probabilities is positive and can be bounded from below by a positive constant

$$A \geq \left(\frac{1}{q-1}\right)^{t_s}\cdot\frac{1}{4}\left(\frac{1}{(q-2)(q-1)p}\right)^{t_d+t_s}\cdot\left(\frac{1}{q}\right)^{t_i}\cdot\frac{1}{t_i+1} \qquad (13)$$

which does not depend on n. Hence, the probability for not obtaining $\mathbf{y}_1$ in a channel is at most $(1-A)^N$ which tends to 0 as N grows. Furthermore, we are less or equally likely to obtain $\mathbf{y}_1$ than $\mathbf{y}_i$ for other values of i since $M_{i_3}(\mathbf{x}) \leq M_{i_2}(\mathbf{x}) \leq M_{i_1}(\mathbf{x})$. Note that for $\mathbf{y}_4, \mathbf{y}_5$ and $\mathbf{y}_6$ we may have more options (depending on whether $q \geq 5$) for symbols which we can insert or substitute into these words and hence, the probability to obtain these words is at least the same as the probability to obtain $\mathbf{y}_1$. Thus, the probability to obtain the output words in Y_6 tends to 1 as N grows. □

In the following example, we consider how Algorithm 1 works after we have obtained output words in Y_6.

Example 25. Consider the transmitted word $\mathbf{x} \in \mathbb{Z}_6^{10}$ in Table II together with $t_i = 2, t_d = t_s = 1$, output set Y_6 and words $\mathbf{z}_i$. We have presented words $\mathbf{y}_j \in Y_6$ in the table. Notice that values q, t_d, t_s and n do not satisfy condition $n \geq (q-1)p(t_d+t_s)$ of Lemma 24. However, this is not a problem

TABLE II
WORD $\mathbf{x}$, SET Y_6 AND WORDS $\mathbf{z}_i$.

$\mathbf{x}$	1	2	0	0	3	2	1	0	2	1	
$\mathbf{y}_1(=\mathbf{y}_{2,0})$	1	0	0	0	3	0	1	0	2	1	0
$\mathbf{y}_2(=\mathbf{y}_{0,1})$	1	2	1	3	2	1	1	0	1	2	1
$\mathbf{y}_3(=\mathbf{y}_{1,2})$	2	2	0	0	3	2	0	2	2	1	2
$\mathbf{y}_4(=\mathbf{y}_{0,1,2})$	3	1	2	0	3	2	1	3	2	4	1
$\mathbf{y}_5(=\mathbf{y}_{1,0,2})$	3	4	2	0	3	0	3	2	0	2	1
$\mathbf{y}_6(=\mathbf{y}_{2,0,1})$	3	1	0	0	3	3	5	1	0	2	1
$\mathbf{z}_1$	1	3	1	1							
$\mathbf{z}_2$	2	3	2	2							
$\mathbf{z}_3$	0	0	3	0							
$\mathbf{z}_4$	1	2	2	1	2	1					
$\mathbf{z}_5$	2	0	0	2	0	2					
$\mathbf{z}_6$	1	0	0	1	0	1					

since the requirement was established only for making sure that we obtain set Y_6 with high probability and hence, we do not have to worry about Lemma 24.

Let us now consider Steps from 26 to 31 of the algorithm.

1) $c_1 = 1$ and the first bits of $\mathbf{z}_1, \mathbf{z}_4$ and $\mathbf{z}_6$ are deleted.
2) $c_2 = 2$ and the first bits of $\mathbf{z}_j$ are deleted ($j \in \{2, 4, 5\}$).
3) $c_3 = 0$ and the first bits of $\mathbf{z}_j$ are deleted ($j \in \{3, 5, 6\}$). We continue iterating the process in this way.
4) $c_4 = 0$ and the first bits of $\mathbf{z}_j$ are deleted ($j \in \{3, 5, 6\}$).
5) $c_5 = 3$ and the first bits of $\mathbf{z}_j$ are deleted ($j \in \{1, 2, 3\}$). At this point, we have $\mathbf{z}_1 = 11$, $\mathbf{z}_2 = 22$, $\mathbf{z}_3 = 0$, $\mathbf{z}_4 = 2121$, $\mathbf{z}_5 = 202$ and $\mathbf{z}_6 = 101$.
6) $c_6 = 2$ and the first bits of $\mathbf{z}_j$ are deleted ($j \in \{2, 4, 5\}$).
7) $c_7 = 1$ and the first bits of $\mathbf{z}_j$ are deleted ($j \in \{1, 4, 6\}$).
8) $c_8 = 0$ and the first bits of $\mathbf{z}_j$ are deleted ($j \in \{3, 5, 6\}$). Word $\mathbf{z}_3$ becomes empty but the algorithm continues.
9) $c_9 = 2$ and the first bits of $\mathbf{z}_j$ are deleted ($j \in \{2, 4, 5\}$).
10) Finally, we get $c_{10} = 1$. Now, $\mathbf{c} = \mathbf{x}$ as claimed.

