Theoretical Computer Science 1045 (2025) 115279

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Levenshtein's sequence reconstruction problem and results for larger alphabet sizes ☆

Ville Junnila, Tero Laihonen, Tuomo Lehtilä,*

Department of Mathematics and Statistics, University of Turku, FI-20014 Turku, Finland

ARTICLE INFO

Section Editor: Lila Kari

Handling Editor: Enrico Formenti

In honour of the 60th birthday of Jarkko Kari

Keywords:
Information retrieval
DNA-memory
Levenshtein's sequence reconstruction
Substitution errors
Erasure errors
Deletion errors
Insertion errors

ABSTRACT

The problem of storing large amounts of information safely for a long period of time has become essential. One of the most promising new data storage mediums are the polymer-based data storage systems, like the DNA-storage system. These storage systems are highly durable and they consume very little energy to store the data. When information is retrieved from a storage, however, several different types of errors may occur in the process. It is known that the Levenshtein's sequence reconstruction framework is well-suited to overcome such errors and to retrieve the original information. Many of the previous results regarding Levenshtein's sequence reconstruction method are so far given only for the binary alphabet. However, larger alphabets are natural for the polymer-based data storage. For example, the quaternary alphabet is suitable for DNA-storage due to the four amino-acids in DNA. The results for larger alphabets often require, as we will see in this work, different and more complicated techniques compared to the binary case. Moreover, we show that an increase in the alphabet size makes some error types behave rather surprisingly.

1. Introduction

The *Levenshtein's sequence reconstruction problem* was introduced in [22]. The setup of the problem is as follows. We have a subset C of words of length n where the alphabet has size q. A word $\mathbf{x}$ of C is transmitted through N channels and in each of them some errors occur, for example, substitution, deletion or insertion errors (these error types are defined below) to the word $\mathbf{x}$. With the aid of the (erroneous) output words $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$ from the channels, we try to determine, i.e., reconstruct, the transmitted word $\mathbf{x}$. Sometimes, when we cannot determine $\mathbf{x}$ uniquely, we wish to have as small list as possible of candidates for $\mathbf{x}$, which is naturally included in the list. Let us look at a small example. Suppose that C consists of all the binary (i.e., $q = 2$) words of length 3 and we transmit $\mathbf{x} = 000$ through two channels (i.e., $N = 2$). In each channel it is assumed that at most one substitution error occurs, that is, in one coordinate the symbol 0 can change to 1 or vice versa. Suppose we obtain the output words $\mathbf{y}_1 = 100$ and $\mathbf{y}_2 = 010$ from the channels and we denote the set of output words by $Y = \{\mathbf{y}_1, \mathbf{y}_2\}$. Now we consider all the possible words in C, denote this set by $T(Y)$, that could have been sent when we receive the above set Y, keeping in mind, that at most one substitution error can occur in the channels (the set $T(Y)$ is given more precisely in (2) below). The words that could have been transmitted are 000 and 110. Hence, $T(Y) = \{000, 110\}$ and the length of the list is equal to two. Notice that $\mathbf{x} \in T(Y)$. If we have more channels, say $N = 3$, and, for example, the set of output words is $Y = \{100, 010, 001\}$, then we can determine the transmitted word unambiguously since now

☆ The authors were funded in part by the Research Council of Finland grants 338797 and 358718. Some of the results in this article were presented without proofs in ISIT2023 [18].

* Corresponding author.
E-mail addresses: viljun@utu.fi (V. Junnila), terolai@utu.fi (T. Laihonen), tualeh@utu.fi (T. Lehtilä).

https://doi.org/10.1016/j.tcs.2025.115279
Received 2 September 2024; Received in revised form 14 April 2025; Accepted 24 April 2025

Available online 29 April 2025
0304-3975/© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

$T(Y) = \{\mathbf{x}\}$. In fact, it follows by Theorem 1 that if we send any word $\mathbf{x}$ of C, then with the aid of three distinct output words, we can always determine $\mathbf{x}$ uniquely.

The topic of Levenshtein's sequence reconstruction problem has been widely studied during recent years [1–3,12,13,16,17,19,25,27]. Levenshtein's original motivation came from molecular biology and chemistry, where adding redundancy was not feasible. Levenshtein's problem has recently become again significant due to advanced memory storage technologies such as associative memories [27], racetrack memories [7] and, especially, polymer-based memories like DNA-memories [1], where the information is stored in the DNA molecules. The DNA-storage process consists of three steps, namely, *synthesizing* (writing the information into DNA), *storing* and *sequencing* (retrieving the information). The first step, synthesis, produces artificial DNA molecules called *strands* to encode the user's information units. Due to technical reasons, this phase produces several noisy strands of the encoded data. In the second phase, the DNA strands are stored in an unordered manner in a storage container, where some molecules might be lost due to decay. The final sequencing step involves obtaining numerous erroneous copies of a stored synthesized strand. From these erroneous copies we should determine the original information in the strand. Synthesizing and sequencing DNA cause substitution, insertion and deletion errors to the information (these error types are discussed in more details below). This paper concentrates on the last phase of the process, that is, the information retrieval part. As multiple erroneous copies (corresponding to the output words in Y discussed above) of the stored information unit $\mathbf{x}$ are obtained, the Levenshtein's model is very suitable for this problem where we wish to determine (reconstruct) $\mathbf{x}$ using Y. Another interesting property which we obtain from DNA-applications and polymer-based memory systems in general, is the emphasis on q-ary ($q > 2$) information over the binary (see [6,11,14,28] for more information about DNA-memories).

Let us next consider some notations. We will denote the set $\{1, 2, \ldots, n\}$ by $[1, n]$ and by $\mathbb{Z}_q^n$ the set of q-ary words of length n (over the alphabet $\mathbb{Z}_q$). The set $\mathbb{Z}_q^n$ is often called the *q-ary n-dimensional Hamming space*. The *support* of a word $\mathbf{x} = x_1 \cdots x_n \in \mathbb{Z}_q^n$ is defined by $\mathrm{supp}(\mathbf{x}) = \{i \mid x_i \neq 0\}$, the *weight* of $\mathbf{x}$ by $w(\mathbf{x}) = |\mathrm{supp}(\mathbf{x})|$ and the *Hamming distance* between $\mathbf{x}$ and $\mathbf{y}$ by $d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} - \mathbf{y})$. For the *Hamming balls* we use notation $B_t(\mathbf{x}) = \{\mathbf{y} \in \mathbb{Z}_q^n \mid d(\mathbf{x}, \mathbf{y}) \leq t\}$ and $|B_t(\mathbf{x})| = V_q(n, t) = \sum_{i=0}^{t} (q-1)^i \binom{n}{i}$. A *code* C is a nonempty subset of $\mathbb{Z}_q^n$ and it has *minimum distance*

$$d_{\min}(C) = \min_{\mathbf{c}_1, \mathbf{c}_2 \in C, \mathbf{c}_1 \neq \mathbf{c}_2} d(\mathbf{c}_1, \mathbf{c}_2).$$

Furthermore, C is *e-error-correcting* if $d_{\min} \geq 2e + 1$. Finally, we denote the *zero-word* $00 \cdots 0$ with $\mathbf{0}$. The *value support* of word $\mathbf{x}$ is defined by $\mathrm{vsupp}(\mathbf{x}) = \{(i, x_i) \mid x_i \neq 0\}$. In particular, we have

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x}) + w(\mathbf{y}) - |\mathrm{supp}(\mathbf{x}) \cap \mathrm{supp}(\mathbf{y})| - |\mathrm{vsupp}(\mathbf{x}) \cap \mathrm{vsupp}(\mathbf{y})|. \tag{1}$$

This implies that $d(\mathbf{x}, \mathbf{y}) \geq w(\mathbf{x}) + w(\mathbf{y}) - 2|\mathrm{supp}(\mathbf{x}) \cap \mathrm{supp}(\mathbf{y})|$.

Let us discuss the following three types of errors that are particularly relevant for DNA-memories [15]. In a *substitution error* a symbol in some coordinate position is substituted with another symbol of the alphabet, in an *insertion error* a new symbol of the alphabet is inserted to (any position) in the original word leading to a word of length $n + 1$ and in a *deletion error* a symbol is deleted from (any position of) the original word leading to a word of length $n - 1$. For example, if $\mathbf{x} = 014 \in \mathbb{Z}_5^3$, then the words 0104 and 3014 are obtained using one insertion error and the words 14 and 04 are obtained using one deletion to $\mathbf{x}$. In this paper, we also consider another usual error type discussed in coding theory, namely, the *erasure errors*. In an *erasure error*, we replace x_i, the ith symbol of $\mathbf{x}$, with the symbol $*$ representing a coordinate in an output word where we cannot read the symbol. For example, if $\mathbf{x} = 014$, then the words $*14$ and $0*4$ are obtained by one erasure error.

Depending on the error type, we may need also other types of balls than just the typical Hamming balls which are suitable for substitution errors. By $B_{t_e, t_s}^{e,s}(\mathbf{x})$ we denote the ball containing all words which can be obtained from $\mathbf{x}$ with at most t_e erasures and at most t_s substitutions. By $B_{t_d}^{d}(\mathbf{x})$ we denote the set (ball) of words which can be obtained with at most t_d deletions from $\mathbf{x}$ and by $B_{t_e}^{e}(\mathbf{x})$ we denote the set of words which can be obtained with at most t_e erasures from $\mathbf{x}$.

Now, we define the sequence reconstruction problem more rigorously. 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 substitution errors (or, depending on the case, insertion, deletion or erasure errors) may occur. Also we assume that the number of each type of error is limited by some constant t. Furthermore, we obtain a set $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ of output words from the N channels. Depending on the situation, we may assume that each channel gives a different output word, that is, Y is a *set* of output words, or some output words can be the same, that is, Y is a *multiset* (in this paper, apart from Section 6, we consider Y as a set). Based on the (multi)set of output words Y, we try to deduce the transmitted word $\mathbf{x}$. However, sometimes we cannot do it and instead we have a list of possible transmitted words $T(Y)$ such that $\mathbf{x} \in T(Y)$. The maximum size of this list is denoted by $\mathcal{L}$. Sometimes we use the notation $T_{\mathcal{D}}(Y)$ when we use a specified decoder $\mathcal{D}$ (as in Section 5). We have illustrated the channel model in Fig. 1.

When we have an e-error-correcting code C, only substitution errors occur, and at most $t = e + \ell$ errors occur in a channel, then

$$T(Y) = \bigcap_{\mathbf{y} \in Y} B_t(\mathbf{y}) \cap C. \tag{2}$$

In this setup, the parameter $\mathcal{L}$, i.e., the maximum size of $T(Y)$, has been studied in [17,19,22,27]. For the case $\mathcal{L} = 1$ see Theorem 1 below. However, when combinations of different error types are studied, much less is known [1–3] and even when only deletion or insertion errors occur we have many open problems depending on the code C and the list size $\mathcal{L}$ [1–3,12,13,23]. Decoding algorithms for substitution errors have been studied in [1,22,27] and for deletion and insertion errors in [1,12,23].

Similar problem has also been considered (sometimes under the name *trace reconstruction*), when each error has an independent probability to occur, that is, the maximum number of errors in a channel is not fixed, in for example [5,9,10,26].

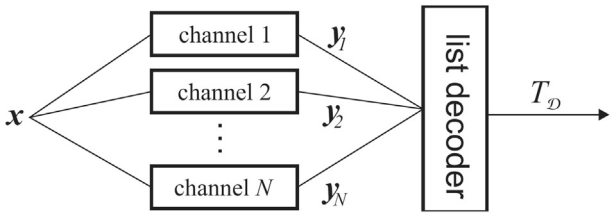

Fig. 1. The Levenshtein's sequence reconstruction.

Next, we introduce the Levenshtein's result which gives the exact required number of channels N to have $\mathcal{L} = 1$ for a q-ary Hamming space. As is usual, when $j > i$ or $j < 0$, we have $\binom{i}{j} = 0$.

Theorem 1 *([22]). Let $N \geq N_{t,e}$ and $C \subseteq \mathbb{Z}_q^n$ be an e-error-correcting code. Then $\mathcal{L} = 1$ if*

$$N_{t,e} = \sum_{i=0}^{\ell-1} \binom{n-2e-1}{i} (q-1)^i \sum_{k=e+i-(\ell-1)}^{t-i} \sum_{j=e+i-(\ell-1)}^{t-i} \binom{2e+1}{k} \binom{2e+1-k}{j} (q-2)^{2e+1-k-j} + 1.$$

For $e = 0$ there exists a simplified version of the bound above

$$N_{t,0} = q \sum_{i=0}^{t-1} \binom{n-1}{i} (q-1)^i + 1. \tag{3}$$

In particular, notice that if $N \geq N_{t,e}$, then $T(Y) = \{\mathbf{x}\}$ for any transmitted $\mathbf{x}$ and corresponding sets of output words Y. Further note, that when $C = \mathbb{Z}_q^n$, we have $e = 0$.

In this article, we concentrate on generalizing the results on the Levenshtein's reconstruction problem in the binary case to a more general q-ary environment. Especially the case with $q = 4$ is relevant for DNA-memories, since the information is encoded in them using the four different types of nucleotides [1], although even larger alphabet sizes are of interest for polymer-based memory systems [4,21].

The structure of this paper is as follows. In Section 2, we consider channels with erasure errors and determine the list size $\mathcal{L}$ (including $\mathcal{L} = 1$) with respect to the number N of channels. Notice that the introduction of erasure errors to a word of $\mathbb{Z}_q^n$ can be interpreted as changing the underlying q-ary alphabet to one with $q + 1$ symbols (including the erasure symbol $*$). In the next section, we allow a channel to have two types of errors simultaneously, namely, substitution errors and erasure errors. We provide the exact number of channels required in q-ary case to reconstruct the transmitted word uniquely (that is, $\mathcal{L} = 1$). In Section 4, we consider only substitution errors and determine the *exact* number of channels to guarantee that the list size $\mathcal{L}$ of possible transmitted words is always a *constant* with respect to the length n. This result is a q-ary generalization of a binary result introduced in [17]. We note that the results in this section also have independent theoretical interest as we give the exact size of the intersection of some q-ary Hamming balls. Previously, in [1], an efficient majority algorithm has been introduced for decoding the transmitted codeword in the binary case. We generalize the previous algorithm for larger q in Section 5. We also consider a new *list*-decoding algorithm for determining $T(Y)$. Here we restrict ourselves to the binary alphabet since the algorithm is based on some rather complicated results, which are known only in the binary case. In Section 6, we consider the likelihood of $|T(Y)| = 1$ for large q when the error type in a channel is, in turn, insertion, substitution, deletion or erasure error. In that section, we consider Y as a multiset while everywhere else in the paper Y is a set. We observe that for large q the insertion, substitution and deletion errors behave in different ways. In particular, the result for insertion error is rather surprising. Finally, we conclude in Section 7.

2. Erasures occurring in the channels

In this section, we consider erasure errors. This may also be understood as replacing a q-ary Hamming space by a $(q+1)$-ary Hamming space and by considering substitute errors where some symbols are substituted by q. From this perspective, we consider the symbol q (or $*$) to belong to the support. Similarly, we consider each symbol q to increase the weight of the word by one. Furthermore, we define the Hamming distance of two words $\mathbf{w}, \mathbf{z}$ which may contain erasures from this perspective, that is, when calculating the distance, if $w_i = q$ and $z_i \neq q$, then this difference increases the distance of $\mathbf{w}$ and $\mathbf{z}$ by one. Furthermore, observe that the size of t-radius erasure ball in $\mathbb{Z}_q^n$ is $V_2(n,t)$. In other words, when at most t erasures may occur in a channel, the number of channels is at most $V_2(n,t)$.

Let us denote by $A_q(n,d)$ the cardinality of a maximum size code $C \subseteq \mathbb{Z}_q^n$ with minimum distance d. Note that when $d > n$, we trivially have $A_q(n,d) = 1$. In general, there is no closed formula for $A_q(n,d)$, but numerous bounds and exact values are known for it (see [24]).

Theorem 2. *Assume that $n \geq t$ and at most t erasures may occur in a channel.*

(i) Let $C = \mathbb{Z}_q^n$. We have the maximum list size $\mathcal{L} = q^a$ if the number of channels N satisfies

$$V_2(n-a-1,t-a-1)+1 \le N \le V_2(n-a,t-a)$$

for some integer $a \in [0,t-1]$. In particular, we get $\mathcal{L} = 1$ when $N \ge V_2(n-1,t-1)+1$.

(ii) Let $C \subseteq \mathbb{Z}_q^n$ have minimum distance d. We have $\mathcal{L} = A_q(a,d)$ if

$$V_2(n-a-1,t-a-1)+1 \le N \le V_2(n-a,t-a)$$

for some integer $a \in [1,t-1]$.

Proof. Observe that if we have output words $\mathbf{y}_i \in Y$ $(i = 1,\dots,N)$ such that at least one of them does not have an erasure in the ith coordinate, then we can deduce what that symbol was in the transmitted word $\mathbf{x}$.

(i) Let us have $V_2(n-a-1,t-a-1)+1 \le N \le V_2(n-a,t-a)$ for some integer $a \in [0,t-1]$.

Consider an output word set such that each $\mathbf{y} \in Y$ has erasures in each coordinate of $[1,a]$. There are $V_2(n-a,t-a)$ possibilities for such words when at most t erasures occur. Clearly there are q^a possibilities for possible transmitted words in this case as $C = \mathbb{Z}_q^n$. Thus, $\mathcal{L} \ge q^a$. Let us then show that $\mathcal{L} \le q^a$.

Suppose to the contrary that $\mathcal{L} > q^a$. This is possible only if there exists such an output word set Y that, for some set S of $a+1$ coordinates, each $\mathbf{y} \in Y$ has the symbol $*$ in all of these coordinates. Now each output word has $a+1$ erasures in the coordinate set S and at most $t-a-1$ erasures in coordinates $[1,n] \setminus S$. Thus, $|Y| \le V_2(n-a-1,t-a-1) < N$. Hence, the claim follows.

(ii) This case goes analogously. Let us have $V_2(n-a-1,t-a-1)+1 \le N \le V_2(n-a,t-a)$ for some integer $a \in [1,t-1]$. Consider an output word set such that each $\mathbf{y} \in Y$ has erasures in the symbols within coordinate positions $[1,a]$. There are $V_2(n-a,t-a)$ possibilities for such words when at most $t \ge a$ erasures occur. Moreover, there are at most $A_q(a,d)$ possible transmitted codewords as the underlying code has minimum distance d. Thus, $\mathcal{L} \ge A_q(a,d)$. Let us then show that $\mathcal{L} \le A_q(a,d)$.

Suppose to the contrary that $\mathcal{L} > A_q(a,d)$. This is possible only if there exists such an output word set Y that, for some set S of at least $a+1$ coordinates, each $\mathbf{y} \in Y$ has the symbol $*$ in all of these coordinates. Now each output word has $a+1$ erasures in coordinate set S and at most $t-a-1$ erasures in coordinates $[1,n] \setminus S$. Thus, $|Y| \le V_2(n-a-1,t-a-1) < N$. Hence, the claim follows. □

In particular, when we compare Case (i) of the previous theorem for erasures with $\mathcal{L} = 1$ to Theorem 1 with code $C = \mathbb{Z}_q^n$ (and hence $e = 0$), we observe that unlike in the case of *substitution*, the number of required channels does not increase as q increases. Furthermore, when we consider a binary case $q = 2$, we notice (see (3)) that substitution errors require approximately twice as many channels as erasure errors, that is, $2V_2(n-1,t-1)+1$ channels.

3. Substitutions and erasures occurring in the channels

In this section, we consider the case where at most t_s substitutions and t_e erasures occur in any channel. Note that we do not define the order in which these different types of errors may occur as it will not affect in our results. Moreover, when the underlying code C is e-error-correcting, we denote $t_s = e + \ell$. We denote

$$V_q(n,a_1,a_2) = \sum_{i=0}^{a_1} \binom{n}{i} \sum_{j=0}^{a_2} \binom{n-i}{j} (q-1)^j.$$

Observe that value $V_q(n,a_1,a_2)$ denotes in how many different ways we may assign at most a_1 erasures and a_2 substitutions to a word in a q-ary Hamming space, that is, $|B^{e,s}_{t_e,t_s}(\mathbf{x})| = V_q(n,t_e,t_s)$. We denote by $d_e(\mathbf{x},\mathbf{y}) = d(\mathbf{x},\mathbf{y}) - |\{i \mid x_i = * \text{ xor } y_i = *\}|$, that is, d_e denotes the number of coordinates with symbols other than $*$ in which $\mathbf{x}$ and $\mathbf{y}$ differ.

First, we recall a result from Levenshtein for intersections of substitution balls. Note that Levenshtein required that $d(\mathbf{x},\mathbf{x}') \le 2t$. However, when this is not true, the intersections of balls are simply empty and the statement still holds.

Lemma 3 *(Corollary 1 of [22]). Let $\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4 \in \mathbb{Z}_q^n$ and $d(\mathbf{x}_1,\mathbf{x}_2) \le d(\mathbf{x}_3,\mathbf{x}_4)$. We have $|B_t(\mathbf{x}_1) \cap B_t(\mathbf{x}_2)| \ge |B_t(\mathbf{x}_3) \cap B_t(\mathbf{x}_4)|$.*

In the following lemma, we generalize Levenshtein's result for combinations of substitution and erasure errors.

Lemma 4. *Let $\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4 \in \mathbb{Z}_q^n$ be such that $d(\mathbf{x}_1,\mathbf{x}_2) \le d(\mathbf{x}_3,\mathbf{x}_4)$. We have*

$$|B^{e,s}_{t_e,t_s}(\mathbf{x}_1) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}_2)| \ge |B^{e,s}_{t_e,t_s}(\mathbf{x}_3) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}_4)|.$$

Proof. Notice that we are only interested in the cardinalities of intersections of balls. Moreover, the balls $B^{e,s}_{t_e,t_s}(\mathbf{x}_1)$ or $B^{e,s}_{t_e,t_s}(\mathbf{x}_2)$ do not interact with balls $B^{e,s}_{t_e,t_s}(\mathbf{x}_3)$ or $B^{e,s}_{t_e,t_s}(\mathbf{x}_4)$. Hence, we may assume without loss of generality that $\mathbf{x}_1 = \mathbf{0} = \mathbf{x}_3$. Observe that now value $|B^{e,s}_{t_e,t_s}(\mathbf{x}_1) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}_2)|$ depends only on $w(\mathbf{x}_2)$ due to the symmetries of Hamming space and the same holds for $\mathbf{x}_3$ and $\mathbf{x}_4$. Hence, if $d(\mathbf{x}_1,\mathbf{x}_2) = d = d(\mathbf{x}_3,\mathbf{x}_4)$, then the claim follows. We prove the lemma for $d(\mathbf{x}_1,\mathbf{x}_2) = d$ and $d(\mathbf{x}_3,\mathbf{x}_4) = d+1$. The claim follows by applying this iteratively. Hence, we assume without loss of generality that $\mathrm{supp}(\mathbf{x}_2) = \{1,\dots,d\}$ and $\mathrm{supp}(\mathbf{x}_4) = \{1,\dots,d+1\}$.

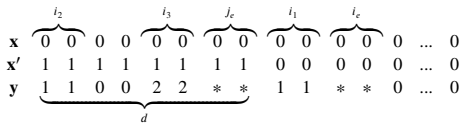

Fig. 2. Transmitted word candidates $\mathbf{x}$ and $\mathbf{x}'$ at distance d from each other together with output word $\mathbf{y}$ and variables i_1, i_2, i_3, i_e and j_e as in the proof of Theorem 5. Note that due to the symmetries of Hamming space, we may assume without loss of generality that $\mathbf{x} = \mathbf{0}$ and $\mathbf{x}'$ contains only 0's and 1's.

Observe that if $\mathbf{y} \in B^{e,s}_{t_e,t_s}(\mathbf{x}_1) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}_2)$, then the erasures must have occurred in the same coordinates in $\mathbf{x}_1$ and $\mathbf{x}_2$. Let us denote by $D(\mathbf{e},\mathbf{w})$, where $\mathbf{e} \in \mathbb{Z}_2^n$ and $\mathbf{w} \in \mathbb{Z}_q^n$, a word $\mathbf{w}' = D(\mathbf{e},\mathbf{w}) \in \mathbb{Z}_q^{n-w(\mathbf{e})}$ obtained from $\mathbf{w}$ by deleting the symbols on the coordinates defined by $\mathrm{supp}(\mathbf{e})$. We note that

$$|B^{e,s}_{t_e,t_s}(\mathbf{x}_1) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}_2)| = \sum_{w(\mathbf{e})\le t_e, \mathbf{e}\in\mathbb{Z}_2^n} |B_{t_s}(D(\mathbf{e},\mathbf{x}_1)) \cap B_{t_s}(D(\mathbf{e},\mathbf{x}_2))|$$

and

$$|B^{e,s}_{t_e,t_s}(\mathbf{x}_3) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}_4)| = \sum_{w(\mathbf{e})\le t_e, \mathbf{e}\in\mathbb{Z}_2^n} |B_{t_s}(D(\mathbf{e},\mathbf{x}_3)) \cap B_{t_s}(D(\mathbf{e},\mathbf{x}_4))|.$$

Notice that we have $d(D(\mathbf{e},\mathbf{x}_1), D(\mathbf{e},\mathbf{x}_2)) \le d(D(\mathbf{e},\mathbf{x}_3), D(\mathbf{e},\mathbf{x}_4))$ for each $\mathbf{e} \in \mathbb{Z}_2^n$. Hence, by Lemma 3, we have

$$|B_{t_s}(D(\mathbf{e},\mathbf{x}_1)) \cap B_{t_s}(D(\mathbf{e},\mathbf{x}_2))| \ge |B_{t_s}(D(\mathbf{e},\mathbf{x}_3)) \cap B_{t_s}(D(\mathbf{e},\mathbf{x}_4))|.$$

Therefore, $|B^{e,s}_{t_e,t_s}(\mathbf{x}_1) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}_2)| \ge |B^{e,s}_{t_e,t_s}(\mathbf{x}_3) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}_4)|$, as claimed. □

Theorem 5. *Let $n \ge t_e + t_s$ and C be a code with minimum distance d. We have $\mathcal{L} = 1$ if*

$$\begin{aligned} N \ge N' = &\sum_{i_e=0}^{t_e}\sum_{j_e=0}^{t_e-i_e}\binom{n-d}{i_e}\binom{d}{j_e}\sum_{i_1=0}^{t_s-\lceil (d-j_e)/2\rceil}\binom{n-d-i_e}{i_1}(q-1)^{i_1} \\ &\cdot \sum_{i_2=d+i_1-j_e-t_s}^{t_s-i_1}\binom{d-j_e}{i_2}\sum_{i_3=0}^{t_s-i_1-i_2}\binom{d-j_e-i_2}{i_3}(q-2)^{i_3}+1 \\ =&V_q(n,t_e,t_s-d)+\sum_{i_e=0}^{t_e}\sum_{j_e=0}^{t_e-i_e}\binom{n-d}{i_e}\binom{d}{j_e}\sum_{i_1=0}^{t_s-\lceil (d-j_e)/2\rceil}\binom{n-d-i_e}{i_1}(q-1)^{i_1} \\ &\cdot \sum_{i_2=d+i_1-j_e-t_s}^{t_s-i_1}\binom{d-j_e}{i_2}\sum_{i_3=t_s+1-i_1-i_2-d}^{t_s-i_1-i_2}\binom{d-j_e-i_2}{i_3}(q-2)^{i_3}+1. \end{aligned}$$

If $N < N'$, then $\mathcal{L} \ge 2$.