Remark 26. Algorithm 1 requires that $q \geq 4$. Let us consider the case with $q = 3$. If the insertion, deletion and substitution errors occur in some word $\mathbf{y}$, for example, to symbols 0 and 1, then we only know how many symbols 2 there are in $\mathbf{x}$ but we do not know their location in respect to other symbols. This prevents us from reconstructing the transmitted word $\mathbf{x}$ in a similar way.

Recall Lemma 24 in which we showed that the probability of finding a suitable set Y_6 of output words in the algorithm approaches 1 as N increases. In addition to the asymptotical result of the lemma, we have also run some simulations for obtaining estimates on the exact number of required channels when $q = 4$. The simulations have been performed in a rather simple and straightforward manner: The given number of (at most) t_s substitution, t_d deletion and t_i insertion errors have been randomly applied to an arbitrarily chosen transmitted word $\mathbf{x} \in C$ and then channel outputs have been read until the set Y_6 has been obtained. In Table III, for chosen lengths n and number of different errors, we have given an average and median number of channels required when the simulations have run for 100000 samples. It should be noted that in each case the number of 100000 samples seems to be enough for the average and median values to converge to the extent that they give a sensible approximation on the number of required channels.

TABLE III
THE SIMULATIONS WITH 100000 SAMPLES FOR APPROXIMATING THE AVERAGE AND MEDIAN NUMBER OF REQUIRED CHANNELS FOR $q = 4$ AND VARIOUS CHOICES OF n, t_s, t_d AND t_i.

n	t_s	t_d	t_i	Average	Median
20	1	1	1	489	390
60	1	1	1	310	280
100	1	1	1	288	263
200	1	1	1	274	252
100	2	1	1	3940	3506
100	1	2	1	1310	1166
100	1	1	2	1163	1059
100	1	2	2	5243	4685
100	0	0	1	7	6
100	0	1	1	21	20
100	0	0	2	32	29
100	0	0	3	133	118

Based on Table III, we can make the following observations which also seem plausible by Equation (13) (and careful analysis of the algorithm):

- The number of required channels decreases when the length n increases.
- The substitution errors are the most difficult ones for the algorithm to handle.
- The algorithm works surprisingly well when no substitution errors occur.

ACKNOWLEDGEMENTS

We thank the reviewers for their helpful comments for improving the readability of the paper.

REFERENCES

[1] V. Junnila, T. Laihonen, and T. Lehtilä, "Levenshtein's reconstruction problem with different error patterns," in *2023 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2023, pp. 1300–1305.
[2] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Trans. Inform. Theory*, vol. 47, no. 1, pp. 2–22, 2001.
[3] E. Yaakobi and J. Bruck, "On the uncertainty of information retrieval in associative memories," *IEEE Trans. Inform. Theory*, vol. 65, no. 4, pp. 2155–2165, 2018.
[4] M. Abu-Sini and E. Yaakobi, "On Levenshtein's reconstruction problem under insertions, deletions, and substitutions," *IEEE Trans. Inform. Theory*, vol. 67, no. 11, pp. 7132–7158, 2021.
[5] V. L. P. Pham, K. Goyal, and H. M. Kiah, "Sequence reconstruction problem for deletion channels: A complete asymptotic solution," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 992–997.
[6] R. Gabrys and E. Yaakobi, "Sequence reconstruction over the deletion channel," *IEEE Trans. Inform. Theory*, vol. 64, no. 4, pp. 2924–2931, 2018.
[7] M. Abu-Sini and E. Yaakobi, "On list decoding of insertions and deletions under the reconstruction model," in *Proceedings of 2021 IEEE International Symposium on Information Theory*, 2021, pp. 1706–1711.
[8] V. Junnila, T. Laihonen, and T. Lehtilä, "The Levenshtein's sequence reconstruction problem and the length of the list," *IEEE Trans. Inform. Theory*, 2023.
[9] ——, "On Levenshtein's channel and list size in information retrieval," *IEEE Trans. Inform. Theory*, vol. 67, no. 6, pp. 3322–3341, 2020.
[10] M. Horovitz and E. Yaakobi, "Reconstruction of sequences over non-identical channels," *IEEE Trans. Inform. Theory*, vol. 65, no. 2, pp. 1267–1286, 2018.
[11] J. Chrisnata, H. M. Kiah, and E. Yaakobi, "Correcting deletions with multiple reads," *IEEE Trans. Inform. Theory*, vol. 68, no. 11, pp. 7141–7158, 2022.