Proof. Let $d(\mathbf{x},\mathbf{x}') = d' \ge d$ for some $\mathbf{x},\mathbf{x}' \in \mathbb{Z}_q^n$. We may assume, without loss of generality, that $\mathbf{x} = \mathbf{0}$. Thus, $w(\mathbf{x}') = d'$. Consider $B = B^{e,s}_{t_e,t_s}(\mathbf{x}) \cap B^{e,s}_{t_e,t_s}(\mathbf{x}')$. To maximize the size of B, we assume by Lemma 4 that $d' = d$. The number of channels N' which is required for $\mathcal{L} = 1$ equals $N' = |B| + 1$. Furthermore, if $N \le |B|$, then we have $\mathcal{L} \ge 2$ as we can have $\mathbf{x}$ and $\mathbf{x}'$ in $T(Y')$ for $Y' \subseteq B$. Hence, it is enough to show that $N' = |B| + 1$.

Consider next a word $\mathbf{y} \in B$. In the following discussions, we consider variables i_1, i_2, i_3, i_e and j_e together with words $\mathbf{x},\mathbf{x}'$ and $\mathbf{y}$. For an example illustration of these variables, see Fig. 2. Let us denote by j_e the number of erasures in $\mathbf{y}$ such that they occur within $\mathrm{supp}(\mathbf{x}')$, let $i_1 = |\mathrm{supp}(\mathbf{y}) \setminus (\mathrm{supp}(\mathbf{x}') \cup \{i \mid y_i = *\})|$ and $i_2 = |\mathrm{vsupp}(\mathbf{y}) \cap \mathrm{vsupp}(\mathbf{x}')|$. We have

$$d_e(\mathbf{y},\mathbf{x}') = (d - i_2 - j_e) + i_1 \le t_s. \tag{4}$$

Indeed, $w(\mathbf{x}') = d$ and each element in $\mathrm{vsupp}(\mathbf{x}')$ which is not contained in $\mathrm{vsupp}(\mathbf{y})$ or in which $\mathbf{y}$ does not have an erasure increases $d_e(\mathbf{y},\mathbf{x}')$ by one. This implies the term $d - i_2 - j_e$. Furthermore, each element in $\mathrm{supp}(\mathbf{y}) \setminus \mathrm{supp}(\mathbf{x}')$ which is not an erasure, increases $d_e(\mathbf{y},\mathbf{x}')$ by one.

As the total number of substitutions in $\mathbf{y}$ is at most t_s, we have $i_2 \le t_s - i_1$, since $d_e(\mathbf{x},\mathbf{y}) \le t_s$, and i_1 is an integer. By Equation (4), we have $i_1 \le t_s + i_2 - (d - j_e) \le 2t_s - i_1 - (d - j_e)$. Thus,

$$i_1 \le t_s - \lceil (d - j_e)/2 \rceil. \tag{5}$$

Moreover, using (4) we get

$$i_2 \ge d - j_e - t_s + i_1. \tag{6}$$

Finally, let us denote by i_e the number of erasures, in $\mathbf{y}$, outside of coordinates of $\mathrm{supp}(\mathbf{x}')$ and by i_3 the number of substitution errors, in $\mathbf{y}$, in $\mathrm{supp}(\mathbf{x}')$ such that the values of $\mathbf{y}$ and $\mathbf{x}'$ differ within those coordinates (hence there are $(q-2)^{i_3}$ ways to choose them). In other words, $i_3 = |\mathrm{supp}(\mathbf{x}') \cap \mathrm{supp}(\mathbf{y}) \cap \{i \mid y_i \neq * \text{ and } x'_i \neq y_i\}|$. Note that since the total number of substitutions in $\mathbf{y}$ is at most t_s, we have

$$0 \leq i_3 \leq t_s - i_1 - i_2. \tag{7}$$

Assume next that word $\mathbf{y}$ is such that above Inequalities (5), (6) and (7) hold for it. Together with lower bound $0 \leq i_1$, upper bound $i_2 \leq t_s - i_1$, $0 \leq i_e \leq t_e$ and $0 \leq j_e \leq t_e - i_e$ we show that then $\mathbf{y} \in B$. First of all, $\mathbf{y}$ is obtained with $i_1 + i_2 + i_3$ substitutions and $i_e + j_e$ erasures from $\mathbf{x}$. Since $i_1 + i_2 + i_3 \leq t_s$ and $i_e + j_e \leq t_e$, we have $\mathbf{y} \in B^{e,s}_{t_e,t_s}(\mathbf{x})$. Furthermore, we may obtain $\mathbf{y}$ from $\mathbf{x}'$ by first placing the $i_e + j_e \leq t_e$ erasure symbols $*$ to corresponding coordinates, then executing i_1 suitable substitutions to coordinates outside of $\mathrm{supp}(\mathbf{x}')$ and finally by making i_3 suitable substitutions to the coordinates in $\mathrm{supp}(\mathbf{x}')$. Since $i_1 + i_3 \leq t_s$, we have $\mathbf{y} \in B^{e,s}_{t_e,t_s}(\mathbf{x}')$ and thus, $\mathbf{y} \in B$ as claimed.

Therefore, we have

$$\begin{aligned}
|B| &= \sum_{i_e=0}^{t_e} \sum_{j_e=0}^{t_e-i_e} \binom{n-d}{i_e} \binom{d}{j_e} \sum_{i_1=0}^{t_s-\lceil (d-j_e)/2 \rceil} \binom{n-d-i_e}{i_1} (q-1)^{i_1} \\
&\quad \cdot \sum_{i_2=d+i_1-j_e-t_s}^{t_s-i_1} \binom{d-j_e}{i_2} \sum_{i_3=0}^{t_s-i_1-i_2} \binom{d-j_e-i_2}{i_3} (q-2)^{i_3} \\
&= V_q(n, t_e, t_s-d) + \sum_{i_e=0}^{t_e} \sum_{j_e=0}^{t_e-i_e} \binom{n-d}{i_e} \binom{d}{j_e} \sum_{i_1=0}^{t_s-\lceil (d-j_e)/2 \rceil} \binom{n-d-i_e}{i_1} (q-1)^{i_1} \\
&\quad \cdot \sum_{i_2=d+i_1-j_e-t_s}^{t_s-i_1} \binom{d-j_e}{i_2} \sum_{i_3=t_s+1-i_1-i_2-d}^{t_s-i_1-i_2} \binom{d-j_e-i_2}{i_3} (q-2)^{i_3}.
\end{aligned}$$

The latter equality above follows from the observations 1) that $\mathbf{y} \in B$, if $d_e(\mathbf{0}, \mathbf{y}) \leq t_s - d$ and at most t_e erasures occur to $\mathbf{y}$. Indeed, we may transform $\mathbf{x}'$ into $\mathbf{0}$ with d substitutions and then $\mathbf{0}$ into $\mathbf{y}$ with $t_s - d$ substitutions and at most t_e erasures. Hence, $B^{e,s}_{t_e,t_s-d}(\mathbf{0}) \subseteq B$. Furthermore, we observe 2) that the last sum term considers exactly the cases where at least $t_s - d + 1$ substitutions occur since in the last sum term we have $i_3 \geq t_s + 1 - i_1 - i_2 + d$ and hence, $i_1 + i_2 + i_3 \geq t_s - d + 1$. Hence, the sum does not consider the output words already considered in $B^{e,s}_{t_e,t_s-d}(\mathbf{0}) \subseteq B$. Since $N' = |B| + 1$, we have $\mathcal{L} = 1$. □

4. A tight bound for constant q-ary list size

In this section, we consider only *substitution* errors and our aim is to find out the exact number of channels which guarantees that the list size is always a *constant* with respect to the length n. We note that our results are also independently interesting from theoretical perspective as they give new information about the size of the intersection of q-ary Hamming balls. This perspective is briefly discussed at the end of the section. We first show that if we have too few channels, then there are e-error correcting codes in $\mathbb{Z}_q^n$ such that the list size grows when n increases. Indeed, assume that the number of channels satisfies $N \leq V_q(n, \ell - 1)$ and that all the output words in $Y \subseteq \mathbb{Z}_q^n$ are located inside the ball $B_{\ell-1}(\mathbf{0})$. Clearly, any word of weight $e+1$ in $\mathbb{Z}_q^n$ belongs to the intersection $\bigcap_{\mathbf{y} \in Y} B_t(\mathbf{y})$. If our e-error-correcting code is such that each codeword has weight $e+1$, then all the codewords are in the intersection. Such codes (usually called *q-ary constant-weight codes*) have been widely studied in the literature (see, for example, [8] and references therein). For our purposes (since we are only interested in the dependence on n and not on the largest possible constant-weight codes in $\mathbb{Z}_q^n$), it is enough to choose the code C_e^n which consists of the words with the following value supports

$$\mathrm{vsupp}(\mathbf{c}_j) = \{(j(e+1)+k, 1) \mid k = 1, 2, \ldots, e+1\}$$

where $j = 0, 1, \ldots, \lfloor n/(e+1) \rfloor$. (For example, if $n = 7$ and $e = 2$, then the code $C_2^7 = \{1110000, 0001110\}$). Since the minimum distance of C_e^n equals $2e+2$ by (1), we obtain the following result.

Theorem 6. *Let $t = e + \ell$. If $N \leq V_q(n, \ell - 1)$ then there exists an e-error-correcting code in $\mathbb{Z}_q^n$ such that*

$$\mathcal{L} \geq \lfloor n/(e+1) \rfloor.$$

In what follows, we show that if the number of channels is larger than $V_q(n, \ell - 1)$ (considered in Theorem 6), that is, $N \geq V_q(n, \ell - 1) + 1$, then, given *any* e-error-correcting code, the list size is immediately independent of n (when n is large enough). Moreover, we will show that the constant upper bound, which we obtain in Theorem 10 for the list size, is actually *optimal* as will be seen in Theorem 11.

It should be noticed that although the Lemmas 7 and 8 are rather similar to the binary case discussed in [17], our new techniques in Lemma 9 and Theorem 10 differ significantly from the ones in [17] as we consider the more complicated case of q-ary alphabet for $q > 2$.

In the following lemma, we show that if n and N are large enough, then for a set of coordinate positions $\overline{D}$ of size b and any codeword $\mathbf{c}$, there exists an output word $\mathbf{y}$ such that $\mathbf{y}$ differs from $\mathbf{c}$ in at least $\ell-1$ coordinate positions outside of $\overline{D}$.

Lemma 7. *Assume that $Y \subseteq \mathbb{Z}_q^n$, $|Y| = N \geq V_q(n, \ell-1)+1$, C is an e-error-correcting code and b is a positive integer. If $n \geq \ell - 2 + (\ell-1)^2 2^b (q-1)^{b-1}$, then for any codeword $\mathbf{c} \in T(Y)$ and for any set $\overline{D} \subseteq [1,n]$ with $|\overline{D}| = b$, there exists a word $\mathbf{y} \in Y$ such that*

$$|supp(\mathbf{c}-\mathbf{y}) \setminus \overline{D}| \geq \ell - 1.$$

Proof. Let $\overline{D} \subseteq [1,n]$ and $|\overline{D}| = b$ for some fixed b. Without loss of generality, we may assume that $\mathbf{c} = \mathbf{0}$. Suppose to the contrary that there does not exist a word $\mathbf{y} \in Y$ such that $|\mathrm{supp}(\mathbf{c}-\mathbf{y}) \setminus \overline{D}| = |\mathrm{supp}(\mathbf{y}) \setminus \overline{D}| \geq \ell - 1$, i.e., $|\mathrm{supp}(\mathbf{y}) \setminus \overline{D}| < \ell - 1$ for all $\mathbf{y} \in Y$. Note that since $b > 0$, we have $n \geq 2(\ell-2)$. This implies that the number of words in Y is at most

$$\begin{aligned}
&\sum_{j=0}^{\ell-2} \sum_{i=0}^{\min\{b,t-j\}} (q-1)^{i+j} \binom{b}{i} \binom{n-b}{j} \\
\leq &\sum_{j=0}^{\ell-2} \sum_{i=0}^{b} (q-1)^{i+j} \binom{b}{i} \binom{n-b}{j} \\
\leq &2^b (q-1)^b \sum_{j=0}^{\ell-2} (q-1)^j \binom{n-b}{j} \\
\leq &(\ell-1) 2^b (q-1)^{b+\ell-2} \binom{n}{\ell-2} \\
= &(\ell-1) \binom{n}{\ell-1} \frac{\ell-1}{n-\ell+2} 2^b (q-1)^{b+\ell-2} \\
\leq &(q-1)^{\ell-1} \binom{n}{\ell-1} \\
< &V_q(n, \ell-1),
\end{aligned}$$

when $n \geq \ell - 2 + (\ell-1)^2 2^b (q-1)^{b-1}$. This contradicts with the assumption that $N = |Y| \geq V_q(n,\ell-1)+1$. Thus, the claim follows. □

In the following lemma, we show that if n is large enough and $N \geq V_q(n,\ell-1)+1$, then the pairwise distances of codewords in T are rather small.

Lemma 8. *Let $n \geq \ell - 2 + (\ell-1)^2 2^{2t} (q-1)^{2t-1}$, C be an e-error-correcting code and $|Y| = N \geq V_q(n,\ell-1)+1$. Then we have $d(\mathbf{c}_1, \mathbf{c}_2) \leq 2e+2$ for any $\mathbf{c}_1, \mathbf{c}_2 \in T(Y)$.*

Proof. Let $\mathbf{c}_1$ and $\mathbf{c}_2$ be codewords in $T(Y)$. Without loss of generality, we may assume that $\mathbf{c}_1 = \mathbf{0}$. In order to show that $d(\mathbf{c}_1, \mathbf{c}_2) \leq 2e+2$, we suppose to the contrary that $d(\mathbf{c}_1, \mathbf{c}_2) \geq 2e+3$, i.e., $w(\mathbf{c}_2) \geq 2e+3$. Since $\mathbf{c}_1, \mathbf{c}_2 \in T(Y)$, we have $w(\mathbf{c}_2) = d(\mathbf{c}_1, \mathbf{c}_2) \leq 2t$. Hence, there exists a set $\overline{D} \subseteq [1,n]$ such that $|\overline{D}| = 2t$ and $\mathrm{supp}(\mathbf{c}_2) \subseteq \overline{D}$.

Since $n \geq \ell - 2 + (\ell-1)^2 2^{2t} (q-1)^{2t-1}$, by Lemma 7, there exists an output $\mathbf{y} \in Y$ such that $|\mathrm{supp}(\mathbf{y}) \setminus \mathrm{supp}(\mathbf{c}_2)| \geq \ell - 1$. Since $w(\mathbf{y}) = d(\mathbf{y}, \mathbf{c}_1) \leq t$, we have $|\mathrm{supp}(\mathbf{c}_2) \cap \mathrm{supp}(\mathbf{y})| \leq e+1$; indeed, if $|\mathrm{supp}(\mathbf{c}_2) \cap \mathrm{supp}(\mathbf{y})| \geq e+2$, then $w(\mathbf{y}) = |\mathrm{supp}(\mathbf{c}_2) \cap \mathrm{supp}(\mathbf{y})| + |\mathrm{supp}(\mathbf{y}) \setminus \mathrm{supp}(\mathbf{c}_2)| \geq (e+2) + (\ell-1) = t+1$ (a contradiction). This further implies that

$$\begin{aligned}
d(\mathbf{c}_2, \mathbf{y}) &\geq (w(\mathbf{c}_2) - |\mathrm{supp}(\mathbf{c}_2) \cap \mathrm{supp}(\mathbf{y})|) + \ell - 1 \\
&\geq (2e+3-(e+1)) + \ell - 1 = t+1.
\end{aligned}$$

This leads to a contradiction, and the claim follows. □

Lemma 9. *Let $\ell \geq 1, q \geq 3$, C be an e-error-correcting code and $|Y| = N \geq V_q(n,\ell-1)+1$. Moreover, let $\mathbf{w} \in \mathbb{Z}_q^n$ be such a word that $d(\mathbf{w}, \mathbf{c}) \leq e+1$ for each $\mathbf{c} \in T(Y)$. Then we have*

$$|T(Y)| \leq \ell(q-1)+1.$$

Proof. We may assume without loss of generality that $\mathbf{w} = \mathbf{0}$. Since $d(\mathbf{w}, \mathbf{c}) \leq e+1$ for each $\mathbf{c} \in T(Y)$, we have $w(\mathbf{c}) \leq e+1$. Thus, since C is an e-error-correcting code with minimum distance $2e+1$, if $|T(Y)| > 1$, then $w(\mathbf{c}) \in \{e, e+1\}$ for each $\mathbf{c} \in T(Y)$ and there is at most one codeword of weight e, denoted by $\mathbf{c}_0$. Since $N \geq V_q(n,\ell-1)+1$, there is at least one output word of weight at least ℓ. Let us say that for some $\mathbf{y} \in Y$ we have $w(\mathbf{y}) = \ell - 1 + a$ where $1 \leq a$. Observe that $a \leq 2e+2$ since $d(\mathbf{y}, \mathbf{c}) \leq t$ for each $\mathbf{c} \in T(Y)$.

Let us consider $\mathbf{c} \in T(Y)$ with $w(\mathbf{c}) = e+1$. Since $d(\mathbf{y}, \mathbf{c}) \leq t$ for each $\mathbf{c} \in T(Y)$ and, by (1), $d(\mathbf{y}, \mathbf{c}) = w(\mathbf{y}) + w(\mathbf{c}) - |\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| - |\mathrm{vsupp}(\mathbf{y}) \cap \mathrm{vsupp}(\mathbf{c})| = t + a - |\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| - |\mathrm{vsupp}(\mathbf{y}) \cap \mathrm{vsupp}(\mathbf{c})|$, we have $a \leq |\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| + |\mathrm{vsupp}(\mathbf{y}) \cap$

vsupp($\mathbf{c}$)|. Hence, we have $|\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| \geq \lceil \frac{a}{2} \rceil$. Moreover, since C is e-error-correcting, we have $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{c}')| = 0$ for each $\mathbf{c}' \in T(Y)$ other than $\mathbf{c}$. Thus, we have at most $1 + \frac{w(\mathbf{y})(q-1)}{\lceil \frac{a}{2} \rceil}$ codewords in $T(Y)$. Indeed, we have

$$|T(Y) \setminus \{\mathbf{c}_0\}| \cdot \lceil a/2 \rceil \leq \sum_{\mathbf{c} \in T(Y)} |\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| \leq w(\mathbf{y})(q-1).$$

Here the first inequality is due to $|\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| \geq \lceil \frac{a}{2} \rceil$ and the second one follows from $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{c}')| = 0$.

Notice that when $a = 1$, this gives the claimed value $\ell(q-1)+1$. Next we consider separately the cases where a is odd and even.

Let us first consider the case where $a \geq 1$ is odd and $\ell \geq 2$. In that case, we have

$$1 + \frac{w(\mathbf{y})(q-1)}{\lceil \frac{a}{2} \rceil} = 1 + \frac{2(\ell - 1 + a)(q-1)}{a+1} = 1 + 2 \cdot \frac{(\ell-2)+(a+1)}{a+1} \cdot (q-1) \leq 1 + \ell(q-1).$$

Let us then consider the case with even $a \geq 4$ and $\ell \geq 3$. We have

$$1 + \frac{w(\mathbf{y})(q-1)}{\lceil \frac{a}{2} \rceil} = 1 + \frac{2(\ell - 1 + a)(q-1)}{a} \leq 1 + \frac{(\ell+3)(q-1)}{2} \leq 1 + \ell(q-1).$$

Hence, we are left with the cases 1) $a = 2$ and $\ell \geq 1$, 2) $a \geq 4$ even and $\ell = 2$, and 3) $a \geq 3$ and $\ell = 1$. Let us next concentrate on Case 1) $a = 2$ and $\ell \geq 2$. Now, $w(\mathbf{y}) = \ell + 1$ and for $\mathbf{c} \in T(Y)$ with $w(\mathbf{c}) = e+1$, we have $2 \leq |\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| + |\mathrm{vsupp}(\mathbf{y}) \cap \mathrm{vsupp}(\mathbf{c})|$. Thus, $|\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| \geq 2$ or $|\mathrm{vsupp}(\mathbf{y}) \cap \mathrm{vsupp}(\mathbf{c})| \geq 1$. Since codewords in $T(Y)$ do not have overlapping value supports, there are at most $\ell + 1$ words in $T(Y)$ such that $|\mathrm{vsupp}(\mathbf{y}) \cap \mathrm{vsupp}(\mathbf{c})| \geq 1$. If the single word $\mathbf{c}_0 \in T(Y)$ with weight e exists, then it has $|\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c}_0)| \geq 1$ since $d(\mathbf{c}_0, \mathbf{y}) \leq t$. Let us denote by k the number of codewords $\mathbf{c} \in T(Y)$ with $|\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| \geq 2$ and by k' the number of codewords $\mathbf{c} \in T(Y)$ with $|\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| = 1$. Then, we have $\sum_{\mathbf{c} \in T(Y)} |\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| \geq k' + 2k$. Moreover, $k' \leq \ell + 2$ as at most $\ell + 1$ words may have intersecting value support with $\mathbf{y}$ and the word $\mathbf{c}_0$ may have its support intersecting $\mathbf{y}$ without having an intersecting value support. Furthermore, since none of these words have overlapping value supports, we have $\sum_{\mathbf{c} \in T(Y)} |\mathrm{supp}(\mathbf{y}) \cap \mathrm{supp}(\mathbf{c})| \leq (\ell+1)(q-1)$. Hence, we get $k' + 2k \leq (\ell+1)(q-1)$. To maximize $k' + k$, we may assume that k' is maximal, that is, $k' = \ell + 2$. Thus,

$$(\ell+2) + 2k = k' + 2k \leq (\ell+1)(q-1) \quad \Longleftrightarrow \quad k \leq (\ell+1)(q-1)/2 - \ell/2 - 1.$$

Hence, $k' + k \leq (\ell+1)(q-1)/2 + \ell/2 + 1 = \ell(q-1)/2 + (q-1)/2 + \ell/2 + 1$. For $\ell = 1$, this gives upper bound $\ell(q-1) + 3/2$ which implies $k' + k \leq 1 + \ell(q-1)$ as $k' + k$ is integer. Moreover, we notice that the value is at most $\ell(q-1)+1$ also for $\ell \geq 2$. Indeed, $(q-1)/2 + \ell/2 = (q-1+\ell)/2 \leq (q-1)\ell/2$ and hence, $k' + k \leq \ell(q-1)/2 + (q-1)\ell/2 + 1 \leq \ell(q-1) + 1$.

Let us next consider Case 2) $\ell = 2$ and $a \geq 4$ is even. We have $w(\mathbf{y}) = a + 1$. Recall that for each $\mathbf{c} \in T(Y)$ with weight $e+1$, we have $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| \geq \lceil a/2 \rceil = a/2$ and this lower bound is attained only when $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{y})| = a/2$. Indeed, if $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{y})| \leq a/2 - 1$, then $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| \geq a/2 + 1$. Let us denote by k the number of codewords of weight $e+1$ with $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| \geq a/2 + 1$ and by k' the number of codewords of weight $e+1$ with $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| = a/2$. Note that $|\mathrm{supp}(\mathbf{c}_0) \cap \mathrm{supp}(\mathbf{y})| \geq a/2$. Recall that for any two codewords in $T(Y)$, we have $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{c}')| \leq 1$ and $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{c}')| = 0$. Hence, we have $k \leq 2$ and $k' \leq 2$. Furthermore, the only codeword of weight e which may exist is $\mathbf{c}_0$ and we have $|T(Y)| \leq k + k' + 1$. Hence, $k + k' + 1 \leq 5 \leq 1 + \ell(q-1)$ as $q \geq 3$.

Let us finally consider Case 3) $\ell = 1$ and $a \geq 3$. We have $w(\mathbf{y}) = a$. Recall that for each $\mathbf{c} \in T(Y)$ with weight $e+1$, we have $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| \geq \lceil a/2 \rceil$ and this lower bound is attained only when $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{y})| \geq \lfloor a/2 \rfloor$. Similarly to the codewords of weight $e+1$ discussed in the second paragraph of the proof, we obtain that $|\mathrm{supp}(\mathbf{c}_0) \cap \mathrm{supp}(\mathbf{y})| \geq \lfloor a/2 \rfloor$. Denote by k the number of codewords with $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| \geq \lceil a/2 \rceil + 1$ and by k' the number of codewords with $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| \leq \lceil a/2 \rceil$. When $a \geq 5$ is odd, since we have $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{c}')| \leq 1$, there can be at most two codewords with $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| \geq (a+1)/2$. Hence, we have $k + k' \leq 3 \leq 1 + \ell(q-1)$, taking into account $\mathbf{c}_0$. When $a = 3$ and $q = 3$, the codewords in $T(Y)$ have a total of six 'symbols' at three coordinate positions available for them since $(q-1) \cdot w(\mathbf{y}) = 6$. As each codeword besides $\mathbf{c}_0$ uses at least two of them (since $\lceil a/2 \rceil = 2$) and the value supports may not intersect, we have $k + k' \leq 3 \leq \ell(q-1) + 1$. When $a = 3$ and $q \geq 4$, we may have at most three codewords whose supports intersect with $\mathrm{supp}(\mathbf{y})$ in addition to $\mathbf{c}_0$ since $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{c}')| \leq 1$ for any two codewords. Hence, $k + k' \leq 4 < \ell(q-1) + 1$.

When a is even, supports of any two codewords of weight $e+1$ cover $\mathrm{supp}(\mathbf{y})$. Indeed, since $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{c}')| \leq 1$ for any two codewords $\mathbf{c}, \mathbf{c}'$ of weight $e+1$, if codeword $\mathbf{c}$ intersects $\mathbf{y}$ on $a/2 + 1$ coordinates, then together with another codeword intersecting $\mathbf{y}$ on at least $a/2$ coordinates, they cover $\mathrm{supp}(\mathbf{y})$. If a codeword $\mathbf{c}$ has $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| = a/2$, then $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{y})| = a/2$. Since $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{c}')| = 0$ for two codewords $\mathbf{c}, \mathbf{c}'$ of weight $e+1$, they cover $\mathrm{supp}(\mathbf{y})$. In conclusion, any two codewords of weight $e+1$ cover $\mathrm{supp}(\mathbf{y})$. Hence, when $a \geq 6$, there are at most two codewords in $T(Y)$ in addition to $\mathbf{c}_0$ and $k + k' \leq 3$. Assume next $a = 4$. We have $k \leq 1$ since $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{c}')| \leq 1$ for any two codewords. Recall that if $|\mathrm{supp}(\mathbf{c}) \cap \mathrm{supp}(\mathbf{y})| = a/2$, then $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{y})| = a/2$ for any codeword $\mathbf{c}$ with $w(\mathbf{c}) = e+1$. If $k = 1$ and $\mathbf{c}_1$ is such that $|\mathrm{supp}(\mathbf{c}_1) \cap \mathrm{supp}(\mathbf{y})| \geq a/2 + 1 = 3$, then we can have only one (other) codeword with $|\mathrm{vsupp}(\mathbf{c}) \cap \mathrm{vsupp}(\mathbf{y})| \geq 2$ since $|\mathrm{vsupp}(\mathbf{c}') \cap \mathrm{vsupp}(\mathbf{c}'')| = 0$ and $|\mathrm{supp}(\mathbf{c}') \cap \mathrm{supp}(\mathbf{c}'')| \leq 1$ for each $\mathbf{c}', \mathbf{c}'' \in T(Y)$. Hence, $k' \leq 2$ in this case and $k + k' \leq 3$. Finally, if $k = 0$, then $k' \leq 3$ since there can be at most two codewords whose value supports intersect $\mathrm{vsupp}(\mathbf{y})$ at two coordinate positions. Hence, also in this case $k + k' \leq 3 \leq 1 + \ell(q-1)$.