[12] Y. M. Chee, R. Gabrys, A. Vardy, V. K. Vu, and E. Yaakobi, "Reconstruction from deletions in racetrack memories," in *2018 IEEE Information Theory Workshop (ITW)*. IEEE, 2018, pp. 1–5.

[13] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 2, pp. 637–649, 2016.

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

[15] 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. Edit.*, vol. 54, no. 8, pp. 2552–2555, 2015.

[16] S. H. T. Yazdi, H. M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, "DNA-based storage: Trends and methods," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 1, no. 3, pp. 230–248, 2015.

[17] O. Sabary, H. M. Kiah, P. H. Siegel, and E. Yaakobi, "Survey for a decade of coding for DNA storage," *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, 2024.

[18] R. Heckel, G. Mikutis, and R. N. Grass, "A characterization of the DNA data storage channel," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

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

[20] M. Abu-Sini and E. Yaakobi, "On the intersection of multiple insertion (or deletion) balls and its application to list decoding under the reconstruction model," *IEEE Trans. Inform. Theory*, 2023.

[21] A. Abbasian, M. Mirmohseni, and M. N. Kenari, "On the size of error ball in dna storage channels," *arXiv preprint arXiv:2410.15290*, 2024.

[22] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," in *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, 2004, pp. 910–918.

[23] M. Cheraghchi, R. Gabrys, O. Milenkovic, and J. Ribeiro, "Coded trace reconstruction," *IEEE Trans. Inform. Theory*, vol. 66, no. 10, pp. 6084–6103, 2020.

[24] K. Viswanathan and R. Swaminathan, "Improved string reconstruction over insertion-deletion channels," in *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, 2008, pp. 399–408.

[25] X. Chen, A. De, C. H. Lee, R. A. Servedio, and S. Sinha, "Near-optimal average-case approximate trace reconstruction from few traces," in *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2022, pp. 779–821.

[26] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[27] O. Levin, "Discrete mathematics: An open introduction," 2021.

[28] P. Flajolet, D. Gardy, and L. Thimonier, "Birthday paradox, coupon collectors, caching algorithms and self-organizing search," *Discrete Applied Mathematics*, vol. 39, no. 3, pp. 207–229, 1992.

[29] R. M. Young, "75.9 Euler's constant," *The Mathematical Gazette*, vol. 75, no. 472, pp. 187–190, 1991.

Ville Junnila was born in Turku, Finland in 1981. He received the M.Sc. and Ph.D. degrees in mathematics from the University of Turku, Finland, in 2007 and 2011, respectively.

From 2007 to 2011, he was a Doctoral Student at the Department of Mathematics and Statistics in the University of Turku, Finland. From 2011 to 2014, he was a Postdoctoral Researcher on a grant. In 2014, he joined the faculty of the Department of Mathematics and Statistics, University of Turku, where he first worked as a Postdoctoral Researcher and, since 2020, as a University Lecturer. His research interests include combinatorial coding and graph theory as well as related areas of discrete mathematics. He has over 30 journal articles in these topics.

Tero Laihonen received the M.Sc. and Ph.D. degrees in mathematics from the University of Turku, Turku, Finland, in 1995 and 1998, respectively. He was a Postdoctoral Researcher in 1999–2002 and an Academy Research Fellow in 2003–2008 at the Academy of Finland. He joined the faculty of the Department of Mathematics and Statistics at the University of Turku in 2008 where he is currently a Professor in discrete mathematics and theoretical computer science. His research interests include coding theory and graph theory with applications to DNA data storage, information retrieval and sensor networks.

Tuomo Lehtilä received his M.Sc. and Ph.D degrees in mathematics from the University of Turku, Turku, Finland, in 2016 and 2020, respectively. He was a Postdoctoral Researcher in 2021–2022 at LIRIS, Université Claude Bernard Lyon 1, Lyon, France, in University of Helsinki from 2023 to 2024 and a Postdoctoral Researcher in University of Turku during 2022–2023 and 2024 onward. Currently, he is a Postdoctoral Researcher at the Department of Mathematics and Statistics, University of Turku, Turku, Finland. His current research interests include coding theory, graph theory and related areas of discrete mathematics.