Therefore, in each case we have $|T(Y)| \leq \ell(q-1) + 1$ and the claim follows. □

Theorem 10. *Let $n \geq \ell - 2 + (\ell-1)^2 2^b (q-1)^{b-1}$, C be an e-error-correcting code, $|Y| = N \geq V_q(n, \ell-1) + 1$, $\ell \geq 1$, $q \geq 3$ and $b \geq (\ell(q-1)+1)(2e+2)$. Then we have $\mathcal{L} \leq \ell(q-1)+1$.*

Proof. Let us denote $T(Y) = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{|T(Y)|-1}\}$. Moreover, let us assume without loss of generality that $\mathbf{c}_0 = \mathbf{0}$. We have $2e+1 \leq d(\mathbf{c}_i, \mathbf{c}_j) \leq 2e+2$ for each distinct pair of indices $i, j \in [0, |T(Y)|-1]$ by Lemma 8 and the definition of e-correcting codes. Thus, $w(\mathbf{c}_i) \leq 2e+2$ for each i. Suppose to the contrary that $|T(Y)| \geq \ell(q-1)+2$. Moreover, let us denote by $\overline{D} = \bigcup_{i=0}^{\ell(q-1)+1} \operatorname{supp}(\mathbf{c}_i)$ and let $\mathbf{y} \in Y$ be an output word such that $|\operatorname{supp}(\mathbf{y}) \setminus \overline{D}| \geq \ell - 1$. Word $\mathbf{y}$ exists by Lemma 7 since $n \geq \ell - 2 + (\ell-1)^2 2^b (q-1)^{b-1}$. Let us denote by $\mathbf{s} \in \mathbb{Z}_q^n$ a word such that $\operatorname{supp}(\mathbf{s}) = \overline{D} \cap \operatorname{supp}(\mathbf{y})$ and $\operatorname{vsupp}(\mathbf{s}) \subseteq \operatorname{vsupp}(\mathbf{y})$. We have

$$t \geq d(\mathbf{y}, \mathbf{c}_i) \geq \ell - 1 + d(\mathbf{s}, \mathbf{c}_i) \iff e+1 \geq d(\mathbf{s}, \mathbf{c}_i).$$

By Lemma 9 together with choice $\mathbf{w} = \mathbf{s}$, there are at most $\ell(q-1)+1$ codewords in $T(Y)$. This contradicts our assumption that $|T(Y)| \geq \ell(q-1)+2$. Thus, $|T(Y)| \leq \ell(q-1)+1$. Since this holds for any Y with $|Y| = N$, we have $\mathcal{L} \leq \ell(q-1)+1$. □

The next result shows that our bound on the list size is actually *optimal* for e-error-correcting code when $N = V_q(n, \ell-1)+1$.

Theorem 11. *Let $n \geq e(q-1)\ell + t$ and $N \leq V_q(n, \ell-1)+1$. There exists such an e-error-correcting code that*

$$\mathcal{L} \geq (q-1)\ell + 1.$$

Proof. Let us denote $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_{(q-1)\ell+1}\}$. For integers $h \in [0, \ell-1]$ and $p \in [1, q-1]$ let $i = h(q-1)+p$. For each $i \in [1, (q-1)\ell]$, we have

$$\operatorname{vsupp}(\mathbf{c}_i) = \{(h+1, p)\} \cup \{(j, 1) \mid j \in [(i-1)e + \ell + 1, i \cdot e + \ell]\}$$

and

$$\operatorname{vsupp}(\mathbf{c}_{(q-1)\ell+1}) = \{(j, 1) \mid j \in [e(q-1)\ell + \ell + 1, e(q-1)\ell + t]\}.$$

For example, if $e = \ell = 2$, $q = 3$ and $n = 12 = e(q-1)\ell + t$, then the code consists of the words

$\mathbf{c}_1 = 10\ 11\ 00\ 00\ 00\ 00$,

$\mathbf{c}_2 = 20\ 00\ 11\ 00\ 00\ 00$,

$\mathbf{c}_3 = 01\ 00\ 00\ 11\ 00\ 00$,

$\mathbf{c}_4 = 02\ 00\ 00\ 00\ 11\ 00$,

$\mathbf{c}_5 = 00\ 00\ 00\ 00\ 00\ 11$.

Let us now assume that $Y = \{\mathbf{y} \mid w(\mathbf{y}) \leq \ell - 1\} \cup \{\mathbf{y}_\ell\}$ where $\operatorname{vsupp}(\mathbf{y}_\ell) = \{(j, 1) \mid j \in [1, \ell]\}$. Observe that $|Y| = V_q(n, \ell-1)+1$.

We claim that C is an e-error-correcting code and $C \subseteq \bigcap_{\mathbf{y} \in Y} B_t(\mathbf{y})$. Let us first consider the minimum distance of C. Observe, $w(\mathbf{c}_{(q-1)\ell+1}) = e$ while all other codewords have weight $e+1$. Furthermore, no codewords have intersecting value supports, supports of $(e+1)$-weight words intersect in at most one coordinate (within the first ℓ coordinates) and the support of the e-weight codeword does not intersect with the support of any other codeword. Thus, C has minimum distance of $2e+1$ and is an e-error-correcting code.

Let us then consider the distance between the codewords in C and the output words in Y. Let $\mathbf{y} \in Y$. We have $d(\mathbf{c}_i, \mathbf{y}) \leq w(\mathbf{c}_i) + w(\mathbf{y})$. Thus, when $w(\mathbf{y}) \leq \ell - 1$, the distance is at most t. Hence, we only have to consider the distance between $\mathbf{y}_\ell$ and codewords of weight $e+1$. We immediately notice that $\mathbf{y}_\ell$ and those codewords have intersecting supports. So the distance is at most $d(\mathbf{c}_i, \mathbf{y}_\ell) \leq w(\mathbf{c}_i) + w(\mathbf{y}_\ell) - 1 = t + 1 - 1 = t$. Hence, the claim follows. □

We note that give large enough n, $\ell \geq 1$, $q \geq 3$ and $\ell(q-1)+1$ words $\mathbf{c} \in W$ with minimum distance $2e+1$, then the intersection of t-radius Hamming balls centred at words in W has cardinality of at most

$$\bigcap_{\mathbf{c} \in W} |B_t(\mathbf{c})| \leq V_q(n, \ell-1)+1.$$

5. Decoding algorithm for substitution errors

We describe the (well-known) majority algorithm in the q-ary Hamming space for substitution errors using similar terminology and notation as in [1]. In this section, we assume that $q \geq 3$ and $e \geq 1$. Previously a decoding algorithm of optimal complexity has been presented for all values of q and $e = 0$ in [23], for $q = 2$ and $e = 1$ in [27] and for $q = 2$ and all values of e in [1].

We denote the coordinates of the output words $\mathbf{y}_j \in Y$ by $\mathbf{y}_j = (y_{j,1}, y_{j,2}, \ldots, y_{j,n})$. Furthermore, the number of symbols j in the ith coordinates of the output words are respectively denoted by

$$m_{i,j} = |\{k \in \{1,2,\dots,N\} \mid y_{k,i} = j\}|.$$

Moreover, in the ith coordinate, the number of occurrences of the most common value j is denoted by $M(i) = m_{i,j}$ where $m_{i,j} \geq m_{i,j'}$ for each $0 \leq j' \leq q-1$.

Based on the set of output words Y, the *majority algorithm* with some threshold $\tau \geq N/2$ outputs the word $\mathrm{maj}_\tau(Y) = (z_1, z_2, \dots, z_n) \in \mathbb{Z}_q^n$, where

$$z_i = \begin{cases} j & \text{if } M(i) > \tau \text{ and } M(i) = m_{i,j} \text{ for some } j \in [0, q-1], \\ ? & \text{otherwise.} \end{cases}$$

In other words, for each coordinate of $\mathbf{z}$, we choose? or the most frequent symbol j ($j \in [1,q]$) if it is common enough.

Below we introduce a decoding algorithm for substitution errors in the q-ary Hamming space ($q \geq 3$). The algorithm takes $N_{t,e}$ output words as input and gives the transmitted word $\mathbf{x}$ as output. The algorithm requires some threshold constants. Let

$$\tau'_{t,e} = \frac{1}{e+1} V_q(n-e-1, \ell-1)$$

and

$$\tau_{t,e} = \tau'_{t,e} + \frac{e}{e+1} N_{t,e}.$$

In Algorithm 1, the notation $\mathcal{D}_C(\mathbf{u})$ means that the decoder of the code C decodes the word $\mathbf{u}$ and returns a codeword of C. We denote by $\mathcal{COM}(\mathcal{D}_C)$ the time complexity of decoder $\mathcal{D}_C$. The algorithm and proofs are inspired by [1].

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

Input: $N_{t,e}$ output words $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_{N_{t,e}}\}$
Output: Transmitted word $\hat{\mathbf{c}}$ ($= \mathbf{x}$)
1: $\mathbf{z} = \mathrm{maj}_{\tau_{t,e}}(Y)$
2: $S = \{i \mid i \in [1,n], z_i = ?\}$
3: $Z = \{\mathbf{u} \in \mathbb{Z}_q^n \mid u_i = z_i \text{ for all } i \notin S\}$
4: **for** each $\mathbf{u} \in Z$ **do** $\hat{\mathbf{c}} = \mathcal{D}_C(\mathbf{u})$.
5: **if** $Y \subseteq B_t(\hat{\mathbf{c}})$ **then**
6: **return** $\hat{\mathbf{c}}$
7: **end if**
8: **end for**

Let $i \in [1,n]$, $e_i = |\{\mathbf{y} \in Y \mid y_i \neq x_i\}|$ and $ce_i = |\max_{a \in \mathbb{Z}_q} \{\mathbf{y} \in Y \mid y_i \neq x_i \text{ and } y_i = a\}|$ where $\mathbf{x}$ is the transmitted word. In other words, e_i is the number of output words in Y for which an error has occurred in the ith bit and ce_i is the number of output words in Y for which the most common error has occurred in the ith bit (there are $q-1$ different error possibilities for each coordinate). Moreover, an error occurs in z_i if $ce_i > \tau_{t,e}$ and having the symbol? at the ith position requires that $N_{t,e} - \tau_{t,e} \leq e_i$. Indeed, otherwise $N_{t,e} - e_i > \tau_{t,e}$. Notice that we do not have a symbol? at the ith position even though $N_{t,e} - \tau_{t,e} \leq e_i$ if $ce_i > \tau_{t,e}$.

In the following lemma, we give an upper bound for the number coordinates in which z_i is interpreted as an incorrect symbol (other than?).

Lemma 12. *There are at most e errors in the word $\mathbf{z}$ in Step* 1.

Proof. Let us assume on the contrary that there are at least $e+1$ errors in $\mathbf{z}$. We may assume, without loss of generality, that those errors occur in the coordinates $[1, e+1]$. If more than $e+1$ errors occur, then we are interested only in the $e+1$ errors in the coordinates $[1, e+1]$. Let us first count the maximum number of output words which can have $e+1$ common errors among those coordinates. In other words, we count the number of output words which have the same erroneous symbols among the $e+1$ first coordinates. Recall that each output word is unique and thus, this number has an upper bound. Indeed, if we have $e+1$ shared/common errors in the $e+1$ first coordinates, then we can have at most $\ell-1$ errors in the other coordinates. Thus, there are at most $V_q(n-e-1, \ell-1)$ such words.

Let us approximate ce_i for $i \in [1, e+1]$. We have

$$ce_i > \tau_{t,e} = \tau'_{t,e} + \frac{e}{e+1} N_{t,e}.$$

Therefore,

$$\sum_{i=1}^{e+1} ce_i > (e+1)\tau'_{t,e} + eN_{t,e}. \tag{8}$$

Let us next consider an upper bound for $\sum_{i=1}^{e+1} ce_i$. As we have seen, there are at most $V_q(n-e-1, \ell-1) = (e+1)\tau'_{t,e}$ output words which can have shared errors in all of the $e+1$ first coordinates. Other $N_{t,e} - V_q(n-e-1, \ell-1)$ output words can have at most e errors in those coordinates. Thus,

$$\begin{aligned}\sum_{i=1}^{e+1} ce_i \leq &(e+1)^2\tau'_{t,e} + e(N_{t,e} - (e+1)\tau'_{t,e})\\ =&(e+1)\tau'_{t,e} + eN_{t,e}.\end{aligned}$$

Hence, we have a contradiction between the upper bound and the lower bound in (8) and the claim follows. □

Let us then consider the maximum cardinality of the set S in the algorithm.

Lemma 13. *We have $|S| < t(e+2)$ when n is large enough.*

Proof. Let us first give a useful approximation for $N_{t,e}$. Recall that e, ℓ and q are constants with respect to n. Hence, we have

$$\begin{aligned}N_{t,e} =&\sum_{i=0}^{\ell-1}\binom{n-2e-1}{i}(q-1)^i \sum_{k=e+i-(\ell-1)}^{t-i}\sum_{j=e+i-(\ell-1)}^{t-i}\binom{2e+1}{k}\binom{2e+1-k}{j}(q-2)^{2e+1-k-j}+1\\ =&\binom{n-2e-1}{\ell-1}(q-1)^{\ell-1}\sum_{k=e}^{e+1}\sum_{j=e}^{e+1}\binom{2e+1}{k}\binom{2e+1-k}{j}(q-2)^{2e+1-k-j}+\Theta(n^{\ell-2})\\ =&\binom{n-2e-1}{\ell-1}(q-1)^{\ell-1}\left(\binom{2e+1}{e}((e+1)(q-2)+2)\right)+\Theta(n^{\ell-2}).\end{aligned}$$

In the following, we consider the term of the previous sum with $n^{\ell-1}$. When n is large enough, we have

$$\begin{aligned}&\binom{n-2e-1}{\ell-1}(q-1)^{\ell-1}\left(\binom{2e+1}{e}((e+1)(q-2)+2)\right)\\ \geq&(q-1)^{\ell-1}(e+2)\binom{n-2e-1}{\ell-1}\binom{2e+1}{e}+(q-1)^{\ell-1}\binom{n-2e-1}{\ell-1}\binom{2e+1}{e}\\ \geq&(q-1)^{\ell-1}(e+2)\binom{n-2e-1}{\ell-1}\binom{2e+1}{e}+(e+2)V_q(n-e-1,\ell-2) \qquad (9)\\ =&(e+2)\left(\frac{\prod_{i=1}^{e}n-e-t+i}{\prod_{j=1}^{e}n-2e-1+j}\binom{2e+1}{e}(q-1)^{\ell-1}\binom{n-e-1}{\ell-1}+V_q(n-e-1,\ell-2)\right)\\ \geq&(e+2)\left((q-1)^{\ell-1}\binom{n-e-1}{\ell-1}+V_q(n-e-1,\ell-2)\right) \qquad (10)\\ =&(e+2)V_q(n-e-1,\ell-1).\end{aligned}$$

Above, Inequality (9) follows with the observations $\binom{n-2e-1}{\ell-1}\in\Theta(n^{\ell-1})$ and $V_q(n-e-1,\ell-2)\in\Theta(n^{\ell-2})$. In Inequality (10), we have $\frac{\prod_{i=1}^{e}n-e-t+i}{\prod_{j=1}^{e}n-2e-1+j}\binom{2e+1}{e}\geq\left(\frac{n-e-t+1}{n-e-1}\right)^e\binom{2e+1}{e}\geq 1$, when n is large and $e\geq 1$. Thus, for large enough n, we may use $N_{t,e}/(e+2) > V_q(n-e-1,\ell-1)$ and $\tau'_{t,e} < \frac{1}{(e+1)(e+2)}N_{t,e}$.

We have $N_{t,e}$ channels, in each of which at most t errors occurs. Thus, $\sum_{i=1}^{n} e_i \leq tN_{t,e}$. Moreover, for each element in S, we require at least $N_{t,e}-\tau_{t,e}$ errors. Thus,

$$|S| \leq \frac{tN_{t,e}}{N_{t,e}-\tau_{t,e}} = t\frac{N_{t,e}}{\frac{1}{e+1}N_{t,e}-\tau'_{t,e}} < t\frac{N_{t,e}}{\frac{1}{e+1}(\frac{e+1}{e+2}N_{t,e})} = t(e+2). \quad \square$$

Finally, we conclude in the next theorem that the algorithm works as intended with complexity $\Theta(nN_{t,e}) = \Theta(n^\ell)$ (provided that $\mathcal{COM}(\mathcal{D}_C)\in O(n^\ell)$). Indeed, we have $N_{t,e}\in\Theta(n^{\ell-1})$, see Theorem 1. Notice that $\Theta(n^\ell)$ is the optimal complexity in the sense that it takes $\Theta(nN_{t,e})$ time to read all the input words.

Theorem 14. *The complexity of Algorithm 1 is $\Theta(n^\ell + \mathcal{COM}(\mathcal{D}_C))$ and it outputs the transmitted codeword $\mathbf{x}$ when it has input of $N_{t,e}$ words in Y.*

Proof. By Lemma 12, there are at most e errors in $\mathbf{z}$. Moreover, by Lemma 13, there are at most $t(e+2)$ symbols? in $\mathbf{z}$. Thus, there are at most $q^{t(e+2)}$ different possible choices for $\hat{\mathbf{c}}$ in Step 4, when we go through all the possible combinations of symbols that might occur in the coordinates with?. Thus, at least one of these words $\mathbf{u}$ in Step 4 is identical in these particular coordinates to the transmitted word $\mathbf{x}$. Moreover, since there are at most e errors in this word, the decoder $\mathcal{D}_C$ correctly decodes this word as $\mathbf{x}$. Therefore, the transmitted word $\mathbf{x}$ is among the codewords $\hat{\mathbf{c}}$. Finally, $\mathbf{x}$ is the only word which we can output, since we have $N_{t,e}$ channels and there is only one codeword in $\bigcap_{\mathbf{y}\in Y} B_t(\mathbf{y})$ by Theorem 1. Thus, the algorithm works.

We observe that Step 1 has complexity $\Theta(Nn)$. Steps 2 and 3 have complexity $\Theta(n)$. Finally, Steps 4 – 8 have complexity $q^{t(e+2)}\mathcal{COM}(\mathcal{D}_C)$ (here $q^{t(e+2)}$ is constant) plus complexity of $q^{t(e+2)}\Theta(Nn)$ since it takes $\Theta(Nn)$ time for each candidate $\hat{\mathbf{c}}$ to test whether $Y \subseteq B_t(\hat{\mathbf{c}})$. Thus, the claim holds. □

In the following, we introduce a list-decoding algorithm for substitution errors in binary Hamming spaces. In [19], the authors have studied substitution errors in the *binary* Hamming space together with list-error-correcting codes. In what follows, we first recall some useful results for the list size $\mathcal{L}$ previously presented in [19]. Then, we proceed by providing a new decoding algorithm of optimal time complexity list-error-correcting codes. In this section, we assume that C is a list error-correcting code with $|B_{e+a}(\mathbf{u}) \cap C| \leq M$ for some constants M and $a \in [0, \ell - 1]$.

The first theorem gives a general upper bound.

Theorem 15 *(Theorem 21, [19]). Let $N \geq V_2(n, \ell - a - 1) + 1$ where $0 \leq a \leq \ell - 1$. Let C be an e-error-correcting code such that $|B_{e+a}(\mathbf{u}) \cap C| \leq M$ for every $\mathbf{u} \in \mathbb{Z}_2^n$. Consequently,*

$$\mathcal{L} \leq 2^{\ell - a} M.$$

The second useful theorem gives an improved upper bound but requires that n is rather large.

Theorem 16 *(Theorem 26, [19]). Let $N \geq V_2(n, \ell - a - 1) + 1$, $n \geq (\ell - a - 1)^2 2^b + \ell - a - 2$, $\ell - 1 \geq a \geq 1$ and $b = \left\lceil (2e + 2a + 2)^{e \cdot (e+a+1)!} \right\rceil$. Moreover, let C be such an e-error-correcting code that there are at most M codewords in any $(e + a)$-radius ball. Then*

$$\mathcal{L} \leq \max\{(t + 1)M, b/(2e + 2a + 2)\}.$$

The decoding algorithm we are using is quite similar to Algorithm 1 and the algorithm in [1]. One of the main differences is that we introduce a new variable $a \in [0, \ell - 1]$ which affects our choices for threshold constant τ. Let

$$\tau'_{t,e,a} = \frac{1}{e + a + 1} V_2(n - e - 1 - a, \ell - 1 - a)$$

and

$$\tau_{t,e,a} = \tau'_{t,e,a} + \frac{e + a}{e + a + 1} N_{t,e,a}.$$

Moreover, let us have $N_{t,e,a} = \lceil (1 + \epsilon) V_2(n - e - 1 - a, \ell - 1 - a) \rceil + 1$ for some constant $\epsilon > 0$. Let $\mathcal{D}_C$ be a decoder for C such that, given a word $\mathbf{u} \in \mathbb{Z}_2^n$, the decoder outputs a list $\mathcal{D}_C(\mathbf{u})$ of codewords such that $C \cap B_{e+a}(\mathbf{u}) \subseteq \mathcal{D}_C(\mathbf{u})$ and $|\mathcal{D}_C(\mathbf{u})| \leq M$.

Algorithm 2 List decoding substitution errors in $\mathbb{Z}_2^n$.

Input: $N_{t,e,a}$ output words $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_{N_{t,e,a}}\}$
Output: List of possible transmitted words $T(Y)$ with $\mathbf{x} \in T(Y)$
1: $\mathbf{z} = \mathrm{maj}_{\tau_{t,e,a}}(Y)$
2: $S = \{i \mid i \in [1, n], z_i = ?\}$
3: $Z = \{\mathbf{u} \in \mathbb{Z}_2^n \mid u_i = z_i \text{ for all } i \notin S\}$
4: **for** each $\mathbf{u} \in Z$ **do** calculate $\mathcal{D}_C(\mathbf{u})$
5: **for** each $\hat{\mathbf{c}} \in \mathcal{D}_C(\mathbf{u})$ **do**
6: **if** $Y \subseteq B_t(\hat{\mathbf{c}})$ **then**
7: Add $\hat{\mathbf{c}}$ to $T(Y)$
8: **end if**
9: **end for**
10: **end for**
11: **return** $T(Y)$

Observe that since we are now considering binary Hamming space, an error occurs in the ith coordinate if and only if $e_i > \tau_{t,e,a}$. Moreover, we have the symbol? in the ith coordinate if $N_{t,e,a} - \tau_{t,e,a} \leq e_i \leq \tau_{t,e,a}$. Let us first show that we have at most $e + a$ errors in the majority word $\mathbf{z}$.

Lemma 17. *There are at most $e + a$ errors in the word $\mathbf{z}$ in Step 1.*

Proof. As the proof is quite similar to the proof of Lemma 12, we postpone the proof to Appendix. □

Let us then consider the maximum cardinality of the set S in the algorithm.

Lemma 18. *We have $|S| < \frac{t(e+a+1)(1+\epsilon)}{\epsilon}$.*

Proof. Let $N_{t,e,a} = \lceil (1+\epsilon)V_2(n-e-a-1,\ell-a-1)\rceil + 1 = (1+\epsilon)V_2(n-e-a-1,\ell-a-1) + 1 + c$ where $0 \le c < 1$ is a constant. We have $N_{t,e,a} = \lceil (1+\epsilon)V_2(n-e-a-1,\ell-a-1)\rceil + 1 = (1+\epsilon)(e+a+1)\tau'_{t,e,a} + 1 + c$ channels, in each of which at most t errors occurs. Thus, $\sum_{i=1}^{n} e_i \le tN_{t,e,a}$. Moreover, for each element in S, we require at least $N_{t,e,a} - \tau_{t,e,a}$ errors.

Thus,

$$\begin{aligned}
|S| \le & \frac{tN_{t,e,a}}{N_{t,e,a} - \tau_{t,e,a}} \\
= & t\frac{N_{t,e,a}}{\frac{1}{e+a+1}N_{t,e,a} - \frac{1}{e+a+1}V_2(n-e-1-a,\ell-1-a)} \\
= & t\frac{(e+a+1)((1+\epsilon)V_2(n-e-a-1,\ell-a-1)+1+c)}{(1+\epsilon)V_2(n-e-a-1,\ell-a-1) - V_2(n-e-a-1,\ell-a-1)+1+c} \\
= & \frac{t(e+a+1)(1+\epsilon)}{\epsilon} \cdot \frac{V_2(n-e-a-1,\ell-a-1) + \frac{1+c}{1+\epsilon}}{V_2(n-e-a-1,\ell-a-1) + \frac{1+c}{\epsilon}} \\
< & \frac{t(e+a+1)(1+\epsilon)}{\epsilon}. \quad \square
\end{aligned}$$

Finally, we conclude that the algorithm works as intended with optimal (assuming that we read all the output words) complexity $\Theta(Nn)$ when ϵ and M are constants, and the complexity of the list-decoder $\mathcal{COM}(\mathcal{D}_C)$ belongs to $O(Nn)$.

Theorem 19. *Algorithm 2 outputs a list $T_D(Y)$, with $\mathbf{x} \in T_D(Y)$ and $|T_D(Y)| \le \mathcal{L}$, and its (optimal) complexity is $\Theta(Nn + \mathcal{COM}(\mathcal{D}_C))$ when the input is $N_{t,e,a}$ words in Y.*

Proof. As the proof is quite similar to the proof of Theorem 14, we postpone the proof to Appendix. $\square$

In Theorem 19, we say that $|T_D(Y)| \le \mathcal{L}$. Some upper bounds for $\mathcal{L}$ have been presented in Theorems 15 and 16. These Theorems require that $N \ge V_2(n,\ell-a-1)+1$. Observe that when $n \ge \frac{(t-1)(1+\epsilon)^{1/(e+a+1)}}{(1+\epsilon)^{1/(e+a+1)}-1}$, we have $(1+\epsilon)V_2(n-e-a-1,\ell-a-1) \ge V_2(n,\ell-a-1)$. Indeed, $V_2(n-e-a-1,\ell-a-1) \ge (\frac{n-t+1}{n})^{e+a+1}V_2(n,\ell-a-1)$, since for any integer $b \in [0,\ell-a-1]$ we have

$$\begin{aligned}
\binom{n-e-a-1}{\ell-a-1-b} = & \frac{\prod_{i=1}^{e+a+1} n-t+b+i}{\prod_{j=1}^{e+a+1} n-e-a-1+j}\binom{n}{\ell-a-1-b} \\
\ge & \left(\frac{n-t+1}{n}\right)^{e+a+1}\binom{n}{\ell-a-1-b}
\end{aligned}$$

and $(1+\epsilon)(\frac{n-t+1}{n})^{e+a+1} \ge 1$ when $(1+\epsilon)^{1/(e+a+1)} \ge \frac{n}{n-t+1}$. Moreover, this last condition is satisfied when $n \ge \frac{(t-1)(1+\epsilon)^{1/(e+a+1)}}{(1+\epsilon)^{1/(e+a+1)}-1} > t-1$. Hence, when this requirement is fulfilled, we may use Theorems 15 and 16. Overall, the smaller the constant ϵ is, the less channels we require. However, we have a trade-off; smaller ϵ requires larger n and the complexity of the algorithm will increase (slightly) as we will see in Theorem 19 based on Lemma 18.

In particular, Theorems 15 and 16 offer following bounds for $\mathcal{L}$. If $a \in [1,\ell-1]$, $|B_{e+a}(\mathbf{u}) \cap C| \le M$ for every $\mathbf{u} \in \mathbb{Z}_2^n$, $b = \lceil (2e+2a+2)^{e\cdot(e+a+1)!}\rceil$ and $n \ge (\ell-a-1)^2 2^b + \ell - a - 2$, then $\mathcal{L} \le \min\{2^{\ell-a}M, \max\{(t+1)M, b/(2e+2a+2)\}\}$ and if $n < (\ell-a-1)^2 2^b + \ell - a - 2$, then $\mathcal{L} \le 2^{\ell-a}M$. Moreover, if $a = 0$, then $\mathcal{L} \le 2^{\ell}$ (in this case $M = 1$ since C is an e-error-correcting code).

Remark 20. Observe that when $a = 0$ (and $M = 1$), by choosing a suitable ϵ, we can find a number of channels which gives an efficient decoding algorithm for the case $\mathcal{L} \le 2$ by using [27, Theorem 6] or [19, Corollary 16]. Indeed, now $N_{t,e,0} = \lceil (1+\epsilon)V_2(n-e-1,\ell-1)\rceil + 1 \ge V_2(n,\ell-1)+1$ when $n \ge \frac{(t-1)(1+\epsilon)^{1/(e+1)}}{(1+\epsilon)^{1/(e+1)}-1}$ and hence, the number of channels is large enough for $\mathcal{L} \le 2$ when n is large enough.

6. Probabilities with large q for various error types

Next, we consider the likelihood for $|T(Y)| = 1$ when q is large, other parameters are constants and only single error type occurs. In particular, we show that insertion, substitution and deletion errors behave differently while deletion and erasure errors have similar behaviours. In this section, we work under the assumption that every output word is equally probable and that we may obtain the same output word from multiple channels. In other words, given a transmitted word $\mathbf{x}$, the probability that we obtain a given ordered multiset Y of N output words is $1/|B_t(\mathbf{x})|^N$ where $B_t(\mathbf{x})$ is the t-radius ball centred at $\mathbf{x}$ with suitable error-metric. Note that a case in which each channel having a unique set of errors (instead of unique output word) has been considered in [18,20]. These approaches differ in the cases of insertion and deletion errors.

In the following theorem, we first consider *insertion* errors together with large q from probabilistic viewpoint. The result can be somewhat surprising as in [23, Equation (51)], Levenshtein has shown that increasing q implies that we require more channels to have $\mathcal{L} = 1$, that is, $|T(Y)| = 1$ for all Y.

Theorem 21. *Let $N \geq 2$ and $C \subseteq \mathbb{Z}_q^n$. Assume that a word $\mathbf{x}$ is transmitted through N channels and a multiset of output words Y is obtained. Moreover, at most t insertion errors occur in each channel. Then the probability that $|T(Y)| = 1$ approaches 1 as q grows and we have a probabilistic decoding algorithm with complexity $O(q)$ which never returns an incorrect result.*

Proof. Let $\mathbf{y}_1$ and $\mathbf{y}_2$ be any (possibly $\mathbf{y}_1 = \mathbf{y}_2$) output words of Y. Let their lengths be $n + t_1$ and $n + t_2$, respectively where $t_1, t_2 \leq t$. Thus, there are at most $2t$ new symbols in $\mathbf{y}_1$ and $\mathbf{y}_2$ which are not present in the transmitted word $\mathbf{x}$. Moreover, there are at most n different symbols in $\mathbf{x}$. Recall that we assumed every output word to have the same probability. We note that when the insertion error inserts a symbol which already appears in an arbitrary word $\mathbf{w}$, we may arrive to the same output word with multiple different insertion errors. For example, if $\mathbf{w} = 0$, then inserting 0 as the first or the second symbol, leads to the same word 00 while if we had inserted symbol 1, then we could obtain words 10 and 01. In general, inserting symbols not present in $\mathbf{x}$ can result in more unique output words than inserting symbols which appear in $\mathbf{x}$. Thus, there are more combinations of insertions leading to output words with inserted symbol not present in $\mathbf{x}$ than there are combinations leading to output words with inserted symbol being in $\mathbf{x}$. Hence, the probability that the first inserted symbol is not in $\mathbf{x}$, is at least $(q-n)/q$ and for the jth inserted symbol the probability for not appearing in $\mathbf{x}$ or being already inserted is at least $(q-n-(j-1))/q$. Since $\mathbf{y}_1$ has at most t symbols not in $\mathbf{x}$, the probability that among the t_2 symbols inserted to $\mathbf{x}$ to form $\mathbf{y}_2$, we are only using symbols which are not present in the word $\mathbf{x}$ or in $\mathbf{y}_1$ and the t_2 inserted symbols all differ from each other, is at least

$$\prod_{i=0}^{t_2-1} \frac{q-n-t-i}{q} \geq \prod_{i=0}^{t-1} \frac{q-n-t-i}{q} \geq \left(1 - \frac{n+2t-1}{q}\right)^t \overset{q\to\infty}{\to} 1.$$

Furthermore, any symbol which appears in $\mathbf{x}$, appears at least twice in the words $\mathbf{y}_1$ and $\mathbf{y}_2$ (at least once in both of these words). Thus, by removing any symbol from $\mathbf{y}_2$ which does not appear in $\mathbf{y}_1$, we are very likely to obtain the transmitted word $\mathbf{x}$ as q tends to infinity. Finally, notice that we do only simple operations on $\mathbf{y}_2$. Thus, the time complexity of the decoding algorithm is $O(q)$. Moreover, we may verify whether the decoding algorithm gives a correct result since we know that the transmitted word has length n and we can check whether $\mathbf{y}_2$ has length n after we have deleted the symbols. Indeed, if after the deletions $\mathbf{y}_2$ has length n, then it forms the transmitted word $\mathbf{x}$ since we have deleted only symbols which have not belonged to $\mathbf{x}$. □

In the following proposition, we show that *substitution* errors do not behave similarly as insertion errors for any set of parameters as q grows.

Proposition 22. *Let $N \geq 1$ and $C = \mathbb{Z}_q^n$. Assume that there occur at most $t > 0$ substitution errors in each channel. Then the probability for $|T(Y)| = 1$ does not approach 1 as q grows for any transmitted word and set of parameters n, N and t.*

Proof. We first show that the probability that exactly t errors occur in a channel tends to 1 as q increases. Notice that exactly i errors may occur in $(q-1)^i\binom{n}{i}$ ways. Hence, the probability is

$$\begin{aligned}
&\frac{(q-1)^t\binom{n}{t}}{\sum_{i=0}^{t}(q-1)^i\binom{n}{i}} \\
=&1 - \frac{\sum_{i=0}^{t-1}(q-1)^i\binom{n}{i}}{\sum_{i=0}^{t}(q-1)^i\binom{n}{i}} \\
\geq&1 - \frac{\sum_{i=0}^{t-1}(q-1)^i\binom{n}{i}}{(q-1)^t} \overset{q\to\infty}{\to} 1.
\end{aligned}$$

Moreover, when exactly t substitution errors occur in each channel, the probability that there occurs an error in each channel in the ith coordinate, for some $i \in [1,n]$, depends only on the parameters n, N and t. Indeed, for a single channel, that probability is $(q-1)\cdot\binom{n-1}{t-1}(q-1)^{t-1}/\left(\binom{n}{t}(q-1)^t\right) = \binom{n-1}{t-1}/\binom{n}{t}$. Hence, the probability that each channel has a substitution error at the ith coordinate is $P = \left(\binom{n-1}{t-1}/\binom{n}{t}\right)^N$. Furthermore, the probability P is a positive constant since n, N and t are constants. Since $C = \mathbb{Z}_q^n$ and assuming that a substitution error occurs in the ith coordinate of each output word of Y, we have $|T(Y)| > 1$. Indeed, for example, if a substitution error occurs in each channel in the first symbol, then $\mathbf{x}, \mathbf{x}' \in T(Y)$ where $\mathbf{x}'$ is such that $x'_1 = x_1 + 1$ and $x'_i = x_i$ for each $i \in [2,n]$. Thus, the probability that $|T(Y)| = 1$ approaches at most $1 - P$, instead of 1. Thus, the claim follows. □

We can also see that in the case of substitution errors, the increase of q may affect on the probability that $|T(Y)| = 1$. Consider for example the case with $n = 2, t = 1, N = 3, \mathbf{x} = 00$. The number of possible output words is $1 + 2(q-1)$. Moreover, we have $|T(Y)| = 1$

if and only if each output word is unique and at least one substitution error occurs in the first coordinate for some output word $\mathbf{y}_1$ and in the second coordinate in another output word $\mathbf{y}_2$.

Moreover, there are $2q(q-1)(q-2)$ possible ordered combinations for output words $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ such that they are unique and substitutions resulting to them occur either all in the first coordinate or all in the second coordinate. Beyond these ordered output word sets, there are also $(1+2(q-1))^3 - (1+2(q-1))(2(q-1))(2(q-1)-1)$ possible output word sequences with at least two identical output words. Therefore, the probability that we may deduce $\mathbf{x}$ is

$$\begin{aligned} P' &= \frac{(1+2(q-1))^3 - ((1+2(q-1))^3 - (1+2(q-1))(2(q-1))(2(q-1)-1)) - 2q(q-1)(q-2)}{(1+2(q-1))^3} \\ &= \frac{6q^3 - 18q^2 + 18q - 6}{8q^3 - 12q^2 + 6q - 1} = \frac{6(q-1)^3}{(2q-1)^3}. \end{aligned}$$

We notice in particular, that P' is not a constant on q and therefore, the choice for q affects on the probability whether we may deduce $\mathbf{x}$ from Y. Notice that the fact that probability P' depends on q, does not contradict the proof of Proposition 22 since we only claimed that the probability P has an upper bound with value less than 1 which is independent of q.

Finally, let us consider *deletion* and *erasure* errors with $q > n \geq 2$. In the following proposition, for clarity, we avoid writing $\mathbf{x} = \mathbf{x}'$ for two identical words when they are considered over different alphabets. For example, for $\mathbf{x} = 00 \in \mathbb{Z}_4^2$ and $\mathbf{x}' = 00 \in \mathbb{Z}_6^2$ we say that the words are *identical* rather than write $\mathbf{x} = \mathbf{x}'$. Similarly, two sets over different alphabets are called *identical* if their words are identical.

Proposition 23. *Let N, n and t be constant. Further, let $q' > q > n$ and $C = \mathbb{Z}_q^n$. Let $\mathbf{x} \in \mathbb{Z}_q^n$ and $\mathbf{x}' \in \mathbb{Z}_{q'}^n$ be identical to $\mathbf{x}$. Then, sets $B_t^d(\mathbf{x})$ (resp. $B_t^e(\mathbf{x})$) and $B_t^d(\mathbf{x}')$ (resp. $B_t^e(\mathbf{x}')$) are identical and we have*

$$P(T(Y) = \{\mathbf{x}\}) = P(T(Y') = \{\mathbf{x}'\})$$

for the probabilities over all possible output word multisets Y and Y' obtained from $\mathbf{x}$ and $\mathbf{x}'$ with deletion (resp. erasure) errors.

Proof. We prove this proposition for deletion errors. The proof for erasures is similar. First we show that deletion balls $B_t^d(\mathbf{x})$ and $B_t^d(\mathbf{x}')$ are identical. We may observe that if a word $\mathbf{y}$ can be obtained from $\mathbf{x}$ with at most t deletions, then it can also be obtained from $\mathbf{x}'$ with at most t deletions and vice versa. Thus, the two balls are identical. Therefore, the probability for a particular ordered multiset Y or Y' of output words to occur is $1/|B_t^d(\mathbf{x})|^N$ where Y and Y' are two identical output word sets one of which is obtained from $\mathbf{x}$ and one from $\mathbf{x}'$.

As $q' > q > n$, we may assume without loss of generality that the symbols of the identical words $\mathbf{x}$ and $\mathbf{x}'$ belong to the set $\{0, 1, 2, \ldots, n-1\}$. In the following, we show that if Y and Y' are identical, then $T(Y) = \{\mathbf{x}\}$ if and only if $T(Y') = \{\mathbf{x}'\}$. Assume first that $T(Y) \neq \{\mathbf{x}\}$. In particular, let $\mathbf{c} \neq \mathbf{x}$ be such that $\mathbf{c} \in T(Y)$. Notice that in this case we have $\mathbf{c}'$, a word which is identical to $\mathbf{c}$, in the set $T(Y')$ and hence $T(Y') \neq \{\mathbf{x}'\}$. Assume then that $T(Y) = \{\mathbf{x}\}$ and $T(Y') \neq \{\mathbf{x}'\}$. We have $\mathbf{x}' \in T(Y')$. Hence, we may assume that $\mathbf{x}' \neq \mathbf{c}' \in T(Y')$. Notice that if there exists $\mathbf{c} \in \mathbb{Z}_q^n$ which is identical to $\mathbf{c}'$, then $\mathbf{c} \in T(Y)$, a contradiction. Hence, $\mathbf{c}'$ has symbols in $\{q, q+1, \ldots, q'-1\}$. Consider next a word $\mathbf{c}'' \in \mathbb{Z}_q^n$ which is identical to $\mathbf{c}'$ except that each symbol in $\{q, q+1, \ldots, q'-1\}$ is converted to n (note that since $n < q$, we have $n \in \mathbb{Z}_q$). Since $\mathbf{c}' \in T(Y')$, we have $\mathbf{c}'' \in T(Y)$. Indeed, since Y and Y' are identical, no word in Y' contains a symbol in $\{q, q+1, \ldots, q'-1\}$. Thus, if the transmitted word had contained any of them, then we would have deleted each of them. Furthermore, in this case the resulting output word is unaffected by whether the deleted symbol was n or some $n' > n$. Hence, $\mathbf{c}'' \in T(Y)$. Recall that n does not belong to the presentation of $\mathbf{x}$. Thus, $\mathbf{c}'' \neq \mathbf{x}$ and $T(Y) \neq \{\mathbf{x}\}$, a contradiction.

We have now shown, that if Y and Y' are identical, then $T(Y) = \{\mathbf{x}\}$ if and only if we have $T(Y') = \{\mathbf{x}'\}$. Furthermore, we have shown that if the probability for obtaining ordered multiset Y from $\mathbf{x}$ is P, then the probability for obtaining identical ordered multiset Y' from $\mathbf{x}'$ is also P. Now, the claim follows from these two observations. □

To conclude, we have observed that when q tends to infinity, deletions, substitutions and insertion errors operate in different ways. In the case of insertion errors, the probability that $|T(Y)| = 1$ tends to 1. For substitution errors, the probability that $|T(Y)| = 1$ is bounded from above by a constant other than 1. Finally, in the case of deletion (or erasure) errors, the probability that $|T(Y)| = 1$, is unaffected by the growth of q when $q > n$. The result for insertion errors is somewhat surprising.

7. Conclusion

In this paper, we have focused on the Levenshtein's reconstruction problem in the case of q-ary alphabets with respect to various error types. One of our main objectives has been generalizing known results for binary words to q-ary cases:

- Previously, in [17], the exact number of channels required to obtain a constant list size $\mathcal{L}$ (with respect to n) has been determined in the binary case. In Section 4, we generalized the result for the q-ary alphabet (with $q \geq 3$). Moreover, our result gave an insight to the number of words in the intersection of Hamming balls of radius t, which is a rather difficult problem in general.
- For substitution errors, a majority algorithm of optimal complexity for decoding the submitted word has been presented for $e = 0$ ([23]), $q = 2$ and $e = 1$ ([27]), and $q = 2$ and $e > 1$ ([1]). In Section 5, we extended these results to the cases with $q \geq 3$ and $e \geq 1$.

Moreover, we presented a list-decoding version of the algorithm for $q = 2$. Notice that here we are restricted to the case with $q = 2$ due to the fact that some underlying results (of [19]) required for the algorithm are only known for the binary words. Hence, a natural direction for future research would be to generalize the results of [19] to larger q and then extend the list-decoding algorithm for general q.

The theme of q-ary alphabets also continues in other sections of the paper:

- In Section 2, we studied erasure errors in q-ary alphabets (with $q \geq 2$), where the introduction of the symbol $*$ corresponding to the erasure error could be interpreted as switching from a q-ary alphabet to one with $q + 1$ symbols. Then, in Section 3, we combine erasures with substitution errors. For future studies, combining erasures with other error types would be interesting and natural.
- In Section 6, we investigated the behaviour of different error types when the size q of the alphabet grows (other parameters remain constant). We notice that under these circumstances different error types behave differently.

CRediT authorship contribution statement

Ville Junnila: Writing – review & editing, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Tero Laihonen:** Writing – review & editing, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Tuomo Lehtilä:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ville Junnila reports financial support was provided by Research Council of Finland. Tero Laihonen reports financial support was provided by Research Council of Finland. Tuomo Lehtila reports financial support was provided by Research Council of Finland. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Decoding algorithm for list-error-correcting codes

Lemma 17. *There are at most $e + a$ errors in the word $\mathbf{z}$ in Step 1.*

Proof. Let us assume on the contrary that there are at least $e + a + 1$ errors in $\mathbf{z}$. We may assume, without loss of generality, that those errors occur in the coordinates $[1, e + a + 1]$. If more than $e + a + 1$ errors occur, then we are interested only in the $e + a + 1$ errors in the coordinates $[1, e + a + 1]$. Let us first count the maximum number of output words, which can have $e + a + 1$ errors in those coordinates. Recall that each output word is unique and thus, this number has an upper bound. Indeed, if we have $e + a + 1$ errors in the $e + a + 1$ first coordinates, then we can have at most $\ell - a - 1$ errors in the other coordinates. Thus, there are at most $V_2(n - e - a - 1, \ell - a - 1)$ such words.

Let us approximate e_i for $i \in [1, e + a + 1]$. We have

$$e_i > \tau_{t,e,a} = \tau'_{t,e,a} + \frac{e + a}{e + a + 1} N_{t,e,a}.$$

Therefore,

$$\sum_{i=1}^{e+a+1} e_i > (e + a + 1)\tau'_{t,e,a} + (e + a)N_{t,e,a}. \tag{11}$$

Let us next consider an upper bound for $\sum_{i=1}^{e+a+1} e_i$. As we have seen, there are at most $V_2(n - e - a - 1, \ell - a - 1) = (e + a + 1)\tau'_{t,e,a}$ output words which can have errors in all of the $e + a + 1$ first coordinates. Other $N_{t,e,a} - V_2(n - e - a - 1, \ell - a - 1)$ output words can have at most $e + a$ errors in those coordinates. Thus,

$$\begin{aligned}\sum_{i=1}^{e+a+1} e_i \leq & (e + a + 1)^2 \tau'_{t,e,a} + (e + a)(N_{t,e,a} - (e + a + 1)\tau'_{t,e,a}) \\ = & (e + a + 1)\tau'_{t,e,a} + (e + a)N_{t,e,a}.\end{aligned}$$

Hence, we have a contradiction between the upper bound and the lower bound (11) and the claim follows. □

Theorem 19. *Algorithm 2 outputs a list $T_D(Y)$, with $\mathbf{x} \in T_D(Y)$ and $|T_D(Y)| \leq \mathcal{L}$, and its (optimal) complexity is $\Theta(Nn + C\mathcal{OM}(\mathcal{D}_C))$ when the input is $N_{t,e,a}$ words in Y.*

Proof. By Lemma 17, there are at most $e+a$ errors in $\mathbf{z}$. Moreover, by Lemma 18, we have $|S| \leq \frac{t(e+a+1)(1+\epsilon)}{\epsilon}$ where ϵ is a constant. Thus, we have $|Z| \leq 2^{\lfloor \frac{t(e+a+1)(1+\epsilon)}{\epsilon} \rfloor}$. Now, in Step 4, we calculate $\mathcal{D}_C(\mathbf{u})$ for $|Z|$ words $\mathbf{u}$. Furthermore, the list decoder $\mathcal{D}_C$ outputs a list of at most M codewords $\hat{\mathbf{c}}$ for each word $\mathbf{u}$. Now, for each $\hat{\mathbf{c}}$ we check if $\hat{\mathbf{c}} \in \bigcap_{\mathbf{y}\in Y} B_t(\mathbf{y})$ and if it is, then we add it to $T_D(Y)$. Observe that there exists a word $\mathbf{u}$ such that $d(\mathbf{u},\mathbf{x}) \leq e+a$ and $\mathbf{x} \in T_D(Y)$ where $\mathbf{x}$ is the transmitted word. Since we have $N_{t,e,a} \geq V_2(n, \ell - a - 1) + 1$ channels, cardinality of the set $\bigcap_{\mathbf{y}\in Y} B_t(\mathbf{y}) \cap C$ is bounded from above by Theorems 15 and 16. Thus, the algorithm works.

We observe that Step 1 has complexity $\Theta(Nn)$. Step 2 has complexity $\Theta(n)$. Similarly Step 3 has complexity $\Theta(n)$ since we compute $2^{|S|}$, a constant number, different words of length n. Finally, in Steps 4 – 10, the complexity of calculating codewords $\hat{\mathbf{c}} \in \mathcal{D}_C(\mathbf{u})$ is $\mathcal{COM}(\mathcal{D}_C)$, there are at most M codewords in $\mathcal{COM}(\mathcal{D}_C)$ and we do it at most $|Z| \leq 2^{\lfloor \frac{t(e+a+1)(1+\epsilon)}{\epsilon} \rfloor}$ times (which is a constant). After that, in Step 5 checking whether a codeword $\hat{\mathbf{c}}$ belongs to $\bigcap_{\mathbf{y}\in Y} B_t(\mathbf{y}) \cap C$ has complexity of $\Theta(Nn)$ and we do it at most $|Z|M$ times. Thus, the claim holds. □

References

[1] M. Abu-Sini, E. Yaakobi, On Levenshtein's reconstruction problem under insertions, deletions, and substitutions, IEEE Trans. Inf. Theory 67 (11) (2021) 7132–7158.
[2] M. Abu-Sini, 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.
[3] M. Abu-Sini, E. Yaakobi, On the intersection of multiple insertion (or deletion) balls and its application to list decoding under the reconstruction model, IEEE Trans. Inf. Theory (2023).
[4] J. Acharya, H. Das, O. Milenkovic, A. Orlitsky, S. Pan, String reconstruction from substring compositions, SIAM J. Discrete Math. 29 (3) (2015) 1340–1371.
[5] T. Batu, S. Kannan, S. Khanna, A. McGregor, Reconstructing strings from random traces, in: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 910–918.
[6] J. Bornholt, R. Lopez, D.M. Carmean, L. Ceze, G. Seelig, K. Strauss, A DNA-based archival storage system, ACM SIGARCH Comput. Archit. News 44 (2) (2016) 637–649.
[7] Y.M. Chee, R. Gabrys, A. Vardy, E. Yaakobi, Reconstruction from deletions in racetrack memories, in: 2018 IEEE Information Theory Workshop (ITW), IEEE, 2018, pp. 1–5.
[8] Y.M. Chee, S. Ling, Constructions for q-ary constant-weight codes, IEEE Trans. Inf. Theory 53 (1) (2007) 135–146.
[9] X. Chen, A. De, C.H. Lee, R.A. Servedio, 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.
[10] M. Cheraghchi, R. Gabrys, O. Milenkovic, J. Ribeiro, Coded trace reconstruction, IEEE Trans. Inf. Theory 66 (10) (2020) 6084–6103.
[11] G.M. Church, Y. Gao, S. Kosuri, Next-generation digital information storage in DNA, Science 337 (6102) (2012) 1628.
[12] R. Gabrys, E. Yaakobi, Sequence reconstruction over the deletion channel, IEEE Trans. Inf. Theory 64 (4) (2018) 2924–2931.
[13] K. Goyal, 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.
[14] R.N. Grass, R. Heckel, M. Puddu, D. Paunescu, W.J. Stark, Robust chemical preservation of digital information on DNA in silica with error-correcting codes, Angew. Chem. Int. Ed. 54 (8) (2015) 2552–2555.
[15] R. Heckel, G. Mikutis, R.N. Grass, A characterization of the DNA data storage channel, Sci. Rep. 9 (1) (2019) 1–12.
[16] M. Horovitz, E. Yaakobi, Reconstruction of sequences over non-identical channels, IEEE Trans. Inf. Theory 65 (2) (2018) 1267–1286.
[17] V. Junnila, T. Laihonen, T. Lehtilä, On Levenshtein's channel and list size in information retrieval, IEEE Trans. Inf. Theory 67 (6) (2020) 3322–3341.
[18] V. Junnila, T. Laihonen, T. Lehtilä, Levenshtein's reconstruction problem with different error patterns, in: 2023 IEEE International Symposium on Information Theory (ISIT), IEEE, 2023, pp. 1300–1305.
[19] V. Junnila, T. Laihonen, T. Lehtilä, The Levenshtein's sequence reconstruction problem and the length of the list, IEEE Trans. Inf. Theory (2023).
[20] V. Junnila, T. Laihonen, T. Lehtilä, On unique error patterns in the Levenshtein's sequence reconstruction model, preprint, arXiv:2406.14125, 2024.
[21] E. Laurent, J.-A. Amalian, M. Parmentier, L. Oswald, A. Al Ouahabi, F. Dufour, K. Launay, J.-L. Clément, D. Gigmes, M.-A. Delsuc, et al., High-capacity digital polymers: storing images in single molecules, Macromolecules 53 (10) (2020) 4022–4029.
[22] V.I. Levenshtein, Efficient reconstruction of sequences, IEEE Trans. Inf. Theory 47 (1) (2001) 2–22.
[23] V.I. Levenshtein, Efficient reconstruction of sequences from their subsequences or supersequences, J. Comb. Theory, Ser. A 93 (2) (2001) 310–332.
[24] V.S. Pless, W.C. Huffman, R.A. Brualdi (Eds.), Handbook of Coding Theory. Vol. I, II, North-Holland, Amsterdam, 1998.
[25] O. Sabary, H.M. Kiah, P.H. Siegel, E. Yaakobi, Survey for a decade of coding for DNA storage, IEEE Trans. Molec. Biolog. Multi-Scale Commun. (2024).
[26] K. Viswanathan, 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.
[27] E. Yaakobi, J. Bruck, On the uncertainty of information retrieval in associative memories, IEEE Trans. Inf. Theory 65 (4) (2018) 2155–2165.
[28] S.H.T. Yazdi, H.M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, O. Milenkovic, DNA-based storage: trends and methods, IEEE Trans. Molec. Biolog. Multi-Scale Commun. 1 (3) (2015) 230–248.